Fachbereich Mathematik und Informatik

AG Mathematics in Life Sciences

Integrated modelling of metabolic and regulatory networks

Aljoscha Palinkas

Dissertation zur Erlangung des Grades Dr. rer. nat.

Erstgutachter: **Prof. Dr. Alexander Bockmayr**
Zweitgutachter: **Prof. Dr. Hermann-Georg Holzhütter**

Tag der Disputation: 11. Mai 2015

meinen Eltern

Danksagung

Herzlicher und großer Dank geht an meinen Betreuer Prof. Alexander Bockmayr. Er hat mir fachlich und organisatorisch den Weg geebnet. Auch für die unkomplizierte Zusammenarbeit und die freundschaftliche Atmosphäre in der gesamten Arbeitsgruppe bin ich ihm sehr dankbar. So waren die vergangenen Jahre eine bereichernde und schöne Zeit voller wertvoller Erfahrungen und spannender Aufgaben. Meine vorliegende Arbeit, aber auch meine mathematische Bildung, haben von seiner Erfahrung und seinem weitreichenden Überblick über die behandelten Gebiete profitiert. Die Hinweise von Prof. Bockmayr, insbesondere auf bestimmte Referenzen, haben mir geholfen die Zusammenhänge zwischen meiner Arbeit und diversen anderen Problemstellungen zu erkennen. Ich bedanke mich auch für viele Hinweise, die zu einem besser lesbaren und logischeren Aufbau des Inhalts meiner Arbeit beigetragen haben.

Ebenso herzlich bedanke ich mich bei Prof. Hermann-Georg Holzhütter für sein großes Engagement in unserem gemeinsamen Projekt, mit dem er die vorliegende Arbeit möglich gemacht hat. Die Zusammenarbeit war für mich ein großes Privileg, da ich meine mathematischen Kenntnisse sofort auf ein aktuelles und kontrovers diskutiertes Thema anwenden konnte. Dank Prof. Holzhütter habe ich wertvolle Einblicke in die Feinheiten der Zellbiologie bekommen, was meinem Verständnis für biologische Prozesse und deren Modellierung einen wichtigen Schub gab. In unserer Kooperation habe ich auch viel über die Aufbereitung und Interpretation von numerischen Resultaten gelernt und Anregungen zur visuellen Darstellung der Daten bekommen. Auch Dr. Sascha Bulik war Teil dieser Kooperation und ich bedanke mich für seinen Einsatz und die gute Zusammenarbeit.

Viele Menschen haben meine Zeit in der AG Bockmayr schon durch ihre Anwesenheit bereichert. Prof. Heike Siebert hat mir zudem oft sehr wertvolle Hinweise zu meinen Präsentationen gegeben. Auch die Diskussionen mit allen meinen Kollegen hat mich immer wieder weitergebracht. Hier möchte ich Ling, Alexandra, Annika, Kirsten, Therese, Firdevs und Marco danken. Insbesondere bedanke ich mich bei Laszlo, Yaron und Arne, mit denen ich viele Fragen zu metabolischen Netzwerken besprechen und klären konnte. Meinen Bürogenossen Hannes, Shahrad und Adam danke ich außerdem für interessante Gespräche und die gute Atmosphäre im Büro. Auch für viele hilfreiche Kommentare zu meiner Dissertation danke ich meinen Kollegen. Schließlich geht mein Dank noch an Katja Geiger und Ekaterina Engel für ihre Hilfe.

Der DFG danke ich für das Stipendium im GK 1772 ”Computational Systems Biology” (CSB) und dem Matheon für die zweimonatige Anstellung. Bei allen Dozenten des CSB bedanke ich mich für die interessanten Vorträge. Den Koordinatorinnen des CSB möchte ich für die organisatorische Arbeit meinen Dank ausprechen.

Zusammenfassung

In der vorliegenden Dissertation wird die Regulation des zellulären Stoffwechsels durch Gene mit mathematischen Modellen beschrieben. Veränderungen in der Expression von Genen, die für die Produktion von Enzymen verantwortlich sind, bewirken längerfristige Umstellungen im Stoffwechsel. Auf diese Weise wird auf der Transkriptionsebene z.B. die Anpassung des Stoffwechsels an äußere Bedingungen oder sich ändernde Bedürfnisse der Zelle gesteuert. Ein Modell der Verteilung von Ressourcen für die Produktion von Enzymen wird in Kap. 3 entwickelt und in ein constraintbasiertes Stoffwechselmodell integriert. Die Flussraten der einzelnen Reaktionen werden dabei von der Menge an verfügbaren Enzymen beschränkt, welche wiederum von der Verteilung der Ressourcen auf das ganze Stoffwechselnetzwerk abhängt. In Kap. 4 wird dieses Modell dann verwendet, um der Hypothese nachzugehen, dass Umstellungen des Stoffwechsels der Zelle dazu dienen könnten, verschiedene benötigte Metaboliten mit höchstmöglicher Effizienz zu produzieren. Zuerst analysieren wir die Effizienz an einem Spielmodell, bevor dann ein Netzwerk des zentralen Kohlenstoffwechsels untersucht wird. In diesem Modell betrachten wir die Produktion von einigen Bausteinen der Biomasse. Zusätzlich wird die permanente Bereitstellung von genügend Energie und Antioxidantien gefordert. Das Ziel ist dabei, die geforderte Produktion von Metaboliten in möglichst kurzer Zeit zu erfüllen. Das mathematische Modell ist ein Optimierungsproblem mit gemischt-ganzzahligen Variablen, linearen und wenigen quadratischen Nebenbedingungen sowie einer quadratischen Zielfunktion. Eine Lösung entspricht einer Abfolge von Flussverteilungen und deren Dauer. Die Berechnungen in Kap. 4 zeigen, dass das Umschalten zwischen verschiedenen Flussverteilungen des Stoffwechsels es ermöglicht, die Biomasse in einer signifikant kürzeren Zeit zu produzieren als es eine einzelne Flussverteilung erlauben würde. Die Robustheit dieser Ergebnisse bezgl. der Parameterwahl wurde empirisch bestätigt. Die mathematischen Eigenschaften des Ressourcenverteilungsmodells werden in Kap. 3 analysiert. Unser Modell geht von einer groben Steuerung der Enzymkonzentrationen auf der Transkriptionsebene aus, was durch binäre Genexpression modelliert wird, die nur festlegt, welche Stoffwechselpfade aktiviert sind und welche nicht. Weiterhin gibt Kap. 3 eine Charakterisierung von bestimmten besonders effizienten Flussverteilungen. Aus dieser endlichen Menge kann immer eine Abfolge zusammengestellt werden kann, die optimal die gegebenen Anforderungen erfüllt, d.h. eine optimale Lösung unseres Optimierungsproblems ist. Für große Netzwerke sind die Optimierungsprobleme, die das Ressourcenverteilungsmodell formuliert, numerisch nicht lösbar. Daher werden in Kap. 6 verschiedene alternative Berechnungsmethoden vorgestellt. In Kap. 5 werden stochastische Störungen der Nebenbedingungen, die den Flussraum beschränken, untersucht. Es zeigt sich, dass hier ein unerwarteter Effekt auftritt. Er wird durch die im mathematischen Sinne nicht eindeutige Darstellung des Stoffwechselmodells durch lineare Nebenbedingungen bestimmt.

Zur Modellierung und Untersuchung der Dynamik von genregulatorischen Netzwerken wird oft der Formalismus der sogenannten logischen Netzwerke verwendet. In Kap. 7 wird ein Algorithmus vorgeschlagen, der eine kurze und gut lesbare Darstellung der benötigten logischen Funktionen liefert. Eine Erweiterung von logischen Netzwerken zu einem Markov-Prozess wird in Kap. 8 vorgeschlagen, um stochastische und quantitative Aspekte darzustellen. Exemplarisch wird gezeigt, wie sich dieser Formalismus direkt mit einer Mastergleichung für bestimmte Reaktionen oder regulatorische Interaktionen kombinieren lässt. In einem Ausblick wird in Kap. 9 das Problem der Feedback-Regulation des genregulatorischen Netzwerkes durch den Stoffwechsel behandelt.

Abstract

Two cellular subsystems are the metabolic network and the gene regulatory network. In systems biology they have mostly been modelled in isolation with ordinary differential equations (ODEs) or with tailored formalisms as e.g. constraint-based methods for metabolism or logical networks for gene regulation. In reality the two systems are strongly interdependent. For mathematical modelling the integration is a challenge and a variety of different approaches has been proposed.

Long term alterations in metabolism result from changes in gene expression, which determines the production of enzymes. This transcriptional control can adjust the metabolic network to changes in the environment or the requirements of the cell. In fact, the cell cycle is connected to cyclic changes in metabolism, so-called metabolic cycling, but alterations are also observed in non-proliferating cells in a constant environment. A mathematical model to describe and explain alterations in metabolism will be proposed here. At first, a resource allocation model for the enzymes in a metabolic network is developed and integrated into a constraint-based model of metabolism in Chap. 3. The reaction rates are bounded depending on the availability of enzymes, which in turn is determined by the overall distribution of the limited resources. In Chap. 4, this model is used to test the hypothesis that metabolic alterations are a means of the cell to achieve the required production of metabolic output most efficiently. First a toy model is analysed and then the method is applied to a core metabolic network of the central carbon metabolism. The tasks of this metabolic network are the production of biomass precursors as well as constantly providing a minimum of energy and anti-oxidants.

The mathematical model gives a mixed integer linear optimisation problem with a few quadratic constraints and a quadratic objective function. Instead of searching for a single flux distribution, a feasible solution corresponds here to a sequence of several flux distributions together with the time that is spent in each of them. The consecutive usage of these flux distributions during the associated time spans yields the required output. The objective is the minimisation of the total time needed. The computations demonstrate that switching between several flux distributions allows producing the output in a significantly shorter time span, compared to an optimal single flux distribution.

In a toy model we could identify the relationship between the model parameters and the results concerning the efficiency of static versus sequential flux distributions. Such a comprehensive analysis is not possible for the large number of parameters in our core metabolic network. To make sure that the confirmation of the hypothesis is not restricted to a minor region in the parameter space of the resource allocation model, we perturbed the parameters randomly and repeated all computations. This empirical analysis showed that the significant gain in performance is a robust feature of the model.

From the mathematical point of view the proposed resource allocation model defines for each gene expression state a flux space from which a flux distribution can be chosen. This flux space is in general not linear and not convex, which turns out to depend on the space of all possible gene expression states. In our model the genes regulate the enzyme concentrations in an on-off manner, only determining the active and inactive parts of metabolism. Furthermore, certain groups of genes are regulated together as functional units. As a consequence, the enzyme concentrations cannot be perfectly adjusted to a given flux distribution in this model and it is for this reason that switching can increase the efficiency. A simpler model of resource allocation, which is solely based on molecular crowding, has been proposed before in the lit-

erature. It allows distributing the resources to perfectly match any given flux distribution and switching is then not necessary to obtain the minimal production time. In contrast to such a resource allocation model, our modelling assumptions and computational results suggest a design principle, where the optimal adjustment to given conditions and requirements is not achieved by fine-tuning of enzyme concentrations, but by switching between different flux distributions, which are only roughly determined by transcriptional control and which do not perfectly match one certain condition or requirement. In terms of geometry, the difference lies in the convexity of the flux space. If it is convex, minimal production time can always be achieved with a single flux distribution.

To characterise a set of flux distributions sufficient to constitute an optimal sequence, the flux space of the network without the resource allocation model is considered in Chap. 3. The corresponding polytope allows characterising a finite subset of the flux space in terms of decomposability, a notion which is closely related to elementary modes. For any output requirements, an optimal sequence can be constituted from this finite set of flux distributions.

In practice, solving the optimisation problem that was derived from the modelling approach as well as computing the sufficient finite subset, is not tractable for large networks. Also divide and conquer strategies are not promising to obtain optimal solutions in general, a counterexample is given in Chap. 6. Alternative computational methods to obtain optimal or approximative optimal solutions are then presented. The gene regulatory network behind the metabolic genes is not fully considered in the resource allocation model of Chap. 3. Only some constraints are added in the application to the core metabolic network in order to exclude unrealistic patterns of gene expression. Incorporating more information about the gene regulation into the computational model is in fact improving the tractability, because the search space is reduced. A sufficiently small search space of gene expression sequences gives the possibility to perform a more precise and extensive analysis using an alternative computational approach.

In Chap. 5, the perturbations of model parameters, as applied to the core metabolic network to verify the robustness, are considered in general. From the mathematical point of view, the linear constraints that bound the flux space are perturbed. The consequences on the geometry of the flux space and on the objective value of an optimisation problem over this flux space are analysed and an effect is discovered, which is surprising at first sight. If the bounds on the reaction rates are perturbed individually, without a bias for increase or decrease, the expected objective value of a given linear optimisation problem is decreased in expectation. This effect emerges from the representation of the flux space. In particular redundancy of the constraints plays a crucial role.

The modelling and the analysis of the dynamics of gene regulatory networks with so-called logical networks is a common discrete approach. Logical networks are often represented by logical functions, which have the advantage of being mathematical objects that can be given in a natural and easily understandable format, namely Boolean expressions. In Chap. 7, a method is presented to obtain a short and well readable representation of a given logical function. It is based on the minimisation of Boolean expressions, but is designed for multi-valued logical functions in particular.

All possible dynamics of a logical network can be represented in the so-called state transition graph. Simply by assigning rates to all edges, which represent the transitions between different states, this directed graph becomes a continuous time Markov chain (CTMC) which we call a stochastic logical network. This modelling approach opens new possibilities for the analysis of quantitative dynamical properties as shown in Chap. 8. In contrast to this abstract model,

detailed mechanistic and stochastic models of biochemical reaction systems can be formulated with the chemical master equation, which also defines a CTMC. In fact, these two formalisms can be combined, so that distinct components of the biological system are modelled in much detail by the master equation and other parts on a higher abstraction level as a stochastic logical network. The combined model can focus on certain aspects, capturing related quantitative and stochastic effects, while keeping the overall complexity to a minimum.

Finally, Chap. 9 discusses the feedback regulation from metabolism to gene regulation. In an integrated dynamic model of gene regulation and metabolism, this aspect should not be missing. Since constraint-based models neglect the concentrations of metabolites, it is difficult to determine the regulatory feedback to the genes. This problem can be circumvented by only inferring metabolic mediated interactions between genes, in the sense that a switch in gene expression leads to an alteration in the metabolic network, which in turn gives a new regulatory input to the gene network. To this end, a constraint-based approach is proposed and compared to a method from the literature, which is based on metabolic sensitivity analysis. Furthermore, a strategy to derive concentration changes from changes in flux rates and enzyme activities is shortly presented.

Contents

Mathematical preliminaries

Basics

The natural numbers are denoted by $\mathbb{N}$ and 0 is not considered as an element of $\mathbb{N}$, but $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. The integers are denoted by $\mathbb{Z}$ and $\mathbb{R}$ is the set of real numbers. Intervals in $\mathbb{R}$ are denoted by $[a,b] := \{x \in \mathbb{R} : a \le x \le b\}$ and $(a,b) := \{x \in \mathbb{R} : a < x < b\}$. By $\mathbb{R}^{m\times n}$ we denote the space of matrices with m rows and n columns and real-valued entries. Vectors are given by lower case letters $a \in \mathbb{R}^n$ and matrices by upper case letters $A \in \mathbb{R}^{m\times n}$. The transpose of a matrix or a vector is denoted by $A^\top$ resp. $a^\top$. For non-negative integers k, upper indices a^k, A^k are usually not an exponent but an index. Exponentiation of $a \in \mathbb{R}$ or $A \in \mathbb{R}^{n\times n}$ is written as $(a)^k$, $(A)^k$, while the multiplicative inverse elements are denoted by a^{-1}, A^{-1}. If $I = \{i(1), i(2), \dots, i(k)\} \subset \{1,\dots,n\}$ is a set of indices, then v_I is the vector $(v_{i(1)},\dots,v_{i(k)})^\top \in \mathbb{R}^k$ and $A_{\cdot I}$ is the matrix that is obtained from A by deleting all columns of indices that are not in I. Similarly, for an index set $I = \{i(1), i(2), \dots, i(k)\} \subset \{1,\dots,m\}$, $A_{I\cdot}$ is defined by deleting rows of A. In particular $A_{\cdot j}$ denotes the j-th column and $A_{i\cdot}$ the i-th row of A. The **support of a vector** $x \in \mathbb{R}^n$ is the set of indices of those entries that are not zero, it is denoted by $\operatorname{supp}(x) := \{i : x_i \neq 0\}$.

Similarly, the **support of a function** $f\colon D \to \mathbb{R}$ is the subset of the domain, where f evaluates not to zero, i.e., $\operatorname{supp}(f) := \{x \in D : f(x) \neq 0\}$. The **image of a function** $f\colon D \to \mathbb{R}^n$ is defined as $\operatorname{im}(f) := \{y \in \mathbb{R}^n : \text{ exists } x \in D \text{ s.t. } f(x) = y\}$. Let $D \subset \mathbb{R}$ and $f_n\colon D \to \mathbb{R}$, $n \in \mathbb{N}$ be an infinite sequence of functions, then f_n is said to **approximate** the function $g\colon D \to \mathbb{R}$, or to **converge to** g, if for every $x \in D$ and for every $\varepsilon > 0$ there exists an $n_{x,\varepsilon} \in \mathbb{N}$, such that $|f_n(x) - g(x)| < \varepsilon$ for all $n \ge n_{x,\varepsilon}$.

The term **scaling** denotes multiplication of numbers, vectors or real-valued functions with a strictly positive scalar $\alpha \in \mathbb{R}_{>0}$.

Linear algebra

Given vectors $x^1,\dots,x^k \in \mathbb{R}^n$, a **linear combination** of these vectors is a vector $z = \sum_{i=1}^k \lambda_i x^i$, with $\lambda_i \in \mathbb{R}$, $i = 1,\dots,k$. If $z = 0$ implies $\lambda_i = 0$, $i = 1,\dots,k$, then the vectors $x^1,\dots,x^k$ are **linearly independent**. The **rank** of a matrix is the maximal number of linearly independent columns. A linear **subspace** $V \subset \mathbb{R}^n$ is a subset that is closed under linear combinations, which means that every linear combination of elements from V is again an element of V. Given a matrix $A \in \mathbb{R}^{m\times n}$, the **nullspace** or kernel of A is $\operatorname{null}(A) := \{x \in \mathbb{R}^n : Ax = 0\}$. The nullspace is a subspace of $\mathbb{R}^n$.

A **mapping** $f\colon \mathbb{R}^n \to \mathbb{R}$ is **linear** if for any $\lambda,\mu \in \mathbb{R}$ and $x,y \in \mathbb{R}^n$ we have $f(\lambda x + \mu y) = \lambda f(x) + \mu f(y)$. The standard **scalar product** in $\mathbb{R}^n$ is a mapping $\langle\cdot,\cdot\rangle\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ given by $\langle x,y\rangle := \sum_{j=1}^n x_j y_j$. Alternatively it can be expressed by matrix multiplication of the two vectors, i.e., $\langle x,y\rangle = x^\top y$. This is bilinear, which means that for fixed x or y the mapping $\langle x,\cdot\rangle\colon \mathbb{R} \to \mathbb{R}$ resp. $\langle\cdot,y\rangle\colon \mathbb{R} \to \mathbb{R}$ is linear.

Convexity and polyhedra

A linear combination $z = \sum_{i=1}^k \lambda_i x^i$ of vectors $x^1,\dots,x^k \in \mathbb{R}^n$ is called a **conic combination** if $\lambda_i \ge 0$, $i = 1,\dots,k$, and a **convex combination**, if additionally $\sum_{i=1}^k \lambda_i = 1$. Given a finite set of vectors $x^1,\dots,x^k \in \mathbb{R}^n$, the set of all linear combinations defines a linear subspace denoted as

$\text{lin}(x^1, \ldots, x^k) := \{z \in \mathbb{R}^n : z = \sum_{j=1}^k \lambda_j x^j,\ \lambda_j \in \mathbb{R}\}$. We say that this subspace is generated by the vectors $x^1, \ldots, x^k$. Similarly, we define the **conic hull** as $\text{cone}(x^1, \ldots, x^k) := \{z \in \mathbb{R}^n : z = \sum_{j=1}^k \lambda_j x^j,\ \lambda_j \in \mathbb{R}_{\geq 0}\}$ and the **convex hull** as $\text{conv}(x^1, \ldots, x^k) := \{z \in \mathbb{R}^n : z = \sum_{j=1}^k \lambda_j x^j,\ \lambda_j \in \mathbb{R}_{\geq 0},\ \sum_{j=1}^k \lambda_j = 1\}$.

If $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, then $Ax \leq b$, $x \in \mathbb{R}^n$ is a **system of linear inequalities**. The solution set of a system of linear inequalities is called a **polyhedron**, $P := \{x \in \mathbb{R}^n : Ax \leq b\}$. If $b = 0$ the system of linear inequalities is called **homogeneous** and the corresponding polyhedron P is called a **polyhedral cone**. More generally, a **cone** is a subset of $\mathbb{R}^n$ that is closed under conic combinations and a polyhedral cone is a cone that is also a polyhedron. A **polytope** is a bounded polyhedron.

Each of the inequalities $A_{i,\cdot}\, x \leq b_i$ defines the so-called **halfspace** $H_i := \{x \in \mathbb{R}^n : A_{i,\cdot}\, x \leq b_i\}$ and the hyperplane $\overline{H}_i := \{x \in \mathbb{R}^n : A_{i,\cdot} x = b_i\}$. The polyhedron $P := \{x \in \mathbb{R}^n : Ax \leq b\}$ can be expressed as the intersection of the corresponding halfspaces, i.e., $P = \bigcap_{i=1}^m H_i$.

Topology

In $\mathbb{R}^n$ we denote by $\|\cdot\|$ the 1-**norm**, i.e., $\|x\| := \sum_{j=1}^n |x_j|$. For $v \in U \subset \mathbb{R}^n$ and $\varepsilon > 0$ the ε-**ball** in U around v is the set $\{x \in U : \|v - x\| \leq \varepsilon\}$. The set $V \subset U$ is an **open subset** of U, if for every $x \in V$ there is an $\varepsilon > 0$, such that the ε-ball around x in U is contained in V. A **neighbourhood** of x is a subset of U that contains an ε-ball around x. A subset $B \subset U$ is a **closed subset** of U if $U \setminus B$ is open. For example, the interval $[a, b] := \{x \in \mathbb{R} : a \leq x \leq b\}$ is closed and the interval $(a, b) := \{x \in \mathbb{R} : a < x < b\}$ is open in $\mathbb{R}$. An arbitrary union of open sets is open and finite intersections of open sets are also open. The empty set is always open and closed by definition. A point $x \in U \subset \mathbb{R}^n$ is an **interior point** of U if an ε-ball around x in $\mathbb{R}^n$ is contained in U. The subset $R \subset U$ of all points that are not interior points of U is called the **boundary** of $U \subset \mathbb{R}^n$.

Linear programming and other optimisation problems

A linear programming problem (also called **linear optimisation problem** or just **LP** or linear problem) can be stated as follows:

- Given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$, find a vector $x^* \in P := \{x \in \mathbb{R}^n : Ax \leq b\}$ maximising the linear function $c^\intercal x$ over P.

The linear function $c^\intercal x$ is called the **objective** function. Every y that fulfils the constraints $Ay \leq b$ is called a **feasible solution**. If it is also maximal, i.e., $c^\intercal y \geq c^\intercal x$ for all feasible vectors x, then y is called an **optimal solution**. An LP can be seen as a special instance from the general class of **optimisation problems**, where a search space $U \subset \mathbb{R}^n$ and an objective function $f\colon \mathbb{R}^n \to \mathbb{R}$ are defined and the goal is to find an element $x^* \in U$, such that $f(x^*) \geq f(y)$ for all $y \in U$. The space U is usually defined by constraints of the form $g(x) \leq b$ where $g\colon \mathbb{R}^n \to \mathbb{R}$ and $b \in \mathbb{R}$. In the case of an LP, the constraints as well as the objective function are linear, but we will also consider more general optimisation problems here.

Generalisations If all variables in an LP are constrained to the range of integers, i.e., $x \in \mathbb{Z}^n$, the problem is called an **integer linear program**. If there are variables with integer ranges as well as variables with continuous ranges, we call it a **mixed-integer linear program (MILP)**. In case the objective function is quadratic, i.e., $f(x) = x^\intercal Q x + c^\intercal x$, with a positive semidefinite matrix $Q \in \mathbb{R}^{n \times n}$ and $c \in \mathbb{R}^n$ and linear constraints, we have a **quadratic program (QP)**.

With integer variables this becomes an MIQP. If we have furthermore quadratic constraints $x^\top Q'x + a^\top x \leq b$, where $Q' \in \mathbb{R}^{n\times n}$ is positive semidefinite, $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, this is called a **quadratically constrained program (MIQCP)**. The condition that the matrices in the constraints and the objective function are positive semidefinite makes sure that these optimisation problems belong to the class of so-called convex optimisation problems [Boyd and Vandenberghe, 2004, Sec. 4.4, p. 152].

Propositional calculus, Boolean expressions and logical functions

In propositional calculus the logical operators "and", "or", "not", (denoted by $\wedge$, $\vee$, $\neg$, respectively, or $\cdot$, $+$, $^-$) are used to compose expressions from so-called atomic expressions $x_1, \ldots, x_n$. If x, y are expressions, then the **negations** $\overline{x}$ and $\overline{y}$ as well as the **conjunction** $x \cdot y$ and the **disjunction** $x + y$ are also expressions. The set of all expressions that can be derived this way is denoted by $\mathcal{E}$, the elements of $\mathcal{E}$ will be called **Boolean expression**. The atomic expressions will be called (Boolean) variables. An **assignment** of the variables is a function $A\colon \{x_1, \ldots, x_n\} \to \mathbb{B}$, where $\mathbb{B} := \{0, 1\}$ denotes the range of the Boolean values 0 for *false* and 1 for *true*. The assignment A extends to a function $\hat{A}\colon \mathcal{E} \to \mathbb{B}$ by a recursive definition as follows. Let $f \in \mathcal{E}$. If f is a variable, $\hat{A}(f) = A(f)$. If $f = g \cdot h$, then $\hat{A}(f) = 1$ in case $\hat{A}(g) = 1$ and $\hat{A}(h) = 1$, otherwise $\hat{A}(f) = 0$. If $f = g + h$, then $\hat{A}(f) = 1$, in case $\hat{A}(g) = 1$ or $\hat{A}(h) = 1$, otherwise $\hat{A}(f) = 0$. If $f = 1$, then the negation $\overline{f}$ of f is mapped to $\hat{A}\left(\overline{f}\right) = 0$, otherwise, if $f = 0$ it is mapped to $\hat{A}\left(\overline{f}\right) = 1$. This formal construction is taken from [Schöning, 1992, p. 14,15].

The space of all assignments of n Boolean variables is $\mathbb{B}^n$ and an expression $f \in \mathcal{E}$ defines a function $f\colon \mathbb{B}^n \to \mathbb{B}$ by $f(A) := \hat{A}(f)$ for an assignment $A \in \mathbb{B}^n$. Two expressions f, g are called equivalent, if they define the same function, i.e., $f(A) = g(A)$ for every assignment $A \in \mathbb{B}^n$.

Probability and random processes

Random variables are functions which are defined on a probability space. This is a triple $(\Omega, \mathscr{F}, P)$ [Georgii, 2007, p. 13], where Ω is the sample space, $\mathscr{F}$ is a σ-algebra and $P\colon \mathscr{F} \to [0, 1]$ is the probability measure, characterised by $P(\Omega) = 1$ and σ-additivity. For the definition of the σ-algebra and σ-additivity see [Georgii, 2007, Sec. 1]. All subsets of Ω that are considered here are implicitly assumed to be elements of $\mathscr{F}$. A real-valued random variable is a mapping $X\colon \Omega \to \mathbb{R}^n$ which is $\mathscr{F}$-measurable, if $\mathbb{R}^n$ is equipped with the Borel-σ-algebra [Georgii, 2007, p. 11 and 21]. We assume that for a random variable $X\colon \Omega \to \mathbb{R}$, the technical conditions are given [Øksendal, 2007, Sec. 2.1] to define the **cumulative distribution function (CDF)** $F_X\colon \mathbb{R} \to [0, 1]$ of X as $F_X(x) = P(X \leq x)$ (abbreviating $P(\{\omega \in \Omega : X(\omega) \leq x\})$) and the **probability density function (PDF)** $f_X\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ as the derivative of F_X, i.e., $f_X = F_X'$. The **expectation value** of X is defined if $\int_{\mathbb{R}} |x| f(x) dx < \infty$ and it is $E(X) := \int_{\mathbb{R}} x f(x) dx$ [Georgii, 2007, Korollar 4.13, let the f there be the identity and the ρ the PDF f of X]. Alternatively it can be given by $E(X) = \int_\Omega X(\omega) dP(\omega)$ [Øksendal, 2007, Sec. 2.1]. The median of X is the value $y \in \mathbb{R}$, where strictly increasing CDF takes the value 0.5, i.e., $F(y) = P(X \leq y) = 0.5$. If we have a finite family of real-valued random variables X_i, $i = 1, \ldots, n$, on the same probability space $(\Omega, \mathscr{F}, P)$, then this family is called **independent** (w.r.t. P), if for arbitrary $x_i \in \mathbb{R}$ it holds that $P(X_1 \leq x_1, \ldots, X_n \leq x_n) = \prod_{i=1}^n P(X_i \leq x_i)$ [Georgii, 2007, Korollar 3.21]. Random variables can also be discrete valued, i.e., $X\colon \Omega \to \mathbb{N}$. In this case, the expectation value can be given as $E(X) = \sum_{x\in\mathbb{N}} f(x) \cdot x$, where f is the PDF of X.

Stochastic processes and Markov chains A stochastic process is a family $\{X_t : t \in T\}$ of random variables $X_t: \Omega \to U$ to the state space $U \subset \mathbb{R}^n$ or $U \subset \mathbb{N}^n$ and indexed by some set T, which represents time [Grimmett and Stirzaker, 1992, p. 332]. Assuming that time has a fixed starting point $t = 0$, we set $T = \{0, 1, 2, \ldots\}$, if discrete time is considered, or $T = \mathbb{R}_{\geq 0}$, if continuous time is considered. The idea is that the random variables describe the evolution of a probability density over time. **Markov chains (MCs)** are discrete time processes that take values in a countable set U and which have the so-called Markov property that $P(X_t = u \mid X_0 = x_0, X_1 = x_1, \ldots, X_{n-1} = x_{n-1}) = P(X_t = u \mid X_{n-1} = x_{n-1})$ for all $t = 1, 2, \ldots$ and $u \in U$ (the notation $P(A \mid B)$ is for conditional probability, i.e., the probability of A given B). It states that the process is **memoryless**, i.e., its future depends not on its history, but only on its current state. We will always assume implicitly that MCs are homogeneous, which means that $P(X_{t+1} = v \mid X_t = u) = P(X_1 = v \mid X_0 = u)$, for all $t = 0, 1, 2, \ldots$ and all $u, v \in U$. In this case the MC is completely defined by the **transition probabilities** $p_{u,v} := P(X_1 = v \mid X_0 = u)$.

Similarly, **continuous time Markov chains (CTMCs)** are defined as memoryless stochastic processes in continuous time. The formal construction is however technically more involved and will be omitted here [Grimmett and Stirzaker, 1992, p. 240f]. Under mild conditions, which are always assumed to be satisfied, the CTMC is completely defined by the **generator matrix** G, analogously to the transition probabilities $p_{u,v}$ for the MC. The off diagonal entries $g_{u,v}$ of the generator matrix give a **rate for the transition** from state u to state v. If $g_{u,v} = 0$, this transition cannot occur and otherwise, the transition will occur after a waiting time that is exponentially distributed with parameter $g_{u,v}$. The CTMC will rest in a given state until one of the outgoing transitions occurs. The waiting time (or exit time) in state u is exponentially distributed with parameter $\sum_{v \in U \setminus \{u\}} g_{u,v}$ [Grimmett and Stirzaker, 1992, p. 243,(13)*]. By definition, the diagonal entries of the generator matrix are $g_{u,u} = -\sum_{v \in U \setminus \{u\}} g_{u,v}$ [Grimmett and Stirzaker, 1992, p. 241,(6)*].

Stationary distribution Given a CTMC or MC on a finite state space U, we might think of the stochastic process as starting in a distinct initial state $u \in U$, which means that $P(X_0 = u) = 1$. The random dynamics lead to a probability density on the state space at all time points $t > 0$. Under certain conditions, the distribution will converge in the long run to a so-called stationary distribution $\pi \in [0, 1]^{|U|}$, with $\sum_{u \in U} \pi_u = 1$ and $\pi \geq 0$. If a stationary distribution exists, it can be characterised in the case of MCs by the property $\pi = \pi P$, where P is the matrix of the transition probabilities, and in the case of CTMCs by $\pi G = 0$ [Grimmett and Stirzaker, 1992, p. 207 resp. p. 244].

1 Introduction

Overview This thesis will be mainly about metabolic networks and the way they are regulated on a transcriptional level. The aim is the integration of metabolism and gene regulation in one model. However, a fully integrated modelling approach which represents the genetic regulation of metabolism as well as the regulatory feedback from metabolism to the gene regulatory network will not be given. At first, the regulation of the metabolic network by genes is described by a new modelling approach. The mathematical description and analysis of this model and its implications is done in Chapters 3,4,6 and partly also in 5. The remainder is then concerned with modelling the dynamics of the gene regulatory networks. The feedback from the metabolic network to gene regulation, which is necessary to give a completely integrated model, is discussed in an outlook in Chap. 9.

Transcriptional regulation of metabolism Models based on differential equations or the chemical master equation give a mechanistic description of the metabolic network. As a consequence, these models require many kinetic parameters that measure different properties of the enzymes. In contrast, this text deals mainly with constraint-based models, a phenomenologically motivated approach which requires much less biological information. It is based on the steady-state assumption which states that all metabolites in the metabolic network are consumed and produced with equal rates. Usually, optimisation is used to find a single flux distribution that fulfils the steady-state assumption as well as further requirements and that optimises a given objective. Reaction fluxes would only alter in this model if a change in the objective or in the environment occurs, so that the current flux is not optimal anymore. Therefore, most constraint-based models describe either one static metabolic phase or changes in metabolism that are caused by the environment.

Here we propose a constraint-based method that represents alterations in metabolism as sequences of different steady-state flux distributions. The motivation is to investigate the interplay between alterations in the flux distribution and the efficiency of metabolic output production. For this purpose, a model of resource allocation in metabolism is introduced, based on the limited total amount of amino acids available for enzyme synthesis. This total amount is distributed to produce enzymes for all reactions of the network. The distribution is controlled by the expression levels of the coding genes. The amount of enzyme for a given reaction determines the maximal flux rate of that reaction. One distinct pattern of expression levels of all metabolic genes defines thus a space of possible flux distributions. This is the usual scenario of constraint-based modelling. One particularity of our resource allocation model is that it leads to non-linear flux spaces. The other particularity is that we consider sequences of different flux distributions

from different flux spaces.

As the gene expression pattern changes, the space of possible flux distributions changes accordingly. This gives a model where a discrete dynamics on the transcriptional level controls possible dynamics in the metabolic network. The main part of this text deals with this model, where the underlying gene regulatory network that drives the dynamics is not considered explicitly. In particular, the feedback regulation from the metabolic network into the gene regulatory network is neglected.

The examination of this new model begins with a theoretical analysis to find out how the mathematical structure determines the behaviour and how it compares to related modelling approaches. In constraint-based models the mathematical structure is given by the flux space and the theoretical analysis hence deals with these geometric objects. A central aim in the field of metabolic pathway analysis is to identify finite subsets of the flux space which give a representation of the capacity of the metabolic network. Also in our modelling approach such finite subsets of steady-state flux distributions play an important role.

The proposed resource allocation model describes precisely how the gene expression patterns are translated to a flux space defining the capacity of metabolism. Although the resource allocation model is stated in a general form where any state of possible gene expression patterns can be used, we subsequently focus on binary gene expression. This choice is meant to reflect the role that transcriptional regulation plays in the control of reactions rates. On the transcriptional level, only the activation or deactivation of the different pathways of the metabolic network is decided. Homeostasis of metabolites is then achieved by mechanisms of the metabolic network itself, as e.g. product inhibition. As a consequence, only few reactions are rate limiting, while for a large part of reactions more enzymes than needed are available. In the mathematical model, the metabolic genes are therefore represented by binary variables which only determine if the associated reaction is activated or not. The limited total amount of available amino acids is then distributed between all activated reactions.

Regulation of gene expression The resource allocation model describes how a gene expression pattern is translated into bounds on the flux rates and thus into a flux space of possible steady-state flux distributions. Other questions are which kind of gene expression patterns and which sequences of different gene expression patterns are possible. The dynamics of the gene regulatory network comes into play here. Ideally we would have a complete model of the gene regulatory network which would allow inferring possible dynamics of the metabolic genes. If such a complete model is not available, it is also possible to incorporate only some aspects of gene regulation. This can be done by restricting the space of possible gene expression patterns or the possible sequences of different gene expression patterns. In the application of our approach in Chap. 4 for example, we identify functional groups of reactions which we assume to be regulated as a unit, i.e., all genes in one group must always have the same expression level. Furthermore, we consider the restriction that only additional genes can be activated in the transition from one phase to the next. This reflects in particular a situation of rapidly dividing cells. In general, more information about the gene regulation will lead to more restrictions on the possible patterns of gene expression.

Modelling gene regulation with logical networks Since we consider binary gene expression and are interested in restricting the possible dynamics rather than predicting one specific evolution of the gene regulatory network, the modelling framework of logical networks is well suited.

Similar to constraint-based models of metabolism, logical networks are highly abstract models and can be built with a minimum of information. In particular, no kinetic parameters of the interactions are required. Instead, interactions are described by logical rules. As a consequence of this limited information incorporated in the model, logical networks are usually not suited to predict a precise dynamical behaviour, but they are useful to restrict the space of possible dynamics and to infer specific properties of the dynamics and to exclude certain behaviours.

Some aspects of this logical modelling approach will be discussed later in this text. Chap. 7 deals with the representation of logical functions and Chap. 8 discusses a probabilistic extension of logical networks. All biochemical reactions or interactions are subject to stochasticity as e.g. fluctuations in the concentration of species and also fluctuations in the precise value of a kinetic parameter. In metabolism, stochasticity can be neglected, due to the large amount of molecules that react on a very fast time-scale. However, in gene regulation, stochasticity can play an important role in the constitution of a phenotype. In the here proposed stochastic logical framework, the gene regulatory network jumps between the discrete gene expression patterns with a certain resting time in each state. This kind of representation of a dynamic behaviour is similar to our method for alterations in the flux distribution of metabolism, where a sequence is computed that jumps between phases of specified duration with different steady-state flux distributions.

Regulatory feedback to the gene regulatory network The information about possible dynamics in the gene regulatory network that we obtain from a logical network model, is well suited to be used directly in our constraint-based model of the metabolic network. In the other direction, it is very difficult to incorporate the regulatory feedback from the metabolism to the genes. This feedback is caused by metabolites that interact with certain genes and hence control their expression. The feedback thus depends on the concentration of these metabolites. However, the simplicity of the constraint-based approach is based on replacing reaction kinetics, including metabolite concentrations, with the steady-state assumption. The concentration levels of the metabolites are then completely neglected. Retrieving metabolite concentrations from a flux distribution can only be done under some strong assumptions and is therefore bound to give false predictions in some cases. However, such approaches have been applied and in Chap. 9 this strategy is discussed. Also other existing and own approaches to identify the regulatory feedback from a metabolic network that is regulated by the gene regulatory network are discussed. To complete the integrated model of metabolism and gene regulation, such a method is indeed necessary. As the metabolism is altered by transcriptional control, also the feedback can change. The dynamics of the whole integrated model can thus only be fully understood if the feedback regulation is incorporated.

2 Mathematical modelling of metabolic networks

Some methods and concepts from constraint-based modelling [Price et al., 2004a] will be introduced in this chapter. To explain the motivation and justification of the constraint-based approach, we first introduce the deterministic kinetic model of a biochemical reaction system, see e.g. [Heinrich and Schuster, 1996, Sec. 2.2] or [Murray, 1989, Sec. 5].

2.1 Basic elements of constraint-based modelling

In its simplest form, constraint-based models just include the stoichiometry of the metabolic reactions and their type, reversible or irreversible. This leads to a simple geometric object, the flux cone, representing states of the metabolic network, where all metabolites are consumed and produced in equal rates. Generating elements of this flux space will be discussed as well as redundancy in the mathematical representation of the bounds on reaction rates and also the problem of futile cycles.

The basic components of a model of a biochemical reaction system are a set of m species and a set of n reactions. These sets are indexed by the natural numbers and we will refer to a distinct metabolite by its index $i \in \{1, \ldots, m\}$ and similarly to a distinct reaction by $j \in \{1, \ldots, n\}$. Based on the postulates that the reaction kinetics depends only on the metabolite concentrations and that no other processes beside the reactions change the concentrations [Heinrich and Schuster, 1996, p. 13], the concentrations and reactions constitute a dynamical system. The state of the system at a given time point t is described by

- the concentrations, $c(t) \in \mathbb{R}^m$, where $c_i(t)$ is the concentration of species i at time t and
- the current change in concentration, i.e., the derivative of the concentrations, $\frac{dc(t)}{dt}$.

Reaction stoichiometry and kinetics The second part, the change in concentration, is by assumption only caused by the reactions. A reaction is formally described by the stoichiometry and the reaction kinetics. The stoichiometry gives the proportions in which the concentrations are changed [Heinrich and Schuster, 1996, p. 13]. A reaction j has stoichiometric coefficients $S_{i,j} < 0$ if the concentration of species i is decreased and $S_{i',j} > 0$ if the concentration of i' is increased. We say that reaction j consumes i and produces i'. In enzymatic reactions the same amount of enzyme e that enters the reaction is also emitted by the reaction and the stoichiometric

coefficient is hence $S_{e,j} = 0$. For the whole reaction system the stoichiometry is comprehended in the stoichiometric matrix $(S_{i,j})_{i=1,\dots m,\, j=1,\dots n} = S \in \mathbb{R}^{m\times n}$. Most reactions have integer stoichiometric coefficients $S_{i,j}$. However, there are also metabolic processes which cannot be given by integer stoichiometry, as e.g. oxidative phosphorylation. The reactions are represented by the columns of S, while the rows correspond to the species. The dynamics of the system depend on how fast each reaction occurs, that is, how fast the changes in the composition of the species, given by the stoichiometry, are happening. This velocity is called reaction rate (or just "rate" or "flux rate") and with the assumption that the rate depends only on the current composition of the species, we can express the reaction kinetics as a function $v_j\colon \mathbb{R}^m \to \mathbb{R}$ of the concentrations, such that $v_j(t) = v_j(c(t))$ is the rate of reaction j at time t. All flux rates together give a so-called flux distribution (or flux vector) $v(t) \in \mathbb{R}^n$. The meaning of the flux distribution is that the concentrations of the species changes at time point t by

$$\frac{dc(t)}{dt} = Sv(t). \tag{2.1}$$

The rates $v(t)$ may be given by mass action kinetics, see [Heinrich and Schuster, 1996, Sec. 2.2.1] or [Beard and Qian, 2008, p. 45f]. Equation (2.1) is then a system of ordinary differential equations (ODEs). Note that most reactions are reversible. The stoichiometric matrix defines one direction of each reaction as positive. If the reaction occurs in the opposite direction, this is reflected by a negative flux rate.

Enzymes In metabolic networks almost all reactions are enzymatic, but the enzymes themselves are often neglected in the model. In the stoichiometric matrix S they do not appear, because their concentration is not affected by the reactions and the stoichiometric coefficients are thus 0. In fact, the concentration of enzymes is controlled from outside of the metabolic network by transcriptional regulation. This offers a starting point to integrate gene regulation into a metabolic network model [Covert et al., 2001, Oyarzún et al., 2009, Bartl et al., 2013]. While enzyme kinetics can have a complex non-linear form, the dependence on the enzyme concentration is always linear, see [Beard and Qian, 2008, Sec. 4] and [King and Altman, 1956], so that integration of this aspect can be done without explicitly modelling the reaction kinetics, as we will see in the second chapter.

Steady-state assumption The essential step from the kinetic ODE model of metabolism to a constraint-based model is the so-called **steady-state assumption**. Metabolic reactions take place on a very fast time scale, compared to growth, environmental changes and also transcriptional regulation or signalling [Varma and Palsson, 1994a]. The dynamics of the kinetic ODE model often leads to a steady state. We can then assume that the system subsists in this state, waiting for changes in the environment or inputs from the regulatory and the signalling network, see [Heinrich and Schuster, 1996, p. 38f] or [Fell et al., 1986]. As a result, the metabolic network would be in different steady states for longer time periods which are interrupted by short dynamic events. The transition from one steady state to another is caused by regulatory processes and environmental changes. The postulation of such a dynamical behaviour justifies focusing the study of metabolic networks on the phases of steady state and to neglect the short phases of dynamic transitions. From the mathematical point of view (2.1) becomes then

$$Sv = 0. \tag{C1}$$

The steady state flux distribution v describes the metabolism during a phase with unspecified duration between events which affect the metabolic network from outside.

Input and output of the network From the thermodynamical point of view, the biochemical reaction system satisfying (C1) with $v \neq 0$ must be in a non-equilibrium state [Beard and Qian, 2008, p. 44]. This means that constantly some molecules are entering the system and some are leaving the system, see also Sec. 2.1.3. For a metabolic network these molecules are for example substrates as glucose, other carbon sources and oxygen. The metabolism is taking up the substrates and turns them into biomass components and by-products which leave the system. The computational analysis of metabolic networks is often focused on this conversion [Durot et al., 2009, Reed et al., 2003, Soh and Inoue, 2010, Genrich et al., 2001], in particular in the field of metabolic engineering [Bailey, 1999, Burgard et al., 2003]. These substrate metabolites as well as the biomass and by-products are not species of the formally defined reaction system, in particular they are not represented in the stoichiometric matrix and not subject to the steady-state assumption. However, they are an important part of the model and we will call them **external metabolites**. There are some distinguished reactions in the system, which are taking up the substrates or produce the biomass (-components) or by-products. These reactions will be called **exchange reactions**. All metabolites represented in the stoichiometric matrix are called **internal metabolites**. The exchange reactions connect external with internal metabolites.

A metabolic network model might intentionally miss some parts of the metabolism. This means that some metabolites are in reality connected to reactions which are not represented in the model. Such metabolites will also be considered as external, because their concentration is determined by processes which are not part of the model (missing reactions) and hence the steady-state assumption does not apply to them.

To represent the input and the output formally, we will use a **stoichiometric uptake matrix**, denoted by $S^{\#}$ and a **stoichiometric output matrix**, denoted by $S^{\star}$. These matrices represent all external metabolites. In particular, $S^{\#}v$ tells us the consumption of the external substrates and $S^{\star}v$ the production of biomass components and by-products. If an exchange reaction is reversible, the corresponding external metabolite can be substrate as well as product of the metabolic network and it is represented in $S^{\star}$ as well as in $S^{\#}$.

2.1.1 Flux cones and flux spaces

Constraint-based modelling of metabolism can be seen as a top-down approach where a set of possible flux distributions is identified by expressing global properties in the form of constraints [Smallbone et al., 2007, Price et al., 2004a]. One such property is the steady-state assumption giving the linear constraints (C1). More constraints are given by the irreversibility of certain reactions. An irreversible reaction is one that can only proceed in the positive direction given by the stoichiometric matrix. In other words, the irreversible reactions can only have non-negative flux rates. Let $J^{Irr} \subset \{1, \ldots, n\}$ be the index set of all reactions that are irreversible, then we have the linear inequalities

$$v_j \geq 0 \text{ for } j \in J^{Irr}. \tag{C2a}$$

So far, two different biologically motivated constraints on steady state flux distributions were introduced, namely (C1) and (C2a). They can be seen as a system of linear inequalities for the flux vector $v \in \mathbb{R}^n$ (the equalities $Sv = 0$ can be represented by the inequalities $Sv \leq 0$ and

$-Sv \leq 0$). The solution set of this homogeneous system of linear inequalities is a polyhedral cone (cf. p .14). It will be called the **flux cone**, denoted by

$$\mathcal{C} := \{v \in \mathbb{R}^n : Sv = 0,\ v_j \geq 0,\ j \in J^{Irr}\}.$$

If bounds on the flux rates of the individual reactions are given by $lb \in \mathbb{R}^n_{\leq 0}$ for the lower bounds and $ub \in \mathbb{R}^n_{\geq 0}$ for the upper bounds, we get the constraints

$$lb \leq v \leq ub. \tag{C2b}$$

The irreversibility constraints can be included here by setting $lb_j = 0$ for $j \in J^{Irr}$, so that (C2a) follows from (C2b). The constraints (C1) and (C2b) define the polytope

$$\mathcal{F} = \{v \in \mathbb{R}^n : Sv = 0,\ lb \leq v \leq ub\}.$$

This polytope is the set of flux distributions which are biologically feasible according to the model. In general, any set of constraints on flux distributions defines a subset of $\mathbb{R}^n$. This set contains all flux distributions which are biologically feasible according to the model and will be called the **flux space**. The elements of the flux space will be called **flux modes**. By $\mathcal{F}$ we will denote flux spaces of different kinds, while $\mathcal{C}$ is exclusively used for the flux cone given by (C1) and (C2a).

Optimising a biological objective Constraint-based modelling is often used in concert with optimisation. The constraints define a subset $\mathcal{F} \subset \mathbb{R}^n$ of flux distributions which are feasible in terms of the biological model. For optimisation we have to derive an objective function $f\colon \mathbb{R}^n \to \mathbb{R}$ from the biological model. If $\mathcal{F}$ is given by the linear constraints (C1), (C2b) and we have a linear objective function $f(v) = c^\top v$ with $c \in \mathbb{R}^n$, this gives the following LP (cf. p. 15)

$$\begin{aligned} &\text{maximise } c^\top v, \text{ such that} \\ &Sv = 0 \\ &lb \leq v \leq ub, \\ &\text{with variables } v \in \mathbb{R}^n \end{aligned} \tag{2.2}$$

which is known as **flux balance analysis** (FBA) [Varma and Palsson, 1994b,Fell et al., 1986, Edwards et al., 2002a]. The objective function refers to a biological objective of the modelled organism. Compared to the constraints, which are theoretically based on physical limitations, the objective is a rather hypothetical concept. Evolutionary pressure is often seen as a possible origin of biological objectives [Feist and Palsson, 2010, Fell et al., 1986]. Once a biological objective is identified, it has to be formulated in terms of flux rates. A typical objective is the maximisation of the flux rate through a biomass reaction. Usually, this is a reaction which has been inserted artificially into the network in order to represent the consumption of metabolites for biomass composition [Reed et al., 2003].

Since LPs can be solved very efficiently, FBA can be applied to genome-scale networks. Apart from the classical FBA given by (2.2), many more or less similar methods were presented, which can be seen as extensions or variations of FBA [Covert et al., 2008,Lee et al., 2008,Jensen et al., 2011, Mahadevan et al., 2002, Covert et al., 2001, Beg et al., 2007, Navid and Almaas,

2012]. Most of the variations include other biological aspects by additional constraints or a different objective function. The steady-state assumption (C1) and the irreversibility constraints (C2a) are the basic constraints which most of the variations of FBA have in common and which are also used in metabolic pathway analysis [Schilling et al., 2000, Schuster et al., 2000]. In general, we can say that the flux cone $\mathcal{C}$ corresponding to these two constraints constitutes the basis for constraint-based modelling. In metabolic pathway analysis the cone itself is the object of analysis, while approaches that use optimisation, as e.g. FBA and its variations, require a bounded flux space. This bounded flux space can be obtained from the flux cone by intersecting it with a bounded set $\mathcal{B} \subset \mathbb{R}^n$. In the case of FBA, $\mathcal{B}$ is the box $\{v \in \mathbb{R}^n : lb \leq v \leq ub\} = \prod_{j=1}^{n}[lb_j, ub_j]$. The solution set of the system of the linear inequalities (C1) and (C2a) is a polyhedral cone, while the linear inequalities (C1) and (C2b) define a polytope (or a polyhedra in case some reactions are unbounded, i.e. $ub_j = \infty$ or $lb_j = -\infty$). Such a representation of polyhedra is called outer description and is well suited for optimisation. The results of FBA and of optimisation problems in general are single elements of the flux space. To get a global view of the flux space, the approach of metabolic pathway analysis [Schilling et al., 2000, Schuster et al., 2000] is providing a comprehensive description of the whole flux space by 'generating' elements. This is also called an 'inner' (or 'parametric') description and will be discussed in the following section. An alternative strategy to grasp the geometry of a bounded flux space is the sampling of flux modes, see Sec. 9.2.1.

2.1.2 Generating elements and elementary modes of the flux cone

The outer description of a polyhedron as the solution set of a system of linear inequalities, i.e., $P := \{x \in \mathbb{R}^n : Ax \leq b\}$ with $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ allows testing easily if $y \in P$ for some $y \in \mathbb{R}^n$. On the other hand, an inner description of the polyhedron can be useful to understand the structure of the polyhedron and is necessary if we want to generate arbitrary elements $y \in P$. Every polyhedral cone C has an inner description based on a finite set of elements $x^1, \ldots, x^k \in C$. We say that $x^1, \ldots, x^k \in C$ **generate** the cone C, if $C = \mathrm{cone}(x^1, \ldots, x^k)$. Similarly, every polytope P is generated by a finite number of elements $x^1, \ldots, x^k \in P$ in the sense that $P = \mathrm{conv}(x^1, \ldots, x^k)$ [Grötschel et al., 1993, p. 10]. These subsets of generating elements $\{x^1, \ldots, x^k\}$ are called **generating subsets**. In the other direction, given a finite number of points in $\mathbb{R}^n$, the convex hull is always a polytope, while the conic hull is always a polyhedral cone [Schrijver, 1998, Corrolary 7.1a,c]. A polyhedral cone C has thus the outer representation $C = \{x \in \mathbb{R}^n : Ax \leq 0\}$ and furthermore the inner representation by generating elements $x^1, \ldots, x^k$ namely $C = \{y \in \mathbb{R}^n : y = X\lambda,\ \lambda \in \mathbb{R}^k_{\geq 0}\}$, where X is the matrix whose columns are the generating elements, i.e., X is defined by $X_{\cdot,i} := x^i$, $i = 1, \ldots, k$. Similarly, if a polytope is given by $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ the inner description is given by $P = \{y \in \mathbb{R}^n : y = X\lambda,\ \lambda \in \mathbb{R}^k_{\geq 0}$ and $\sum_{i=1}^{k} \lambda_i = 1\}$. If $b = 0$, the pair (A, X) of the two matrices of the two representations is called a double description pair [Fukuda and Prodon, 1996]. The size of one description (size of the matrix A or X) can be exponential in the size of the other.

Elementary modes and decomposability The unbounded flux cone is studied in the field of metabolic pathway analysis, where the goal is to identify possible pathways through the metabolic network. An important concept is to define finite sets of flux modes that give a comprehensive representation of all possible pathways. In a mathematical sense, this is achieved as soon as these flux modes generate the flux cone. Most prominent are the extreme rays [Price et al.,

2002] and the elementary flux modes [Schuster and Hilgetag, 1994], [Klamt and Stelling, 2003]. The latter will be introduced below. Also other related concepts were introduced [Wagner and Urbanczik, 2005, Larhlimi and Bockmayr, 2009].

Definition 2.1. An element $v \in \mathcal{C} \setminus \{0\}$ is an **elementary mode** (EM) if there exists no $x \in C \setminus \{0\}$ with $\mathrm{supp}(x) \subsetneq \mathrm{supp}(v)$.

By definition, an EM is a flux mode v, such that there is no other flux mode using a proper subset of the reactions which v is using. It turns out that the flux distribution of an EM v is already defined by its support $\mathrm{supp}(v)$, because there is (up to scaling) no other flux mode except v or $-v$ with the same support [Schuster et al., 2002a], i.e., $\mathrm{supp}(x) = \mathrm{supp}(v) \Rightarrow x = \lambda v, \lambda \in \mathbb{R} \setminus \{0\}$, if v is an EM. The **elementary modes (EMs)** of a flux cone $\mathcal{C}$ are a generating subset [Klamt and Stelling, 2003]. Algorithms for the enumeration of all EMs of a flux cone $\mathcal{C} = \{x \in \mathbb{R}^n : Sx = 0, x_j \geq 0 \text{ for } j \in J^{Irr}\}$ were presented together with implementations as METATOOL [Pfeiffer et al., 1999] and based on the double description method [Fukuda and Prodon, 1996] as efmtool [Terzer and Stelling, 2008, efmtool]. These algorithms compute all EMs simultaneously. This is a drawback, because the number of EMs (and the number of intermediate flux vectors that have to be stored during computation) is often so large that it causes the computation to fail and in this case not a single EM is obtained. An interesting alternative is therefore the approach to compute EMs individually by solving MILPs [De Figueiredo et al., 2009]. In this method an MILP is formulated which iteratively searches for a flux mode v minimising the number of used reactions ($\min \sum_i b_i$, where $b_i = 1$ if $v_i \neq 0$ and $b_i = 0$ if $v_i = 0$).

Definition 2.2. A non-zero element $v \in \mathcal{C}$ (or $v \in \mathcal{F} \subset \mathcal{C}$) is called **decomposable**, if there exist $x, y \in \mathcal{C} \setminus \{0\}$ (resp. $x, y \in \mathcal{F} \setminus \{0\}$) and scalars $\lambda, \mu > 0$ such that $\lambda x + \mu y = v$ and $\mathrm{supp}(x), \mathrm{supp}(y) \subsetneq \mathrm{supp}(v)$.

Proposition 2.3. Let $\mathcal{C}$ be a flux cone and $x \in \mathcal{C}$, then x is non-decomposable if and only if it is an elementary mode.

This is a basic result from pathway analysis [Schuster et al., 2002a]. It is not true in general for $x \in \mathcal{F} \subset \mathcal{C}$, instead non-decomposable elements of $\mathcal{F}$ can be a proper superset of the elementary modes, see e.g. [Goldstein and Bockmayr, 2013]. In Sec. 2.2.4 we will give a sufficient condition on $\mathcal{F}$, such that Prop. 2.3 still holds. Decomposability in $\mathcal{F}$ will play an important role in Chap. 3.

2.1.3 Futile cycles

An aspect which is totally ignored in the models of metabolism based on (C1) and (C2a),(C2b) is the second law of thermodynamics [Beard et al., 2002] which is telling us that "a reaction can only occur in the direction of decreasing Gibbs free energy". Gibbs free energy depends among other factors on the metabolite concentrations, which are completely undetermined in FBA. However, even without knowing the concentrations, certain flux modes can still be excluded. Since Gibbs free energy must be descending in the direction of the fluxes, a direct consequence is that the flux distribution cannot contain cycles [Beard and Qian, 2008, p. 44] which are called **futile cycles** in this context.

To include these thermodynamic constraints in a constraint-based model, additional integer variables indicating the signs of the flux rates are needed and the resulting MILP is of

much higher complexity (cf. Sec. 6.2) than the usual LP of FBA. A method of the COBRA toolbox [Schellenberger et al., 2011a, Schellenberger et al., 2011b] implements these thermodynamic constraints and is thus excluding futile cycles. Incorporating thermodynamic laws into FBA gives a more realistic model of metabolism and often improves the obtained results, as e.g. in [Hoppe et al., 2007] where ranges for realistic enzyme concentrations were retrieved from literature and used to include thermodynamic constraints in an FBA model. Since the thermodynamic properties of a reaction determine how the interplay of enzyme concentration and metabolite concentration establishes the flux rate (cf. Sec. 9.4), constraint-based thermodynamic models can also be used to identify reactions that are subject to allosteric rather than transcriptional regulation [Kümmel et al., 2006].

Similar to the general FBA, also the related method of flux variability analysis (FVA), which will be introduced on p. 29, can be further refined if thermodynamic constraints are considered. Technically, this is done by extending the LP to an MILP. An algorithm that can handle such thermodynamic constraints efficiently and performs FBA and FVA on genome-scale models is given in [Müller and Bockmayr, 2013].

2.2 The bounded flux space

After some remarks about the biological interpretation of bounds we go back to the classical FBA problem (2.2) and discuss the role of redundant constraints in (C2b). Redundancy has to be understood as a mathematical property of the flux space and it is important to keep in mind that for the definition of the biological model, all $2n$ bounds lb, ub are essential. The representation of the flux space can be simplified by discarding redundant bounds from the system of linear inequalities, but then the biological features of the network are not correctly represented anymore. This will be illustrated in Chap. 5, where the effect of perturbations of the bounds is examined. Finally, more general bounds than in FBA are considered in this section in order to be prepared for the discussion in the following chapter.

2.2.1 Biological meaning of bounds on the flux space

Different biological aspects can be implemented by individual bounds on the flux rates or, more generally, by a bounded set $\mathcal{B} \subset \mathbb{R}^n$, such that $\mathcal{F} = \mathcal{C} \cap \mathcal{B}$. To apply FBA, it is necessary to introduce bounds on the flux cone, otherwise the problem can be unbounded and it is impossible to obtain optimal solutions in this case. The problem of insufficient biological data is sometimes circumvented by fixing all bounds to an arbitrary value [Reed et al., 2003], i.e., $ub_j = \alpha$, for $\alpha > 0$, $j = 1, \ldots, n$ and $lb_j = 0$ if $j \in J^{Irr}$, otherwise $lb_j = -\alpha$. In other cases a bounded flux space can be obtained by bounding the uptake of substrates [Edwards et al., 2002b]. From the biological point of view, the flux rates of the enzymatic reactions are among other factors limited by the turnover number of the corresponding enzyme. The turnover number gives the maximal number of molecules of substrate that one enzyme molecule can convert to the product(s) per second [Alberts et al., 1998, p. 159]. This maximal turnover is attained if the substrates are saturated. Multiplied with the amount of enzymes, the turnover number gives the maximal

velocity of a reaction. It can therefore be used to set bounds on the individual reaction rates. Taking into account the limited total amount of enzymes, it is also reasonable to introduce a global bound on the sum of all flux rates [Beg et al., 2007]. This approach will be further discussed in Chap. 3.

The units for metabolites and flux rates Metabolic pathway analysis [Schilling et al., 2000, Schuster et al., 2000] is based on the flux cone, where the absolute magnitude of the flux rates is left unspecified. On the other side, FBA and similar optimisation-based approaches are introducing an absolute measure of the flux rates by the bounds and this allows using chemical units in the model. Metabolite concentrations are usually given in [mol/gDW] (mol per gram dry weight) and flux rates in [(mol/gDW)/h] (mol per hour and gram dry weight). Biological data can give a maximal uptake rate of a certain substrate in [(mol/gDW)/h], which can serve as an upper bound on the reaction rate. The consumption of substrates in the steady state v is given by $S^{\#}v$ and the production by $S^{*}v$ in [(mol/gDW)/h] ($S^{\#}$, S^{*} being the stoichiometric matrices for consumption resp. production, introduced above). If a duration τ [h] of the phase of this steady state is given, the amount of substrate taken up in this phase is exactly $\tau S^{\#}v$ [mol/gDW] and the amount of produced biomass components and by-products is $\tau S^{*}v$, [mol/gDW].

Maintenance constraints Yet another aspect that can be expressed by bounds on the flux rates are vital functions of the cell which are fulfilled by specific reactions in the metabolic network. These indispensable functions must be permanently fulfilled and it is hence not reasonable to assume that they can have zero flux in a steady state. In other words, a certain minimum flux through such a reaction j is a vital requirement. This can be implemented by adding a positive lower bound (or a negative upper bound, in case the vital function requires the reaction to have a permanent flux in negative direction). Such a constraint, which forces the flux through a reaction to be non-zero, will be called **maintenance constraint** and will always be considered separately from the bounds lb, ub. In particular, all lower bounds are non-positive and all upper bounds non-negative, $lb \leq 0 \leq ub$. In contrast to ub, lb, maintenance constraints are rather defining a biological objective than limitations of the metabolic network.

2.2.2 Redundant bounds

Consider the bounded flux space $\mathcal{F} = \{v \in \mathbb{R}^n : Sv = 0,\ lb \leq v \leq ub\}$. Every reaction is bounded individually, but due to the steady-state assumption $Sv = 0$, bounding one reaction is also limiting the flux rate through other reactions. As a consequence, some constraints $lb_j \leq v_j \leq ub_j$ can be redundant as illustrated in Fig. 2.1. Sometimes, bounding a single uptake reaction already leads to a bounded flux space, see e.g. Fig. 2.2.

Formal definition of redundant constraints For the mathematical treatment of the model, redundant constraints can be discarded. The concept of redundancy applies to any polyhedron $P = \{v \in \mathbb{R}^n : Av \leq b\}$, where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. To define redundancy formally, we will use the geometric interpretation of P as an intersection of halfspaces, i.e., $P = \bigcap_{i=1}^{m} H_i$, where $H_i := \{v \in \mathbb{R}^n : A_{i\cdot}v \leq b_i\}$. If one halfspace $H_{i'}$ is missing in this intersection, we get the polyhedron $P^{\neg i'} := \{v \in \mathbb{R}^n : A_{i\cdot}v \leq b_i \text{ for all } i \neq i'\}$. This polyhedron is either larger and contains P, or it coincides with P. In the latter case, the constraint $A_{i'\cdot}v \leq b_{i'}$ can be discarded from the representation of P. Such constraints are called **redundant** [Schrijver, 1998, p. 100]. Geometrically,

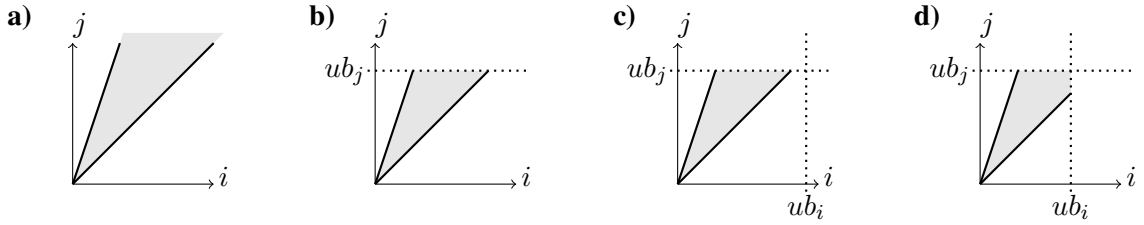


Figure 2.1: Projection of a flux cone onto two reactions i, j. a) The unbounded cone tells us that both reactions are irreversible $v_i, v_j \geq 0$ and the ratio is limited to $v_i \leq v_j \leq 3v_i$. b) Introducing an upper bound ub_j on j implies also a bound on the flux v_i. c) Adding an upper bound $ub_i \geq ub_j$ does not restrict the flux space further, ub_i is a redundant bound in this case. d) Only if $ub_i < ub_j$, the flux space will be further reduced.

redundancy of the constraint $A_{i'}.v \leq b_{i'}$ means that the halfspace $H_{i'}$ contains $P^{\neg i'}$. Now we want to distinguish two cases of redundancy. Consider the hyperplane $\overline{H}_{i'} := \{v \in \mathbb{R}^n : A_{i'}.v = b_i\}$. In case $\overline{H}_{i'} \cap P^{\neg i'} = \varnothing$, the constraint $A_{i'}.v \leq b_i$ is called **strongly redundant** and otherwise **weakly redundant**. This distinction will play an important role in Chap. 5. In the case of metabolic networks, where the flux space is bounded by the constraints $lb \leq v \leq ub$, usually many of the corresponding $2n$ bounds are redundant. Bounding constraints that are generic are either non-redundant or strongly redundant (this property is called **generic** [Griffiths and Harris, 2011, p. 20], because a randomly chosen bound would have it almost surely, in other words, it can only be avoided by a deliberate and exact choice of the bound). Weakly redundant constraints only occur if the values of some bounds are chosen deliberately to be in a certain relation (e.g. $ub_i = ub_j$ in Fig. 2.1). This is e.g. the case if the bounds are obtained by flux variability analysis, see below. As follows directly from the definition, discarding one redundant constraint does not change the flux space. Hence, the representation of a polyhedral flux space as a set of inequalities is equivalently given by a subset of these inequalities obtained by discarding one redundant inequality. It should be noted that only one redundant constraint may be discarded at one time, because weakly redundant constraints can become non-redundant as soon as another redundant constraint is discarded, see Fig. 2.2 for an example.

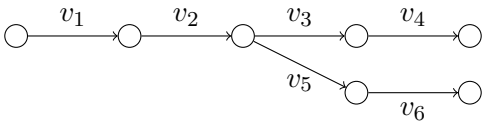


Figure 2.2: In this small example all reactions are irreversible, i.e., $lb = 0$. There are three pairs of fully coupled (Def. 2.5) reactions: $\{v_1, v_2\}$, $\{v_3, v_4\}$ and $\{v_5, v_6\}$. From each of these pairs at least one reaction has a redundant upper bound, so that altogether at least three upper bounds are redundant. In case $ub_1 = ub_2 = ub_3 = ub_4 = ub_5 = ub_6$, we have the situation that all upper bounds are weakly redundant, but of course we cannot discard all of them. It would be admissible to discard the bounds on reactions $1, 2$ for example. If additionally ub_3 is discarded, the bound ub_4 becomes non-redundant. At maximum we can delete all bounds except ub_1 or ub_2, because for $i = 1$ or $i = 2$ the constraint $v_i \leq ub_i$ suffices to bound the whole flux space with the bounds $ub_1 = ub_2 = ub_3 = ub_4 = ub_5 = ub_6$.

Flux variability analysis Different possibilities to compute all redundant constraints of an LP are reviewed in [Paulraj and Sumathi, 2010]. One of the simplest methods is flux variability analysis (FVA), originally introduced to explore the space of optimal solutions in FBA [Mahadevan and Schilling, 2003]. It amounts to testing one after the other for each reaction i' the two bounding constraints $v_{i'} \leq ub_{i'}$ and $v_{i'} \geq lb_{i'}$ for redundancy. Formally, the procedure can be described as follows:

- Solve the optimisation problems to find $v_{i'}^+ := max\ v_{i'}$ and $v_{i'}^- = min\ v_{i'}$ subject to $Sv = 0$ and $lb_i \leq v_i \leq ub_i$ for all $i \neq i'$.
- Compare the objective value $v_{i'}^+$ with $ub_{i'}$ and $v_{i'}^-$ with $lb_{i'}$. If $v_{i'}^+ \leq ub_{i'}$ or $v_{i'}^- \geq lb_{i'}$, the respective constraint is redundant. If $v_{i'}^+ = ub_{i'}$ resp. $v_{i'}^- = lb_{i'}$ the constraint is weakly redundant, otherwise strongly redundant.

If we discard one redundant constraint in each step and proceed with the reduced set of constraints, we end up with a subset of constraints which are all non-redundant and this subset represents the same polytope as the complete set of inequalities. Since only one constraint is deleted at one time and the next constraint is tested without all previously deleted constraints, we are sure that the represented polytope stays the same. This would not be the case if first all constraints are checked and afterwards all redundant constraints are discarded, see for example Fig. 2.2. If for some reaction i' the bound $ub_{i'}$ turns out to be redundant, i.e. $v_{i'}^+ \leq ub_{i'}$, then we know that the flux rate $v_{i'}$ is in fact bounded by $v_{i'}^+$, which is a weakly redundant constraint. This gives new bounds, which will be called stoichiometric bounds and denoted as

$$sub_i := \min(v_i^+, ub_i), \quad slb_i := \max(v_i^-, lb_i).$$

Independent of this construction by FVA, we give the following characterisation.

Definition 2.4. Bounds on flux rates are called **stoichiometric** and denoted as slb, sub if for each reaction j there exist $v^+, v^- \in \mathcal{F}$ with $v_j^+ = sub_j$ and $v_j^- = slb_j$.

The stoichiometric bounds are tighter, $sub \leq ub$ and $slb \geq lb$, but the flux space does not change if we substitute lb, ub with slb, sub, respectively. From a computational point of view the representation of the flux space by slb, sub contains the information of the flux variability, i.e., sub_j, slb_j give exactly the maximal resp. minimal flux rate that reaction j can have in this network. A stoichiometric bound is either non-redundant or weakly redundant. Stoichiometric bounds represent mathematical properties of the constraint-based model and can be useful for computational purposes, but they do not represent flux rate limitations based on biological parameters as the turnover numbers or other properties of the corresponding enzymes.

Redundancy and flux coupling In the context of redundancy, the concept of flux coupling [Burgard et al., 2004, Larhlimi and Bockmayr, 2006] should be shortly mentioned. Flux coupling and redundancy are two different concepts, but are also connected to each other. While flux coupling is a relation between two reactions and refers to the unbounded flux cone, redundancy is a global property of the FBA flux space and depends on the bounds as well as the stoichiometry of the whole network.

Definition 2.5. Let $\mathcal{C}$ be a flux cone. Two reactions i and j are **fully coupled**, if $v_i = \lambda v_j$ for fixed $\lambda \neq 0$ and all flux modes $v \in \mathcal{C}$, **partially coupled** if $v_i \neq 0 \Leftrightarrow v_j \neq 0$ for all $v \in \mathcal{C}$, and i is **directionally coupled** to j if $v_i \neq 0 \Rightarrow v_j \neq 0$ for all $v \in \mathcal{C}$. If none of these relations hold (in any of the two directions) then i, j are called **uncoupled**.

A coupling relation that is valid in the flux cone $\mathcal{C}$ holds evidently also in any subset of $\mathcal{C}$, in particular in a bounded flux space $\mathcal{F} = \mathcal{C} \cap \mathcal{B}$. In the FBA case, where $\mathcal{B} = \{v \in \mathbb{R}^n : lb \leq v \leq ub\}$, coupling relations can imply some obvious redundancies: If the reactions $i_1, \ldots, i_k$ are fully coupled, an arbitrary subset of $k-1$ of the bounds $lb_{i_1}, \ldots, lb_{i_k}$ can be discarded as well as $k-1$ of the bounds $lb_{i_1}, \ldots, lb_{i_k}$ without changing the flux space. In the other direction, we can have redundancies in a network without any flux coupling. The reason is that the flux coupling only considers pairwise relations, while redundancies can emerge from interdependencies between more than two reactions.

2.2.3 Bounding the whole network by constraints on substrate uptake

The bounds are strongly determining the flux space and hence also the outcome of optimisation over this flux space. However, setting the precise values lb, ub for the bounds requires detailed biological knowledge and quantitative data. While the stoichiometry and irreversibility is often available for large, even genome-scale networks, it is rarely possible to acquire enough reliable data to equip the model with bounds as well. Therefore, it can be an interesting alternative to focus on exchange reactions that take up substrates. This is only a small subset of reactions and data about substrate uptake is often available. In many cases bounds on the uptake reactions already suffice to bound the whole network [Edwards et al., 2002b]. Since a thermodynamically feasible steady state flux distribution is only possible if the system has input and output [Beard and Qian, 2008, p. 44], the bound on the uptake is bounding every thermodynamically feasible flux distribution. If only the uptake reactions are bounded, this means that the availability of substrates is the only limiting factor in the metabolic network model, i.e., no reaction except for the uptake is ever rate limiting. In the case that only one uptake reaction u is given as well as in the case that only the sum of the uptake by different reactions $u \in U$ is bounded, the mathematical structure of such a model is very simple. The resulting flux space is then the cone $\mathcal{C} = \{v \in \mathbb{R}^n : Sv = 0, v_j \geq 0 \text{ for all } j \in J^{Irr}\}$ intersected with just one halfspace H_u, defining the bound on the single uptake flux, i.e., $H_u := \{v \in \mathbb{R}^n : v_u \leq ub_u\}$ or the bound on the sum of the uptake fluxes $H_U := \{v \in \mathbb{R}^n : \sum_{u \in U} v_u \leq ub_U\}$, respectively.

2.2.4 Non-linear and non-convex bounds on the flux space

The biological limitations on flux distributions are in reality more complex and cannot be expressed by independent bounds $lb_j \leq v_j \leq ub_j$ on the individual reactions $j = 1, \ldots, n$ alone. Formally, this means that we have to consider different bounds on the flux space and we will therefore introduce the general notion of a bounding set $\mathcal{B} \subset \mathbb{R}^n$ and define the flux space as

$$\mathcal{F} = \mathcal{C} \cap \mathcal{B}.$$

The bounding set $\mathcal{B}$ can be a polytope (for example if the sum of reaction rates is bounded), but also non-linear and non-convex. In the case of FBA it is $\mathcal{B} = \prod_{j=1}^{n} [lb_j, ub_j]$. In general, some conditions should always be satisfied by $\mathcal{B}$ in order to represent the intended biological meaning. Since the bounds $\mathcal{B}$ are supposed to represent limitations on the flux rates, we do not expect them to enforce flux. Formally, if $v \in \mathcal{F} = \mathcal{C} \cap \mathcal{B}$ then for any $\lambda \in [0, 1]$ we should also have $\lambda v \in \mathcal{F}$. In particular this means that maintenance constraints (p. 28) are not expressed as part of the bounding set $\mathcal{B}$. To fulfil this condition, the bounding set $\mathcal{B}$ is required to be star shaped with center 0. Star shaped sets are a generalisation of convex sets, the following definition is taken from [Königsberger, 1997, p. 193].

Definition 2.6. A region $B \subset \mathbb{R}^n$ is called star shaped with center $z \in B$, if for every $x \in B$ and $\lambda \in [0,1]$ the point $\lambda z + (1-\lambda)x$ is element of B.

Furthermore, if we interpret the bounding set as a quantitative specification of the flux cone that preserves its information about possible pathways, it is desirable that for some fixed $\lambda > 0$, every flux distribution $v \in \mathcal{C}$ has a contracted counterpart $\lambda v \in \mathcal{F}$.

Definition 2.7. Let $\mathcal{C} \subset \mathbb{R}^n$ be a flux cone, then $\mathcal{F} \subset \mathcal{C}$ is a called a **conic sprout of** $\mathcal{C}$ if it can be expressed as the intersection of $\mathcal{C}$ and a set $\mathcal{B}$ which is star shaped with center 0 and if there exists an $\varepsilon > 0$, such that all $x \in \mathcal{C}$ with $\|x\| \leq \varepsilon$ are also element of $\mathcal{B}$.

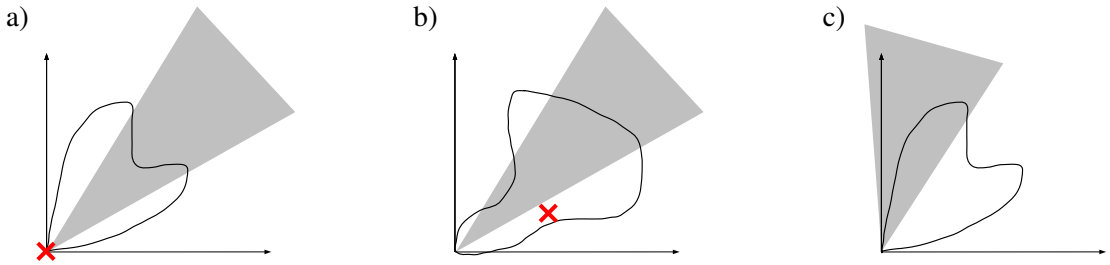


Figure 2.3: An illustrative example of a 2-dimensional unbounded cone indicated by the shaded area and the boundary of the bounding set $\mathcal{B}$ drawn in black. In **a)** the bounding set is star shaped with zero as center. **b)** This bounding set is also star shaped with the center marked by a cross. However, apparently it is not a star shaped set with zero as center and therefore the intersection with the cone is not a conic sprout by Def. 2.7. **c)** The bounding set is the same as in a), but the sketch indicates that contrary to a) the intersection of this cone with the bounding set is not a conic sprout.

In a conic sprout the information about possible pathways and flux coupling is exclusively determined by the flux cone, while the bounding set is only quantifying the possible fluxes through the pathways. This property eases the theoretical treatment of the flux space which then inherits the generators from the cone. Let $\{x^1, \ldots, x^k\}$ be a generating subset of $\mathcal{C}$. If $\mathcal{F}$ is a conic sprout of $\mathcal{C}$, then there is an $\varepsilon > 0$, such that $\varepsilon \cdot \{x^1, \ldots, x^k\} \subset \mathcal{F}$. In particular, every $y \in \mathcal{F}$ can be written as $y = \sum_{i=1}^k \lambda_i x^i = \sum_{i=1}^k (\varepsilon^{-1}\lambda_i)(\varepsilon x^i)$ with $\lambda_i \geq 0$, $i = 1 \ldots, l$. We conclude, that if the elements $x^1, \ldots, x^k \in \mathcal{C}$ are a generating subset of $\mathcal{C}$, then they can be scaled to a generating subset of any given conic sprout of $\mathcal{C}$.

Decomposability and EMs The definitions of decomposability and EMs refer to the flux cone $\mathcal{C}$, but can be equally given for a bounded flux space by replacing $\mathcal{C}$ with $\mathcal{F}$ in the definitions. For any flux space $\mathcal{F} = \mathcal{C} \cap \mathcal{B}$ we have the following implications which follow directly from the inclusion $\mathcal{F} \subset \mathcal{C}$.

$$x \in \mathcal{F} \text{ is } \begin{cases} \text{an EM of } \mathcal{C} \\ \text{non-decomposable in } \mathcal{C} \end{cases} \quad \Longrightarrow \quad x \text{ is also } \begin{cases} \text{an EM of } \mathcal{F} \\ \text{non-decomposable in } \mathcal{F}. \end{cases}$$

Let $\mathcal{F}$ be a conic sprout of $\mathcal{C}$ and $x, y \in \mathcal{C} \setminus \{0\}$ with $\operatorname{supp}(y) \subsetneq \operatorname{supp}(x)$, i.e., x is not an EM of $\mathcal{C}$. These elements can be scaled down to $\varepsilon\, x, \varepsilon\, y \in \mathcal{F}$ which still fulfil the strict inclusion of the

supports. Similarly, a decomposition $v = \lambda x + \mu y \in \mathcal{C}$ as in Def. 2.2 can also be scaled down to a decomposition $\varepsilon\, v = (\varepsilon\, \lambda)\, x + (\varepsilon\, \mu)\, y$ with $\varepsilon\, v,\ (\varepsilon\, \lambda)\, x,\ (\varepsilon\, \mu)\, y \in \mathcal{F}$ and preserving the strict inclusion of the supports. Therefore the above implications become equivalences as soon as $\mathcal{F}$ is a conic sprout, i.e.,

$$x \in \mathcal{C} \text{ is } \begin{cases} \text{an EM of } \mathcal{C} \\ \text{non-decomposable in } \mathcal{C} \end{cases} \quad \Longleftrightarrow \quad \exists \varepsilon > 0 : \varepsilon\, x \in \mathcal{F} \text{ is } \begin{cases} \text{an EM of } \mathcal{F} \\ \text{non-decomposable in } \mathcal{F}. \end{cases}$$

From these equivalences it follows directly that Prop. 2.3 holds also for conic sprouts, i.e., the EMs and the non-decomposable elements of a conic sprout coincide. In the case that we have at least one maintenance constraint, i.e., $v_j \geq mb_j > 0$, it is easy to see that the equivalence does not hold. Just assume that y is an EM in $\mathcal{C}$, $y_j = 0$ and there is an $x \in \mathcal{F}$, with support $\mathrm{supp}(x) = \mathrm{supp}(y) \cup \{j\}$. Then x is not an EM in $\mathcal{C}$, but due to the maintenance constraint it has minimal support in $\mathcal{F}$ and is thus an EM in $\mathcal{F}$.

2.3 The objective function

To perform mathematical optimisation over the flux space, an objective function has to be defined which represents biological objectives of the cell. Translating a biological objective into a mathematical objective is not at all straightforward and different ways have been proposed in the literature. In particular the minimisation of time needed to produce certain minimum amounts of biomass is discussed here, since this objective will be used in the following chapters.

2.3.1 Choice of an objective function and sensitivity of the optimisation

The formulation of an objective function is a difficult task, because first of all it is not evident what the metabolic objectives of the organism under consideration are and second, once a biological objective has been designated it has to be translated into precise numeric values that determine the objective function. The problem of finding an appropriate objective function appears similarly in constraint-based modelling and in dynamic optimisation, see e.g. [Oyarzún et al., 2009]. The choice of an objective function is often justified a posteriori by results which are in good agreement with experimental data. A systematic comparison of different objective functions for FBA and related optimisation problems is given in [Schuetz et al., 2007]. In linear programming the optimal solutions can be sensitive to small changes in the numeric values of the linear objective function, as is illustrated in Fig. 2.4.

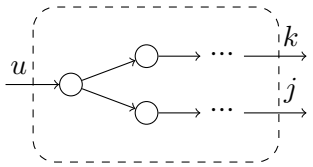


Figure 2.4: Example network with two separated pathways that produce output by reactions k and j from one input that is taken up by reaction u. Assume that ub_u is the only non-redundant upper bound. Let $\max\ c^\top v$ be the objective with $c_i = 0$ except for $c_j, c_k > 0$. In the case $c_j = c_k$, an optimal solution is given by using only the top pathway with reaction k. The objective value is then $ub_u \cdot c_k$. An arbitrarily small change in c can result in $c_k < c_j$ and in this case the only optimal solution would be the bottom pathway to j. Such a situation is not generic (cf. p. 29) in the sense that it only occurs, if the ratio of the entries takes a precise value, here $c_j/c_k = 1$. A randomly chosen $c \in \mathbb{R}^n$ would not satisfy this condition almost surely. However, bounds that were derived artificially, e.g. with FVA or by setting them all to an equal number, are likely to result in such a non-generic situation. In general, the optimal solutions only change if c is perturbed with sufficient amplitude.

2.3.2 Different types of objectives in dynamic and constraint-based modelling

Biological objectives are often stated in terms of metabolites that are demanded to be produced or consumed. In constraint-based models the production of metabolites is given per time unit by the flux rates v. Let S^* be the stoichiometric output matrix (p. 23) and I^* the index-set of the output metabolites. By definition, all $i \in I^*$ denote external metabolites which are not subject to the steady-state assumption. The amounts of produced output per time unit is then given by S^*v. A biological objective specifying the relative requirements of the output metabolites can be given by $\Gamma \in \mathbb{R}^{I^*}$ and the optimisation $\max\ \Gamma^\top S^* v$.

In other situations the biological objective might be directly formulated in terms of flux rates. Consider for example the objective to maximise energy production, represented by reaction j'. In this case the model would be $\max\ c^\top v$, with $c_{j'} = 1$ and $c_j = 0$ for $j \neq j'$. Formally, there is no difference between objectives that refer to metabolites or to flux rates. In both cases, the objective function can be given by $c \in \mathbb{R}^n$, $\max\ c^\top v$.

The objective of time minimisation Objectives of the kind $\max\ c^\top v$, as in FBA, are not always appropriate. Their shortcoming is that the target reactions j with $c_j \neq 0$ cannot be distinguished. If we have $c_i = \lambda c_j$, then a flux rate v_j through reaction j, contributes the same to the objective value as a flux of $v_i = \lambda v_j$ through reaction i. Consider a realistic biological objective that requires the production of minimum amounts of certain metabolites. In this case, the objective would be to minimise the time τ needed to achieve these requirements. Instead of $\max\ \Gamma^\top S^* v$ we would then get the constraint $S^* \tau v \geq \Gamma$ and the objective function $\min\ \tau$. In general, objectives of time minimisation can be given by the constraints $C(\tau v) \geq \Gamma$ with a matrix $C \in \mathbb{R}^{l \times n}$ and a demand vector $\Gamma \in \mathbb{R}^l$. It can be assumed that $\Gamma > 0$, i.e., every component of Γ is strictly positive. The non-negativity is not a restriction, since it can always be achieved by multiplying the corresponding rows of C with -1. If $\Gamma_i = 0$, then $C_{i,\cdot}(\tau v) = 0$ implies that either $C_{i,\cdot} v = 0$ or $\tau = 0$. The case $\tau = 0$ can be excluded as a feasible solution in this modelling context and we conclude that the objective with $\Gamma_i = 0$ would result in a constraint that is independent of the time τ and is therefore not part of the objective function.

To show the similarity to dynamic optimisation (see e.g. [Klipp et al., 2002, Oyarzún et al., 2009]), we will formulate the optimisation of this objective first as a dynamic optimisation problem. The flux distribution is in this case a function of time (e.g. a solution of a system of ODEs). The objectives discussed above can then be formulated as follows:

$$\max \int_0^\tau c^\top v(t)dt \tag{O1}$$

which is the maximisation of the weighted sum of the turnover $\int_0^\tau v_j(t)dt$, $j = 1, \ldots, n$, over a given time interval $[0, \tau]$. Alternatively, we can minimise the time needed to achieve a given requirement with

$$\min \tau, \text{ s.t. } \int_0^\tau Cv(t)dt \geq \Gamma. \tag{O2}$$

In the case of constraint-based modelling we replace $v(t)$ by the constant steady state flux distribution v and the constraints are given by $v \in \mathcal{F}$. The objective function (O1) is then independent of the duration τ, we just maximise $c^\top v$. In (O2) the integral is replaced by the factor τ and the duration τ is minimised. In fact, (O2) is then a generalisation of the FBA objective (O1), because for the one dimensional $C = c^\top$ and $\Gamma = 1$, (O2) becomes $c^\top v \geq \tau^{-1}$ and as a consequence, minimising τ is equivalent to maximising $c^\top v$. The production of minimum amounts of *different* target metabolites in minimal time cannot be appropriately expressed in (O1), see Fig. 2.6. Therefore, we will deal with the general form (O2) in the following chapters. The limitations of FBA that were discussed above can be circumvented by an artificial extension of the metabolic network, which turns out to be equivalent to the generalisation (O2). One part of this extension is the inclusion of an artificial biomass reaction, a construction which is often used in the application of FBA. The extensions of the network and the relation to different objectives are illustrated in Figs. 2.5 and 2.6.

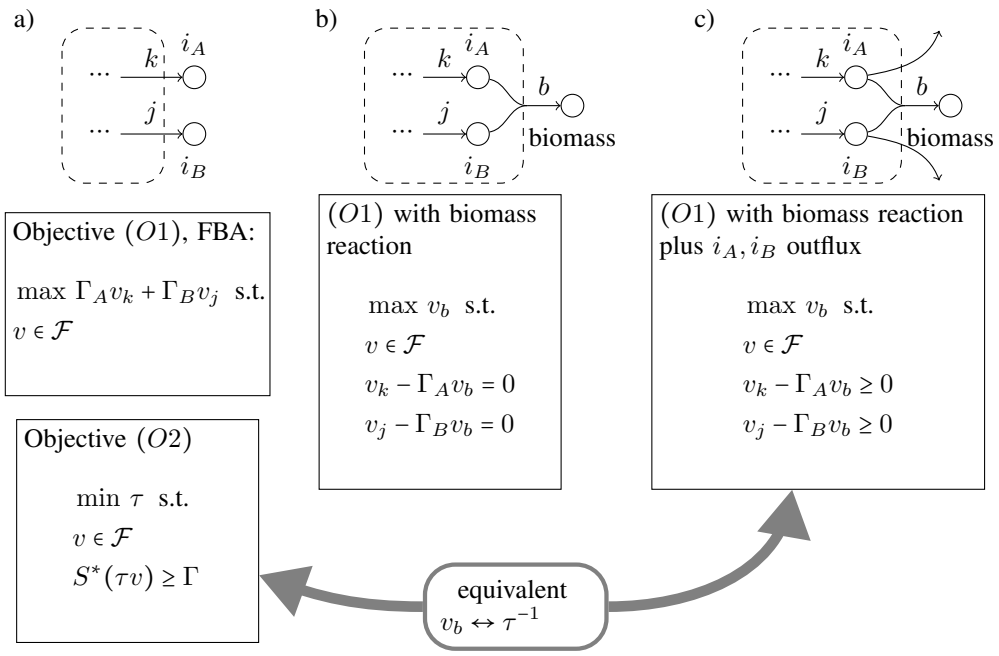


Figure 2.5: **a)** An example network, where only the output of external metabolites i_A and i_B by the exchange reactions k resp. j is specified. The biological objective is to produce in minimal time at least the amount Γ_A of metabolite i_A and Γ_B of i_B. This can be expressed in the form (O2) as $\min \tau$ s.t. $S^*(\tau v) \geq \Gamma := (\Gamma_A, \Gamma_B)^\top$ and $v \in \mathcal{F}$, where S^* is the stoichiometric $2 \times n$ matrix of the two external metabolites i_A, i_B . Assume that the production of i_A, i_B corresponds exactly to the flux rate of reactions k resp. j. This means that $S^*_{1,k} = S^*_{2,j} = 1$ are the only non-zero entries of S^* and we have the equivalence $S^*(\tau v) \geq \Gamma \Leftrightarrow \tau v_k \geq \Gamma_A,\ \tau v_j \geq \Gamma_B$. The FBA model with the objective Γ would be to maximise $\Gamma_A v_k + \Gamma_B v_j$, but this does not distinguish the two target reactions and will not give the desired optimal solution, see Fig. 2.6. **b)** If we extend the network by an artificial biomass reaction, the target metabolites i_A and i_B become internal metabolites and the steady-state assumption applies, forcing them to the ratio $v_k/v_j = \Gamma_A/\Gamma_B$. **c)** By further adding individual output fluxes for i_A and i_B, flux rates of k, j are not fixed to the ratio of the corresponding demand Γ_A, Γ_B anymore. This FBA model is in fact equivalent to the time minimisation (O2) in the original network a). To see this, substitute v_b with τ^{-1} to get the constraints (O2) and note that τ^{-1} is maximal if and only if τ is minimal.

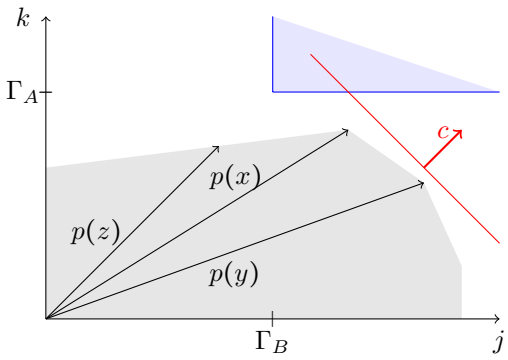


Figure 2.6: A hypothetical flux space for the unspecified example in Fig. 2.5 a), projected onto the reactions k and j. The objective (O2) can be depicted by the blue shaded region $\{x \in \mathbb{R}^2 : x_1 \geq \Gamma_A,\, x_2 \geq \Gamma_B\}$. The optimisation searches for a $v \in \mathcal{F}$, such that the projection $p(v)$ comes close to this region, in the sense that $\tau p(v) \geq \Gamma$ for minimal τ. The unique projection of an optimal solution would be $p(x)$ in this example. The red hyperplane with the normal vector c represents the objective function of FBA, where $c_k = \Gamma_A$ and $c_j = \Gamma_B$ and $c_i = 0$ for $i \neq k, j$. In this case $p(y)$ would be the projection of an optimal solution, although it is not optimal for satisfying $\tau y_k \geq \Gamma_A$ and $\tau y_j \geq \Gamma_B$ with minimal τ. The reason is that in this optimisation the two target reactions are not distinguished and only the total flux rate $c^\top v$ is measured. The extension by an artificial biomass reaction b and without individual output reactions, corresponding to Fig. 2.5 b), would fix the flux rates z_k, z_j to the ratio of the demands Γ_A, Γ_B. If we maximise the biomass production z_b, the projection of an optimal solution would be $p(z)$.

2.3.3 Conventions for notation and formulation of the optimisation problems

In the example of Fig. 2.5, each row of S^* had only one non-zero entry, because the external target metabolites i_A, i_B were produced by individual exchange reactions, as it is usually the case. In general, we will always assume here that in the time minimisation (O2) each row of C contains only one non-zero entry. The constraint $C(\tau v) \geq \Gamma$ can then be written componentwise $C_{h,i_h}\tau v_{i_h} \geq \Gamma_h$ for every row h of C, such that i_h gives index of the unique non-zero entry in row h. The positive direction of the reactions can be chosen freely (by multiplying the corresponding column of the stoichiometric matrix with -1). Therefore we can assume that $C_{h,i_h} > 0$. By replacing Γ_h with $C_{h,i_h}^{-1}\Gamma_h$ we can then assume that $C_{h,i_h} = 1$, for all rows h. The corresponding reactions i_h will be called **target reactions** $t \in T$, where T is the set of all target reactions. The constraint of (O2) is thus expressed in the inequalities

$$v_t \geq \Gamma_t \quad \text{for target reactions} \quad t \in T.$$

Equivalent formulations of the optimisation problem With this convention the computation of an optimal steady state flux distribution subject to (O2) gives the following optimisation problem.

$$\begin{aligned} &\min \tau, \text{ subject to:} \\ &v \in \mathcal{F} \ (\Leftrightarrow Sv = 0 \text{ and } v \in \mathcal{B}) \\ &\tau v_t \geq \Gamma_t \text{ for } t \in T \\ &\text{with variables:} \\ &\tau \in \mathbb{R}_{\geq 0},\ v \in \mathbb{R}^n \end{aligned} \tag{OP1}$$

In the simplest case we would have $v \in \mathcal{B} :\Leftrightarrow lb \leq v \leq ub$ and the quadratic term τv_t, where two variables are multiplied, can be easily avoided by substituting the variables $w_j := \tau v_j$ and solve the LP

$$\begin{aligned} &\min \tau, \text{ subject to:} \\ &Sw = 0 \text{ and } \tau\, lb \leq w \leq \tau\, ub \\ &w_t \geq \Gamma_t \text{ for } t \in T \\ &\text{with variables:} \\ &\tau \in \mathbb{R}_{\geq 0},\ w \in \mathbb{R}^n \end{aligned} \tag{OP1a}$$

Alternatively, it is also possible to shift the duration variable τ into the demand vector Γ by using the equivalences $\tau v_t \geq \Gamma_t \Leftrightarrow v_t - \tau^{-1}\Gamma_t \geq 0$, $t \in T$. Denoting the inverse by $\delta := \tau^{-1}$, we get the LP

$$\begin{aligned} &\max \delta, \text{ subject to:} \\ &Sv = 0 \text{ and } lb \leq v \leq ub \\ &v_t - \delta\Gamma_t \geq 0 \text{ for } t \in T \\ &\text{with variables:} \\ &\delta \in \mathbb{R}_{\geq 0},\ v \in \mathbb{R}^n \end{aligned} \tag{OP1b}$$

If δ^* is an optimal solution of (OP1b), then $\tau^* := (\delta^*)^{-1}$ is an optimal solution of (OP1). In case $\delta^* = 0$ this means that no $v \in \mathcal{F}$ was found with $v_t > 0$ for all $t \in T$ (since $\Gamma_t > 0$ for all $t \in T$). As a consequence the requirement $\tau v \geq \Gamma$ cannot be fulfilled for any τ, i.e., (OP1) is not feasible.

3 Modelling enzymatic resource allocation and alterations in metabolism

3.1 Overview

Most reactions in a metabolic network are enzymatic, which means that the maximal flux rates are among other factors determined by the amount of active enzymes for the specific reactions. Constraint-based models, as e.g. FBA, often neglect this aspect and assume high abundance of all enzymes. This is an unrealistic assumption, since the required resources for enzymes are limited, which holds for the building blocks, the amino acids, as well as for the ribosomes, which assemble the enzymes. The production of enzymes and therewith the distribution of the limited resources is controlled by the gene regulatory network. A metabolic network model can be enriched by taking into account the enzyme concentrations and the control by gene regulation. Some approaches are reviewed in the next section. The remainder of this chapter is then dedicated to the derivation and theoretical analysis of a specific model of resource allocation in enzyme production, which will then be used to investigate how alterations in metabolism can enhance the efficiency of metabolic output production. This modelling approach was developed in cooperation with Hermann-Georg Holzhütter and Sascha Bulik [Palinkas et al., 2015]. Gene expression is modelled in an *on-off* manner by binary variables. All available amino acids are distributed among the activated reactions which are competing for this resource. This coarse regulation of metabolism by binary gene expression is intended to reflect the relationship between gene expression and flux rates as e.g. reported in [Chubukov et al., 2013]. Based on our model, an optimisation problem is formulated to find a sequence of flux modes, such that given amounts of required metabolic output are produced in minimal time. Each flux mode in the sequence comes from the same flux space, which is in general not convex due to the binary implementation of gene expression. It turns out that with a non-convex flux space a sequence of different flux modes can perform better than a single flux mode. The flux space is then investigated w.r.t. those elements that can constitute such an optimal sequence. Under some conditions, which exclude maintenance constraints, finite subsets of the flux space can be identified, such that an optimal solution can always be composed from those elements. They are closely related to and can be obtained by enumeration of non-decomposable elements of a related flux space.

3.2 Enzymes in constraint-based models of metabolism

Including the enzymes into a model of metabolism is often done with focus on the limited amount of amino acids. A consequence is that the sum of the maximal flux rates of all reactions is limited, which can easily be included into a constraint-based model. The production of enzymes controlled by genes is also a means of the cell to decide on the transcriptional level which pathways are used and to change the flux distribution. In constraint-based models, the concentration of active enzymes for a reaction is implicitly included in the bounds on the reactions, since the maximal flux rate depends linearly on the enzyme concentration. Integration of regulatory aspects into the model can hence be implemented with these bounds. Various existing approaches are discussed here before a new modelling approach is further developed in the next section.

The bounds on the flux rates in constraint-based models are usually given by $lb \leq v \leq ub$. If we take a look at the kinetics of a metabolic reaction, we see that the flux rate is determined by the concentrations of all reactants and also by the concentration of the enzyme that catalyses that specific reaction.

Role of enzyme concentration in reaction kinetics The mathematical description of the kinetics depends on the number of reactants and differs for reversible and irreversible reactions. Furthermore, the kinetics of the reaction system is different, depending on whether it operates close to thermodynamic equilibrium or not [Qian and Beard, 2005]. In a well-mixed biochemical system, a reversible reaction $\mathrm{A} \underset{k_b}{\overset{k_a}{\rightleftharpoons}} \mathrm{B}$, can be described by mass action kinetics [Heinrich and Schuster, 1996, Sec. 2.2.1]. The change in concentration is then given by $\frac{dA}{dt} = k_a[A] - k_b[B]$, where $[A]$, $[B]$ denotes the concentration of A, B, respectively, and k_a, k_b are constants. An enzymatic reaction $\mathrm{A} \overset{k}{\rightarrow} \mathrm{B}$, catalysed by enzyme E, is in fact a sequence of two reactions $A + E \rightharpoonup C \rightharpoonup B + E$, where C denotes the enzyme complex. If we write this as $\mathrm{A} \overset{k}{\rightarrow} \mathrm{B}$, the rate k is not constant and hence cannot be described by mass action kinetics. But the explicit reaction scheme $A + E \rightharpoonup C \rightharpoonup B + E$ can be modelled by mass action and this gives a system of ordinary differential equations (ODEs). According to these ODEs, the concentration of the enzyme complex will be approximately in a steady state if the enzyme is saturated, i.e., the substrate concentration $[A]$ is high compared to the enzyme concentration $[E]$. For this case, the kinetics of the overall flux v from A via C to B can be described by the Michaelis-Menten equation

$$v = kc^+[E]\frac{[A]}{K + [A]}, \tag{3.1}$$

where kc^+ is the turnover number and K a constant parameter [Beard and Qian, 2008, p. 73]. As the substrate concentration $[A]$ increases, v is approaching the maximum rate $v_{max} = kc^+[E]$. In the constraint-based approach the concentrations of metabolites are not specified at all. Instead there are given upper bounds on the reaction rates. Since almost all reactions in a metabolic network are enzymatic, the v_{max} values would be a natural choice for the bounds. While (3.1) only refers to the simplest kind of an irreversible reaction $A \rightharpoonup B$, reactions with several reactants or

reversible reactions have more complex rate laws, see e.g. [King and Altman, 1956] or [Beard and Qian, 2008, Sec. 4]. However, these rate laws have in common that a non-linear term contains the concentrations of all reactants and this term is multiplied with the enzyme concentration $[E]$. In other words, the flux rate depends linearly on the enzyme concentration. In fact, the maximal flux rate of an enzymatic reaction is also in general given by $v_{max} = kc^+[E]$. The notation kc^+ indicates that the turnover number refers to the positive direction of the reaction as given by the stoichiometric matrix. In case of a reversible reaction, kc^- denotes the turnover number of the opposite direction. In the following, the enzyme concentration for reaction j is denoted E_j [mol/gDW] (mol per gram dry weight). Under a constant concentration E_j and the assumption of saturated enzyme, the bounds on reaction j can thus be given by $lb_j = -kc_j^- E_j$ and $ub_j = kc_j^+ E_j$. Here we will always assume that $kc_j^+ > 0$ for all reactions $j = 1, \ldots, n$. This excludes the case $kc_j^+ = kc_j^- = 0$ and assumes that all irreversible reactions are oriented in the positive direction.

Enzyme concentration in constraint-based models Based on these enzyme concentrations, the bounding constraint (C2b) from Chap. 2 is then given by

$$-kc_j^- E_j = lb_j \leq v \leq ub_j = kc_j^+ E_j \quad j = 1, \ldots, n. \tag{3.2}$$

A direct consequence is that the limited capacity of the cell to build enzymes translates to a linear constraint on the flux rates. The amount of resources needed for the synthesis depends on the individual enzyme. The limited availability of resources can be represented by the linear constraint $\sum_{j=1}^n q_j E_j \leq 1$, where the weights q_j specify the amount of resources needed for the production of enzyme j. Let $kc_j := \max(kc_j^+, kc_j^-) > 0$, so that (3.2) implies the weaker bounds $|v_j| \leq kc_j E_j$. The constraint $\sum_{j=1}^n q_j E_j \leq 1$ for limited resources then implies the following weaker constraint which is piecewise linear.

$$\sum_{j=1}^{n} q_j (kc_j)^{-1} |v_j| \leq 1 \tag{3.3}$$

This kind of constraint, known as molecular crowding, was shown to improve the standard FBA, in the sense that the computed flux distributions are in better concordance with experimental results. For example, in [Beg et al., 2007] the model was able to capture adaptation to a changing environment. In [Shlomi et al., 2011a], FBA with a constraint similar to (3.3) was applied to explain phenotypes of fast proliferating cancer cells. Also for the theoretical analysis of the relationship between maximal yield (output per uptake) pathways and maximal output pathways, an enzyme capacity constraint was used [Schuster et al., 2011]. A dynamic optimisation approach was used in [Klipp et al., 2002, Oyarzún et al., 2009] to model optimal time courses of enzyme concentrations $E_j(t)$ during the activation of a pathway. These two approaches are based on the limited total amount of enzymes, similar to (3.3), and the flux rates depend linearly on the current amount of specific enzymes. In [Goelzer et al., 2011], the synthesis of enzymes was included in a kinetic model, which implies in particular a limited total enzyme amount, but is a much more detailed description of enzyme synthesis in a metabolic network. What all these approaches have in common, is the linear dependency of the flux rates on the enzyme concentration combined with the limited capacity for enzyme synthesis. Furthermore, all methods optimise over a convex solution space to find an optimal steady state flux distribution or an optimal time course of enzyme concentrations. Optimal means in [Beg et al., 2007, Shlomi et al.,

2011a, Schuster et al., 2011] the maximisation of output or yield. In [Klipp et al., 2002, Oyarzún et al., 2009] the time needed for pathway activation is minimised as well as the amount of enzymes that are used during this process. In [Goelzer et al., 2011] optimality refers to the growth rate. The computation of constant flux distributions can be used to model a dynamic behaviour of the network by computing a sequence of consecutive flux distributions [Covert et al., 2001]. This means that the time is discretised in intervals and the flux distribution is constant on each interval. Such an approach will be developed in this chapter. The synthesis of the enzymes is not incorporated in our method.

Enzyme concentration and gene expression On a coarse scale, the enzyme concentration is controlled by the expression level of the corresponding genes. However, the activity of enzymes is also controlled internally by the metabolic network. Enzymes are controlled by allosteric regulation and can be inactivated by phosphorylation or in the case of product inhibition by metabolites [Alberts et al., 1998, p. 170-175]. Genes are not directly involved in these mechanisms. For this reason, gene expression can only be seen as a coarse control of the concentration of active enzymes. On the gene regulatory level, decisions for activation or shut down of distinct parts of the metabolic network are made. This transcriptional control determines which enzymes will be available and which not, while the precise concentration level of the different active enzymes is further adjusted within the metabolic network. Based on this relationship, major changes in the flux distribution can be expected to depend on the dynamics in the expression of the genes, while small changes in the magnitude of some fluxes are more likely to be controlled by other means. It was shown [Chubukov et al., 2013] that the expression levels of the metabolic genes do not correlate to the flux rates of the corresponding reactions in a majority of cases. Only large fold changes in flux were found to be regulated on a transcriptional levels in an *on-off* manner. However, the precise adjustment of flux rates to achieve metabolite homeostasis could not be explained by transcriptional regulation.

Optimal resource allocation by a sequence of flux distributions An attempt to incorporate this coarse on-off regulation on the transcriptional level into a constraint-based modelling approach will be presented now. Starting with a general formulation, some properties of the corresponding flux spaces and optimal solutions will be derived. We will then focus on a specific model of resource allocation controlled by binary gene expression. This specific model is applied to a core metabolic network in Chap. 4. The approach is based on the computation of flux modes by optimisation. The objective is the minimisation of production time, (O2). The regulation of the metabolism on the transcriptional level enters in the flux bounds. The available proteins for enzyme synthesis are not freely distributed, but controlled by gene expression. The aim of the modelling approach is to reproduce possible alterations in metabolism. Therefore, sequences of flux distributions are computed. The use of different flux modes consecutively can lead to a reduction in production time, which means a gain in efficiency. These predicted alterations are not triggered by an external or an internal event, but emerge merely from the optimisation of the resource allocation to achieve the output production in minimal time.

3.3 A model of resource allocation for the catalysation of reactions

All enzymes in the metabolic network are produced by ribosomes from amino acids. Both are limited resources in the cell. The total amount of amino acids in particular gives hard bounds on the total enzyme amount in the cell. The presented model will be based on this limitation and translate it into bounds on the reaction rates. All expressed metabolic genes compete for amino acids to produce their enzyme. The concentration of a specific enzyme depends on parameters, i.e., the molecular mass and the rate of synthesis and degradation.

We assume that each reaction $j \in \{1, \dots, n\}$ is catalysed by exactly one enzyme and the concentration of that enzyme is regulated by one distinct gene. Formally we have (3.2), where $E_j = E_j(g)$ depends on $g \in [0,1]^n$ and g_j is the expression level of the gene corresponding to reaction j. If the gene is fully expressed we have $g_j = 1$ and $g_j = 0$ if no enzyme at all is produced. Of course, this one-to-one relationship between genes and reactions is not likely to be valid in the whole metabolic network. Some exceptions can still be included into this model. For example, if a reaction is not considered to be regulated in the model, then g_j can be fixed to a certain value that gives the desired constant bounds for the reaction. In case that two reactions i, j are catalysed by the same enzyme, this can be included with the constraint $g_i = g_j$. In general, we can define a set $G \subset [0,1]^n$ of gene expression states, where such restrictions are manifested.

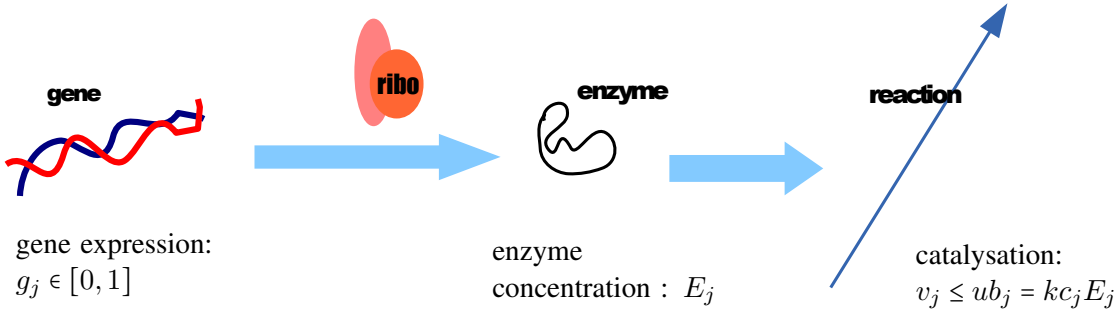


Figure 3.1: Scheme of the model, where gene expression leads to production of enzymes that catalyse a reaction. The maximal flux rate increases linearly with the enzyme concentration.

Dependence of enzyme concentration on gene expression A constitutive feature of our approach is the competition between all expressed genes for amino acids necessary to synthesize the enzymes. The expressed genes also compete for the ribosomes that are needed to produce the enzymes and the availability of ribosomes can even be the limiting factor in transcription [Chu et al., 2011]. Both limitations, for the amino acids and for the ribosomes, result in a limited total amount of amino acids that can be synthesised to enzymes. We assume that in the process of synthesis and degradation of proteins, the mass fraction A [mol/gDW] of free amino acids is constant. It serves for synthesis of new enzymes. The total mass of enzymes in the cell is then $A_{tot} - A\gamma_A$ [g/gDW], where A_{tot} is the total mass of available amino acids and $A \cdot \gamma_A$ is the mass

of free amino acids which is estimated with help of the average molecular weight $\gamma_A = 126$ [Da] (= 126 [g/mol]) of one amino acid.

The time-dependent variation of the enzyme amount E_j [mol/gDW] can be expressed in a simplified manner as the resultant of synthesis and degradation:

$$\frac{\mathrm{d}E_j}{\mathrm{d}t} = g_j\ ks_j\ A\ -\ kd_j\ E_j.$$

The resulting steady-state concentration of the j-th enzyme is

$$E_j = \frac{g_j\ A\ ks_j}{kd_j} \tag{3.4}$$

with ks_i representing an overall rate of protein synthesis (including all regulatory steps between transcription and ribosomal translation) and the first-order rate constant kd_i for the degradation of the enzyme. Taking into account the free amino acids pool A, the total mass of amino acids can be given in terms of the concentrations E_j [mol/gDW] and the weights γ_j [Da] of the enzymes:

$$A_{tot} = A\gamma_A + \sum_i E_i\gamma_i \tag{3.5}$$

Using the relations (3.4) and (3.5), it follows that the amount of the j-th enzyme at steady-state is given by

$$E_j = g_j\ A_{tot}\ \frac{\eta_j}{\gamma_A + \sum_i g_i\gamma_i\eta_i} \approx g_j\ A_{tot}\ \frac{\eta_j}{\sum_i g_i\gamma_i\eta_i}\ , \tag{3.6}$$

where the parameter $\eta_j := \frac{ks_j}{kd_j}$ controls the amount of protein if the coding gene g_j is active and will be referred to as **expression efficiency**. With equation (3.6), the upper bounds on the flux rates depending on enzyme E_j are given for $j = 1, \ldots, n$ by

$$v_j\ \le\ ub_j(g) := A_{tot}kc_j^+ g_j\ \frac{\eta_j}{\gamma_A + \sum_i g_i\gamma_i\eta_i} \approx A_{tot}kc_j^+ g_j\ \frac{\eta_j}{\sum_i g_i\gamma_i\eta_i}. \tag{3.7}$$

Since the molecular weights γ_j of the enzymes are usually many orders of magnitude larger than γ_A, the approximation in (3.7) can be expected not to affect practical results of this modelling approach. In the application of the resource allocation model in the next chapter, the computations are done with and without the approximation. The numerical results exhibit no significant difference, thus confirming the eligibility of the approximation. Nevertheless we will consider both versions in the theoretical analysis of this chapter. Analogously to (3.7), the lower bounds are defined as

$$v_j \ge lb(g) := -A_{tot}kc_j^- g_j\ \frac{\eta_j}{\gamma_A + \sum_i g_i\gamma_i\eta_i} \approx -A_{tot}kc_j^- g_j\ \frac{\eta_j}{\sum_i g_i\gamma_i\eta_i}.$$

The turnover numbers are $kc_j^-,\ kc_j^+ \ge 0$ for the negative and the positive direction, respectively. For irreversible reactions we set $kc_j^- = 0$ and hence $lb_j(g) = 0$. As already mentioned we assume $kc_j^+ > 0$ for $j = 1, \ldots, n$. The turnover numbers are given in [1/h] and the flux rates are given in [(mol/gDW)/h].

Note that the molar amount A of free amino acids depends on the absolute magnitude of the gene expression, i.e., from (3.5), abbreviating $W := \sum_i g_i\gamma_i\eta_i$, we get

$$A_{tot} = A\gamma_A + A_{tot}\frac{W}{\gamma_A + W} \quad \Rightarrow \quad A = A_{tot}\gamma_A^{-1}\frac{\gamma_A}{\gamma_A + W}.$$

The magnitude of the gene expression determines the size of W. As W increases, the molar amount A decreases. In contrast, the approximated resource allocation model is not dependent on the absolute gene expression, but only on the relation between the expression levels of different genes, since A_{tot} is always completely distributed between the reactions.

3.3.1 Formal conventions for the theoretical analysis of the resource allocation model

The parameters of the model The theoretical analysis aims at understanding the flux space defined by the resource allocation and the optimisation problems defined on this flux space. The model has $4n+2$ parameters, namely the turnover numbers kc_j^-, kc_j^+, the molecular weights γ_j and the expression efficiencies η_j for $j = 1, \ldots, n$ as well as the total amino acid mass A_{tot} and the average weight of one amino acid γ_A. These parameters are representing the essential biological attributes of the metabolic network.

From the mathematical point of view, the parameter space of the model given by (3.7) has only $3n + 1$ dimensions. It is given by γ_A and the composed parameters $x_j := (A_{tot} kc_j^- \eta_j)$, $y_j := (A_{tot} kc_j^+ \eta_j)$ and $z_j := (\gamma_j \eta_j)$, so that (3.7) is

$$-x_j \frac{g_j}{\gamma_A + \sum_i g_i z_i} \;\leq\; v_j \;\leq\; y_j \frac{g_j}{\gamma_A + \sum_i g_i z_i}$$

for $j = 1, \ldots, n$ without the approximation (and if the approximation is made without γ_A).

In particular, the expression efficiencies do not add more dimensions to the model, because they can be incorporated in $kc_j^\pm$ and γ_j. Therefore, we will neglect them in the following theoretical analysis by setting $\eta_j = 1$, $j = 1, \ldots, n$. The resource allocation model (3.7) is then given by

$$-A_{tot} kc_j^- \frac{g_j}{\sum_i g_i \gamma_i} \leq v_j \leq A_{tot} kc_j^+ \frac{g_j}{\sum_i g_i \gamma_i}, \qquad j = 1, \ldots, n. \tag{3.8}$$

Only in the application of the resource allocation model in Chap. 4 the expression efficiencies will play a role again.

Special reactions In a given metabolic network there are most likely some reactions which will not be subject to the resource allocation described by (3.8). This is the case for reactions which are not catalysed by enzymes, as for example passive transport of O_2 and CO_2 (transport processes are not distinguished from actual reactions in the metabolic network model). Also artificial reactions for biomass production, energy consumption or other processes might be excluded. The bounds lb_j, ub_j of an excluded reaction j are constant, i.e., they do not depend on the gene expression pattern g. To exclude a reaction completely from the resource allocation model, we also have to set the corresponding enzyme weight to $\gamma_j = 0$. In the theoretical analysis we will neglect these excluded reactions and assume that all reactions are subject to resource allocation. In particular $\gamma_j > 0$ for all $j = 1, \ldots, n$. With this convention we can avoid many cumbersome and uninteresting case distinctions. All results would hold equally if some reactions were excluded from resource allocation, except for Sec. 3.4.3, where we additionally have to assume that the excluded reactions are not rate limiting.

Short notation for resource allocation To further simplify the notation and also to give a more general perspective on the resource allocation, we introduce a **resource function** $p : G \to [0,1]^n$ giving the share that each enzyme gets from the amino acids pool under the gene expression

pattern g. For reaction j this share is $p_j(g)$. The general resource allocation model is then given by

$$-A_{tot}kc_j^- p_j(g) \leq v_j \leq A_{tot}kc_j^+ p_j(g), \qquad j = 1\ldots,n, \tag{3.9}$$

which is exactly (3.8), if the above presented resource allocation model is used, i.e.,

$$p_j(g) = \frac{g_j}{\gamma_A + \sum_i g_i\gamma_i} \approx \frac{g_j}{\sum_i g_i\gamma_i}. \tag{3.10}$$

To fulfil the formal condition that $p_j(g) \in [0,1]$, it suffices to require $\gamma_j \geq 1$ for $j = 1\ldots,n$. This is surely fulfilled if γ_j is given in [Da]. In general, the resource function can also have another form, however some basic properties will be required for the theoretical analysis and will be indicated then.

3.4 The flux space of the resource allocation model

This section elucidates the essential properties of the flux space of the resource allocation model. The results in the following sections will mainly be based on these observations. Since the flux spaces that we consider here are star shaped with center zero, they can be completely described in terms of rays starting at zero. This perspective leads to a bijection from the non-linear flux space to the polytope that is obtained if the resource allocation model is neglected and only an FBA model remains. As we will see in the next section, the performance of optimal sequences of different flux modes compared to a single flux mode depends on the non-convexity of the flux space. It turns out here that this is directly dependent on the non-convexity of the gene expression space. Furthermore we will see that in case the gene expressions are completely unrestricted, the resource allocation constraints reduce to one piecewise linear constraint.

For the gene expression pattern $g \in G$, (3.9) defines the box

$$B_g := \prod_{j=1}^{n} [-A_{tot}kc_j^- p_j(g), A_{tot}kc_j^+ p_j(g)].$$

The union of these boxes over all $g \in G$ is the bounding set of the resource allocation model which we denote by

$$\mathcal{B}^* := \bigcup_{g\in G} B_g.$$

The flux space under a distinct gene expression state g is given by the intersection

$$\mathcal{F}_g = B_g \cap \mathcal{C},$$

where $\mathcal{C} = \{v \in \mathbb{R}^n : Sv = 0,\ v_j \geq 0,\ j \in J^{Irr}\}$ is the flux cone. Since G denotes the space of all gene expression patterns that are feasible in the model, the flux space for the model is given by

the union of these flux spaces or, equivalently, as the intersection of the cone with the bounding set of the resource allocation model.

$$\begin{aligned}\mathcal{F}^* &:= \{v \in \mathbb{R}^n \,:\, Sv = 0,\ \exists\, g \in G \text{ s.t. } -A_{tot} kc_j^- p_j(g) \le v_j \le A_{tot} kc_j^+ p_j(g),\ j = 1\ldots, n\} \\ &= \bigcup_{g\in G} \mathcal{F}_g \\ &= \mathcal{C} \cap \mathcal{B}^*.\end{aligned} \tag{3.11}$$

For the theoretical analysis in this chapter we have to assume that $\mathcal{B}^*$ is closed (cf. p. 15), which implies that also $\mathcal{F}^*$ is closed. Furthermore we have to require that for each reaction $j \in \{1,\ldots,n\}$ there exists a $g \in G$, such that $p_j(g) > 0$. This just means that every reaction in the model must be enabled by at least one of the feasible gene expression states of the model. This condition can always be fulfilled by deleting reactions from the network which cannot be enabled by any gene expression state.

3.4.1 Representation of the flux space with ray segments

To grasp the geometry of $\mathcal{F}^*$ we will relate it to the flux space of the network without considering resource allocation, i.e., the polytope

$$\mathcal{F} = \{v \in \mathbb{R}^n : Sv = 0,\ -A_{tot} kc_j^- \ \le\ v_j \ \le\ A_{tot} kc_j^+ \text{ for } j = 1,\ldots,n\}$$

with $-A_{tot} kc_j^- \le 0$ and $A_{tot} kc_j^+ > 0$ (since $kc_j^+ > 0$ is always assumed) for all $j = 1,\ldots,n$. Both flux spaces, $\mathcal{F}^*$ and $\mathcal{F}$, are based on the cone $\mathcal{C}$. To obtain $\mathcal{F}$ we intersect $\mathcal{C}$ with

$$\mathcal{B} = \prod_{j=1}^{n} [-A_{tot} kc_j^-, A_{tot} kc_j^+].$$

Note that the above defined bounding sets $\mathcal{B}_g$, $\mathcal{B}^*$ and $\mathcal{B}$ all contain 0. This follows directly from the definitions and the non-negativity of A_{tot}, kc_j^-, kc_j^+ and $p_j(g)$ for $j = 1,\ldots,n$. Furthermore, they are star shaped with center 0 as follows from

Lemma 3.1. *Let U_k, $V_h \subset \mathbb{R}^n$, with $k \in K$, $h \in H$ for arbitrary index sets K, H, be convex sets containing $z \in \mathbb{R}^n$. Then $(\bigcup_{k\in K} U_k) \cap (\bigcap_{h\in H} V_h)$ is star shaped with center z.*

This lemma is directly verified by applying the definitions of star shaped (Def. 2.6) and convex sets (p. 14). Also the flux spaces $\mathcal{F}^*$ and $\mathcal{F}$ are hence star shaped with center 0. This allows representing them in terms of the rays starting in 0. Let the subset $F \subset \mathbb{R}^n$ be closed, bounded and star shaped with center 0 (we are interested in the cases $F = \mathcal{F}^*$ and $F = \mathcal{F}$, they are both closed by assumption and bounded by definition). For any $v \in \mathbb{R}^n \setminus \{0\}$ let $\bar{v} = \lambda v$ for $\lambda \ge 0$, such that $\lambda v \in F$ and $\lambda' v \notin F$ for any $\lambda' > \lambda$. Such λ always exists because on the one hand we have $\lambda v = 0 \in F$ for $\lambda = 0$ and on the other hand the set $v \cdot \mathbb{R}_{\ge 0} \cap F$ is closed and bounded, since F is closed and bounded, and thus such a maximal element $\bar{v} = \lambda v \in F$ exists.

Definition 3.2. Let F be a closed and bounded flux space. An element $\bar{v} \in F$ is called **maximal** if $\lambda \bar{v}$ is not element of F for any $\lambda > 1$. The set of all maximal elements of F is denoted by M.

If zero is an interior point of F, then M is in fact the boundary of F. In general the boundary is a superset of M, see e.g. Fig. 3.2.

For $v \in \mathbb{R}^n \setminus \{0\}$ and F as above, we define the **ray-segment** of v in F to be the half open interval

$$(0, \bar{v}] := \{v \in \mathbb{R}^n : v = \lambda \bar{v},\ 0 < \lambda \leq 1\}.$$

Since F is star shaped with center 0, the ray segments are always contained in F. In case that $\bar{v} = 0$, the ray segment is defined to be the empty set $\varnothing \subset F$. By definition, every $v \in F \setminus \{0\}$ belongs to a ray segment. It is also clear that no v is contained in two ray segments. Concluding, F can be partitioned into $\{0\}$ and all ray segments.

Lemma 3.3. *Let $F \subset \mathbb{R}^n$ be closed, bounded and star shaped with center 0 and let M be the set of maximal elements, then we have the partition*

$$F = \{0\} \cup \bigcup_{\bar{v} \in M} (0, \bar{v}].$$

A bijection between the two flux spaces Note that every B_g, as defined above, is obtained from $\mathcal{B} = \prod_{j=1}^{n} [-A_{tot} kc_j^-, A_{tot} kc_j^+]$ by contracting along the j-th coordinate by $p_j(g) \in [0, 1]$. Consequently, we have the inclusion

$$\mathcal{F}^* = \mathcal{C} \cap \mathcal{B}^* = \mathcal{C} \cap \bigcup_{g \in G} B_g \subset \mathcal{C} \cap \mathcal{B} = \mathcal{F}. \tag{3.12}$$

For a given $v \in \mathcal{F} \setminus \{0\}$, let $(0, v^*]$ be the ray segment of v in $\mathcal{F}^*$. The inclusion (3.12) implies that $(0, v^*] \subset (0, \bar{v}]$, where $(0, \bar{v}] \neq \varnothing$ is the ray segment of v in $\mathcal{F}$, i.e., $v^* = \lambda^* \bar{v}$ with $0 \leq \lambda^* \leq 1$. Since $\bar{v}$ is not zero, λ^* is uniquely defined by $v^* = \lambda^* \bar{v}$ (if no element of $\mathcal{F}^*$ is in $(0, \bar{v}]$, we have $v^* = 0$ and hence $\lambda^* = 0$ by definition). Therefore, we can define a mapping

$$p' \colon \mathcal{F} \to [0, 1], \text{ by } p'(v) = \lambda^*, \text{ for } v \in \mathcal{F} \setminus \{0\}, \text{ and } p'(0) = 0. \tag{3.13}$$

Since p and p' are both describing the resource allocation model, they will both be called **resource function**. With the help of p' we define then a mapping

$$\phi \colon \mathcal{F} \to \mathcal{F}^*, \text{ by } \phi(v) = p'(v) \cdot v, \tag{3.14}$$

which contracts $\mathcal{F}$ to $\mathcal{F}^*$. Note that the resource function p' is constant on a fixed ray segment, i.e., if $\alpha > 0$ and $v, \alpha v \in \mathcal{F}$, then $p'(v) = p'(\alpha v)$. An element $v \in \mathcal{F}$ fulfils $Sv = 0$ and $-A_{tot} kc_j^- \leq v_j \leq A_{tot} kc_j^+$, $j = 1, \ldots, n$. The contracted element $w := \phi(v) = p'(v) \cdot v = p'(w) \cdot v$ still fulfils $Sw = p'(v) Sv = 0$ and is bounded by $-p'(w)\, A_{tot} kc_j^- \leq w_j \leq p'(w)\, A_{tot} kc_j^+$, $j = 1, \ldots, n$. It is hence clear that ϕ maps $\mathcal{F}$ to the set

$$F^\bullet := \{v \in \mathbb{R}^n : Sv = 0,\ -A_{tot} kc_j^- p'(v) \leq v_j \leq A_{tot} kc_j^+ p'(v),\ j = 1, \ldots, n\} \subset \mathcal{F}.$$

Since $0 \leq p'(v) \leq 1$ for all $v \in \mathcal{F}$ the bounds of $F^\bullet$ are tighter than those of $\mathcal{F}$ and therefore we have the inclusion of $F^\bullet$ in $\mathcal{F}$.

Lemma 3.4. *If $p'(v) > 0$ holds for every $v \in \mathcal{F} \setminus \{0\}$, then $\mathcal{F}^* = F^\bullet$.*

Proof. The elements of $F^\bullet$ are by construction contained in $\mathcal{F}^*$. It thus suffices to show that $\mathcal{F}^* \subset F^\bullet$. Since 0 is clearly contained in both sets, it can be omitted. Let $v \in \mathcal{F}^* \setminus \{0\}$. The maximal element in the ray segment of v in $\mathcal{F}$ is $\bar{v} = \lambda v$ for some $\lambda \geq 1$. Writing this

as $\lambda^{-1}\bar{v} = v \in \mathcal{F}^*$ shows that $\lambda^{-1} \leq \lambda^* = p'(v)$. This gives the following implications (the analogous lower bounds are omitted)

$$\begin{aligned} & \lambda v = \bar{v} \in \mathcal{F} \\ \Rightarrow \quad & \lambda v_j \leq A_{tot} k c_j^+, \quad j = 1, \ldots, n \\ \Rightarrow \quad & v_j \leq A_{tot} k c_j^+ \lambda^{-1} \leq A_{tot} k c_j^+ p'(v), \quad j = 1, \ldots, n \\ \Rightarrow \quad & v \in F^\bullet \end{aligned}$$

□

As soon as the condition $p'(v) > 0$ for all $v \in \mathcal{F} \setminus \{0\}$ is given, it is then also clear that we can define the inverse mapping $\phi^{-1}: \mathcal{F}^* \to \mathcal{F}$ by $\phi^{-1}(0) = 0$ and $\phi^{-1}(w) = w \cdot (p'(w))^{-1}$ for $w \in \mathcal{F}^* \setminus \{0\} \subset \mathcal{F} \setminus \{0\}$. The mapping ϕ is then a bijection.

Corollary 3.5. *If the resource function p' fulfils $p'(v) > 0$ for $v \in \mathcal{F} \setminus \{0\}$, the mapping ϕ is a bijection between $\mathcal{F}$ and $\mathcal{F}^*$.*

Note that the condition $p'(v) > 0$ for all $v \in \mathcal{F} \setminus \{0\}$ is surely satisfied if the flux space $\mathcal{F}^*$ is a conic sprout of $\mathcal{C}$ (Def. 2.7), because then an ε-neighbourhood of $0 \in \mathcal{F}$ is also contained in $\mathcal{F}^*$ and hence $(\varepsilon \|v\|^{-1}) \cdot v \in \mathcal{F}^*$ for every $v \in \mathcal{F}$ and in particular $\lambda^* \geq \varepsilon \|\bar{v}\|^{-1} > 0$. With the resource allocation model given by (3.10) the flux space $\mathcal{F}^*$ is always a conic sprout. The minimum value of p is then given by $(\gamma_A + \sum_{j=1}^n \gamma_j)^{-1}$ (or by $(\sum_{j=1}^n \gamma_j)^{-1}$ in the approximated version) and the ε-ball in $\mathcal{C}$ around 0 with $\varepsilon = A_{tot} \cdot \min_{j=1,\ldots,n}(kc_j^+, kc_j^-) \cdot (\gamma_A + \sum_{j=1}^n \gamma_j)^{-1}$ is hence contained in every $\mathcal{F}_g$, $g \in \{0,1\}^n$ and thus also in $\mathcal{F}^*$.

Perspectives of the two resource functions p and p' On a single ray segment we have either $p'(v) = \lambda^* = 0$ and the whole ray segment is mapped to zero, or $p'(v) > 0$ and ϕ is a bijection between $(0, \bar{v}] \subset \mathcal{F}$ and $(0, v^*] \subset \mathcal{F}^*$. The latter is the case if and only if there is a g such that $p_j(g) > 0$ for all j with $v_j \neq 0$. This condition is clearly necessary. To see that it is sufficient, take $\lambda = \min_{j,\, v_j \neq 0} p'_j(v) > 0$ which gives $0 \neq \lambda \bar{v} \in \mathcal{F}^*$ (cf. (3.11)). This gives an equivalent condition for Corollary 3.5 in terms of p, namely that there exists such a $g \in G$ for every $v \in \mathcal{F} \setminus \{0\}$. If this is given, ϕ is a bijection on all ray segments and by Lemma 3.3 thus also between the whole flux spaces $\mathcal{F}$ and $\mathcal{F}^*$.

The resource function p' tells us how much the resource allocation is restricting the flux of a given flux distribution $v \in \mathcal{F}$, where all possible $g \in G$ are considered. Therefore, p' represents not only the resource allocation model, but also the space G of possible gene expressions. On the other hand, p defines the resource allocation under a given fixed gene expression pattern g. The theory developed in this chapter aims at the resource allocation model given by (3.10) and binary gene expression $G \subset \{0,1\}^n$. However, most of the mathematics will only require more general conditions as will be indicated.

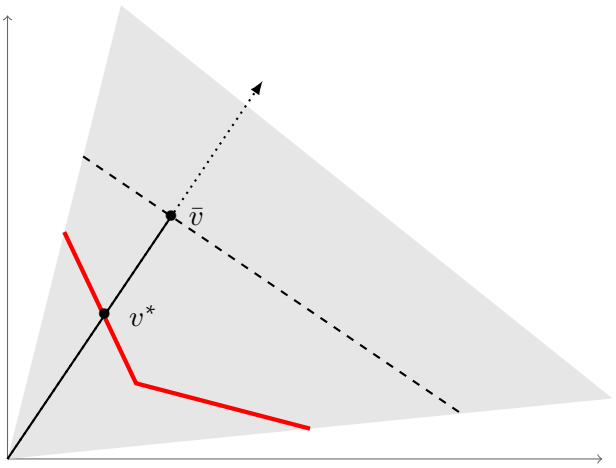


Figure 3.2: A flux cone partly displayed by the shaded area. The dashed line is a linear bound on the flux space and also the set of maximal elements $M \subset \mathcal{F}$. The red line comes from a non-convex bounding set and is also the set of maximal elements $M^* \subset \mathcal{F}^*$. A ray is indicated by the arrow starting in 0. The ray segment $(0, \bar{v}]$ is given by a full line, while the ray outside of $\mathcal{F}$ is dotted. The vector $\bar{v}$ is a maximal element of $\mathcal{F}$. The maximal element in $\mathcal{F}^*$ from this ray segment is $v^* = \phi(\bar{v})\bar{v}$, i.e., the point where the ray crosses $M^* \subset \mathcal{F}^*$.

3.4.2 Convexity of the flux space and the gene expression space

In the following discussion, we will say that the gene expression state $g \in [0,1]^n$ **enables** the flux mode $v \in \mathcal{F}^*$ if $v \in \mathcal{F}_g$ (cf. (3.11)).

If a flux mode v' is enabled by $g' \in G$ and v'' by $g'' \in G$, then a natural candidate for a gene expression state that enables the convex combination $v := \lambda v' + (1-\lambda)v''$ would be $g := \lambda g' + (1-\lambda)g''$. Since $v \in \mathcal{C}$, g enables v if and only if the componentwise inequalities

$$-A_{tot}kc^- p(g) \le v \le A_{tot}kc^+ p(g) \tag{3.15}$$

hold. If the resource function p is concave, i.e., $p(\lambda g' + (1-\lambda)g'') \ge \lambda p(g') + (1-\lambda)p(g'')$ for all $g', g'' \in G$ and $\lambda \in (0,1)$, then (3.15) follows immediately for $g = \lambda g' + (1-\lambda)g''$. Our resource allocation model (3.9) can be written as $p(g) = g \cdot (\gamma_A + \langle g, \gamma \rangle)^{-1}$ using the scalar product notation $\langle g, \gamma \rangle := \sum_{i=1}^n g_i \gamma_i$. This resource function is not concave in general and $g = \lambda g' + (1-\lambda)g''$ need not enable v. However, it is possible to construct a convex combination of g', g'', which enables v. This result will be proven for the approximative model where $p(g) = g \cdot (\langle g, \gamma \rangle)^{-1}$, but the following lemma shows that it is then also implied for $p(g) = g \cdot (\gamma_A + \langle g, \gamma \rangle)^{-1}$, the resource allocation model without the approximation.

Consider the resource allocation model for a given metabolic network. Let $\mathcal{F}^*$ be the flux space if $p(g) = g \cdot (\langle g, \gamma \rangle)^{-1}$ and $\widetilde{\mathcal{F}}$ the flux space if $p(g) = g \cdot (\gamma_A + \langle g, \gamma \rangle)^{-1}$. According to (3.11), both flux spaces are given as the intersection of the flux cone with a bounding set. On the one hand $\mathcal{F}^* = \mathcal{C} \cap \mathcal{B}^*$, where $\mathcal{B}^* = \bigcup_{g \in G} B_g$ and

$$B_g := \{v \in \mathbb{R}^n \ : \ -A_{tot}kc_j^- \frac{g_j}{\langle g, \gamma \rangle} \le v_j \le A_{tot}kc_j^+ \frac{g_j}{\langle g, \gamma \rangle},\ j = 1, \dots, n\}$$

and on the other hand $\widetilde{\mathcal{F}} = \mathcal{C} \cap \widetilde{\mathcal{B}}$, where $\widetilde{\mathcal{B}} = \bigcup_{g \in G} \widetilde{B}_g$ and

$$\widetilde{B}_g := \{v \in \mathbb{R}^n \ : \ -A_{tot} k c_j^- \frac{g_j}{\gamma_A + \langle g, \gamma \rangle} \leq v_j \leq A_{tot} k c_j^+ \frac{g_j}{\gamma_A + \langle g, \gamma \rangle}, \ j = 1, \ldots, n\}.$$

Since the flux cone is always a convex set, the convexity of the flux space depends only on the convexity of the bounding set.

Lemma 3.6. *If $\mathcal{B}^*$ is convex, then $\widetilde{\mathcal{B}}$ is also convex. In particular, if $\mathcal{F}^*$ is convex, then $\widetilde{\mathcal{F}}$ is also convex.*

Proof. The statement for the bounding sets $\mathcal{B}^*$ and $\widetilde{\mathcal{B}}$ implies the same statement for the flux spaces $\mathcal{F}^*$ and $\widetilde{\mathcal{F}}$ as a direct consequence of the convexity of the flux cone $\mathcal{C}$ and Lemma 3.1. Now we will prove the statement for the bounding sets. The elements v given by

$$v_j = \sigma_j A_{tot} k c_j^{\sigma_j} \frac{g_j}{\langle g, \gamma \rangle}, \text{ for some } \sigma \in \{-, +\}^n, \ j = 1, \ldots, n, \tag{3.16}$$

are the vertices (Def. 3.12) of B_g. The vertices of B_g and $\widetilde{B}_g$ are in one-to-one correspondence. The vertex $\tilde{v} \in \widetilde{B}_g$ corresponding to v is given by $\tilde{v}_j = \sigma_j A_{tot} k c_j^{\sigma_j} \frac{g_j}{\gamma_A + \langle g, \gamma \rangle}$, for the same $\sigma \in \{-, +\}^n$ and $j = 1, \ldots, n$.

Let $\mathcal{B}^*$ be convex. For arbitrary $g, g' \in G$ and vertices $v \in B_g$, $v' \in B_{g'}$, the convex combination $\lambda v + (1 - \lambda) v'$ is then element of $B_{g''}$ for some $g'' \in G$. This means that for $j = 1, \ldots n$ we have

$$-A_{tot} k c_j^- \frac{g_j''}{\langle g'', \gamma \rangle} \ \leq \ \lambda \sigma_j A_{tot} k c_j^{\sigma_j} \frac{g_j}{\langle g, \gamma \rangle} + (1 - \lambda) \sigma_j' A_{tot} k c_j^{\sigma_j'} \frac{g_j'}{\langle g', \gamma \rangle} \ \leq \ A_{tot} k c_j^+ \frac{g_j''}{\langle g'', \gamma \rangle}. \tag{3.17}$$

The resource allocation model with the approximation is invariant to scaling of the gene expression state, i.e., the term $\frac{g}{\langle g, \gamma \rangle} \in \mathbb{R}^n_{\geq 0}$ is not changed if g is scaled. Therefore, we can adjust the scaling of g, g', g'' such that

$$\langle g, \gamma \rangle = \langle g', \gamma \rangle = \langle g'', \gamma \rangle.$$

Now we just have to multiply the inequalities (3.17) with $\langle g, \gamma \rangle \left(\gamma_A + \langle g, \gamma \rangle\right)^{-1}$ to obtain the corresponding inequalities without the approximation, i.e., with γ_A in the denominator. These inequalities tell us that the convex combination of the corresponding vertices $\tilde{v} \in \widetilde{B}_g$ and $\tilde{v}' \in \widetilde{B}_{g'}$ lies in $\widetilde{B}_{g''}$. Due to the one-to-one correspondence between the vertices of $\widetilde{B}_g$ and B_g, we have thus shown that an arbitrary convex combination of two vertices of $\widetilde{B}_g$ resp. $\widetilde{B}_{g'}$ is again in $\widetilde{\mathcal{B}}$. To see that this already proves the convexity of $\widetilde{\mathcal{B}}$, let v, v' be two arbitrary elements of $\mathcal{B}^*$. We want to show that $\lambda v + (1 - \lambda) v' \in \mathcal{B}^*$ for any $\lambda \in [0, 1]$.

From (3.16) it is clear that we can obtain vertices $\bar{v}$, $\bar{v}'$ which have in all components the same sign as v resp. v' by componentwise scaling, i.e., $\bar{v}_j = \alpha_j v_j$ with $\alpha_j \geq 1$ and $\bar{v}_j' = \alpha_j' v_j'$ with $\alpha_j' \geq 1$, $j = 1, \ldots, n$. We showed above that for these vertices and any $\lambda \in [0, 1]$ there is some $g'' \in G$ such that

$$\bar{v}'' := \lambda \bar{v} + (1 - \lambda) \bar{v}' \in \widetilde{B}_{g''}.$$

This means that for $j = 1, \ldots, n$ the inequalities

$$-A_{tot} k c_j^- \frac{g_j''}{\gamma_A + \langle g'', \gamma \rangle} \leq \bar{v}_j'' \leq A_{tot} k c_j^+ \frac{g_j''}{\gamma_A + \langle g'', \gamma \rangle}$$

are satisfied. If this holds for $\bar{v}_j''$, then it also holds for $v_j'' := \lambda(\alpha_j)^{-1}\bar{v}_j + (1-\lambda)(\alpha_j')^{-1}\bar{v}_j'$ since $(\alpha_j)^{-1}, (\alpha_j')^{-1} \leq 1$. We thus defined a vector $v'' \in \widetilde{B}_{g''} \subset \widetilde{\mathcal{B}}$. In fact $v'' = \lambda v + (1-\lambda)v'$ and thus convexity of $\widetilde{\mathcal{B}}$ is shown. □

Proposition 3.7. Let $\mathcal{B}^* = \bigcup_{g\in G} B_g$, with $B_g := \{v \in \mathbb{R}^n \,:\, -A_{tot}kc_j^- \frac{g}{\langle g,\gamma\rangle} \leq v \leq A_{tot}kc_j^+ \frac{g}{\langle g,\gamma\rangle},\ j = 1,\ldots,n\}$. If $G \subset [0,1]^n$ is convex, then so is $\mathcal{B}^*$.

Proof. Each B_g, $g \in G$ is defined by the bounds $lb_j(g) := -A_{tot}kc_j^- \frac{g_j}{\langle g,\gamma\rangle},\ ub_j(g) := A_{tot}kc_j^+ \frac{g_j}{\langle g,\gamma\rangle}$. All calculations will only be done for the upper bounds, but hold similarly for the lower bounds. Let $v', v'' \in \mathcal{B}^* \setminus \{0\}$, then we have to show that any convex combination $v := \lambda v' + (1-\lambda)v''$ with $\lambda \in (0,1)$ is in $\mathcal{B}^*$. There are $g', g'' \in G \setminus \{0\}$ with $v' \leq ub(g')$ and $v'' \leq ub(g'')$, so we have $v \leq \lambda ub(g') + (1-\lambda)ub(g'')$. Hence it would suffice to find a $g \in G$, such that

$$\lambda ub_j(g') + (1-\lambda)ub_j(g'') \leq ub_j(g), \quad j = 1,\ldots,n. \tag{3.18}$$

It will be shown now that for some $\mu \in (0,1)$ the convex combination $g = \mu g' + (1-\mu)g''$ satisfies (3.18). Since G is convex, it follows that $g \in G$ and this proves the proposition. For $g' = 0$ or $g'' = 0$, (3.18) holds immediately. Therefore, we can assume that $g',\ g'' \neq 0$. Since we always assume $\gamma_j > 0$, $j = 1,\ldots,n$, in this chapter (see p. 45), it follows that $\langle g',\gamma\rangle > 0$ and $\langle g'',\gamma\rangle > 0$. The calculations will now be done componentwise. Substituting $g = \mu g' + (1-\mu)g''$ in the r.h.s. of (3.18) gives

$$kc_j^+ \frac{\mu g_j' + (1-\mu)g_j''}{\mu\langle g',\gamma\rangle + (1-\mu)\langle g'',\gamma\rangle}$$

The l.h.s., augmented by the factors $\frac{\mu}{\mu}$ and $\frac{(1-\mu)}{(1-\mu)}$, becomes:

$$kc_j^+ \left(\lambda \frac{\mu g_j'}{\mu\langle g',\gamma\rangle} + (1-\lambda)\frac{(1-\mu)g_j''}{(1-\mu)\langle g,\gamma\rangle}\right).$$

With the abbreviations $a := \mu g_j',\ b := (1-\mu)g_j'',\ x := \mu\langle g',\gamma\rangle,\ y := (1-\mu)\langle g'',\gamma\rangle$, where x and y are strictly positive numbers, and by cancelling out the turnover number kc_j^+, we write (3.18) as

$$\begin{aligned}
& \lambda\frac{a}{x} + (1-\lambda)\frac{b}{y} \;\leq\; \frac{a+b}{x+y} \\
\Leftrightarrow\quad & \lambda a + (1-\lambda)b\frac{x}{y} + \lambda a\frac{y}{x} + (1-\lambda)b \;\leq\; a+b \\
\Leftrightarrow\quad & \lambda\left(1+\frac{y}{x}\right)a + (1-\lambda)\left(1+\frac{x}{y}\right)b \;\leq a+b.
\end{aligned}$$

By the choice of μ we will enforce $\frac{x}{y} = \frac{\lambda}{1-\lambda}$ which implies $\lambda(1+\frac{y}{x}) = 1$ as well as $(1-\lambda)(1+\frac{x}{y}) = 1$, so that (3.18) will be satisfied with equality. Since

$$\frac{x}{y} = \frac{\mu}{1-\mu}\frac{\langle g',\gamma\rangle}{\langle g'',\gamma\rangle} \text{ we want to have } \frac{\mu}{1-\mu} = \frac{\lambda}{1-\lambda}\frac{\langle g'',\gamma\rangle}{\langle g',\gamma\rangle} =: \alpha$$

and this is achieved with $\mu = \frac{\alpha}{1+\alpha} < 1$. Note that this calculation, although carried out for a distinct j, leads to an α that is independent of g_j, g_j', but depends only on the scalar products $\langle g,\gamma\rangle$, $\langle g',\gamma\rangle$. Therefore, the corresponding $\mu = \frac{\alpha}{1+\alpha}$ satisfies (3.18) for all $j = 1,\ldots,n$ and we thus have a convex combination $g = \mu g' + (1-\mu)g''$, such that $v \leq ub(g)$. □

This proposition shows that convexity of $\mathcal{B}^*$ follows from convexity of G. Convexity of the bounding set $\mathcal{B}^*$ implies directly convexity of the flux space $\mathcal{F}^*$, see Lemma 3.1. Although Prop. 3.7 was only shown for the approximated resource allocation model, it also holds without the approximation, due to Lemma 3.6 which showed that convexity of the bounding set (and the flux space) with the approximation implies convexity of the bounding set (and the flux space) without the approximation.

3.4.3 Free gene expressions

If the gene expression space does not restrict the gene expression patterns at all, i.e., $G = [0,1]^n$, we are able to represent the corresponding flux space $\mathcal{F}^*$ independently of g. In fact, a maximal $\bar{v} \in M \subset \mathcal{F}^*$ turns out to be enabled by a $g \in [0,1]^n$ which is unique up to scaling and can be given in terms of v. Remember that we assume that all reactions are subject to resource allocation. The following results are in fact only valid if we consider only those reactions that are subject to resource allocation in the calculations. In case the excluded reactions are not rate limiting, they do not contribute to the definition of the flux space and therefore they would not affect the following results.

Lemma 3.8. *If $G = [0,1]^n$, a maximal flux mode $\bar{v} \in M \subset \mathcal{F}^*$ reaches the lower or upper bound in each reaction, i.e., in* (3.8) *we have $v_j = lb_j(g)$ or $v_j = ub_j(g)$ for $j = 1, \dots, n$.*

Proof. Assume the contrary, then there is i, such that w.l.o.g. $\bar{v}_i > 0$ but the upper bound of reaction i is not reached, i.e., $\bar{v}_i = \varepsilon k c_i^+ A_{tot} \frac{g_i}{\gamma_A + \langle g, \gamma \rangle}$ for some $0 < \varepsilon < 1$. Let $\hat{g} \in G$ be equal to g in all components $j \neq i$ and $\hat{g}_i = \varepsilon g_i \in [0,1]$. Then we have $\bar{v}_i = A_{tot} k c_i^+ \frac{\hat{g}_i}{\gamma_A + \langle g, \gamma \rangle}$ and $\langle \hat{g}, \gamma \rangle < \langle g, \gamma \rangle$. We conclude that

$$-A_{tot} k c_j^- \frac{\hat{g}_j}{\gamma_A + \langle \hat{g}, \gamma \rangle} < \bar{v}_j < A_{tot} k c_j^+ \frac{\hat{g}_j}{\gamma_A + \langle \hat{g}, \gamma \rangle} \quad \text{for } j = 1, \dots, n.$$

Therefore, there is some $\lambda > 1$ such that $\lambda \bar{v}$ is still in the bounds given by $\hat{g}$. This is a contradiction because $\bar{v}$ was assumed to be maximal. □

It is now straightforward to see that there is a $g \in [0,1]^n$ which is unique up to scaling and enables the maximal flux mode $\bar{v}$. For convenience we assume w.l.o.g. that $\bar{v} \geq 0$. Then we have $\bar{v} = ub(g) = A_{tot} k c^+ \, g \left(\gamma_A + \langle g, \gamma \rangle\right)^{-1}$. If there is g' with $\bar{v} = ub(g')$, we have

$$\frac{g}{\gamma_A + \langle g, \gamma \rangle} = \frac{g'}{\gamma_A + \langle g', \gamma \rangle} \quad \Leftrightarrow g = g' \cdot \frac{\gamma_A + \langle g, \gamma \rangle}{\gamma_A + \langle g', \gamma \rangle}.$$

To determine a g that enables the flux mode v in terms of v, we will introduce the notation

$$\kappa_j(v) := \begin{cases} k c_j^+, & v_j \geq 0 \\ k c_j^-, & v_j < 0 \end{cases}, \tag{3.19}$$

which allows writing (3.8) equivalently as

$$|v_j| \leq A_{tot} \kappa_j(v) \frac{g_j}{\gamma_A + \langle g, \gamma \rangle} \quad \text{for } j = 1, \dots, n. \tag{3.20}$$

For a maximal flux mode $\bar{v} \in M \subset \mathcal{F}^*$ we know from Lemma 3.8 that all these inequalities are satisfied with equality and for g_j this gives

$$g_j = \frac{|\bar{v}_j|}{\kappa_j(v)} \cdot \frac{\gamma_A + \langle g, \gamma \rangle}{A_{tot}},$$

where we define $\frac{0}{0} := 0$ for the case $v_j = 0$, $\kappa_j(v) = 0$ (note that $\kappa_j(v) = 0$ implies $v_j = 0$ by definition). The resource allocation model with the approximation, i.e., with $p(g) = g \cdot \langle g, \gamma \rangle^{-1}$, is invariant to scaling g. Therefore, the term $\frac{\gamma_A + \langle g, \gamma \rangle}{A_{tot}}$, which is scaling the whole vector g, is irrelevant. This holds similarly for the scaling of $\bar{v}$, which can be substituted by any $v \in (0, \bar{v}]$. The gene expression state for the best adjustment of the metabolic network to a flux distribution $v \in (0, \bar{v}]$ is hence given by $g_j = \frac{|v_j|}{\kappa_j(v)}$. Best adjustment means here that the flux distribution can be maximally increased to $\bar{v}$ under this gene expression state.

Corollary 3.9. *If $G = [0,1]^n$, the constraints* (3.8) *can be reduced to a single constraint. If the approximation is used this is (where $\frac{0}{0} := 0$)*

$$\sum_{j=1}^{n} \frac{|v_j|}{\kappa_j(v)} \gamma_j \leq A_{tot} \quad \textit{and without the approximation} \quad \sum_{j=1}^{n} \frac{|v_j|}{\kappa_j(v)} \gamma_j \leq A_{tot} - A\gamma_A.$$

Proof. This is well defined, because $\kappa_j(v) = 0$ implies $v_j = 0$. Multiplying the inequalities (3.20), which are equivalent to the bounds (3.8) of the resource allocation model, with $\frac{\gamma_j}{\kappa_j(v)}$ and summing them up gives

$$\sum_{j=1}^{n} \frac{|v_j|}{\kappa_j(v)} \gamma_j \leq A_{tot} \frac{\langle g, \gamma \rangle}{\gamma_A + \langle g, \gamma \rangle}.$$

If we use the approximated version of (3.20), the amino acid weight γ_A is neglected and we get just A_{tot} as bound. Otherwise we use (3.5) and (3.6) which gives $A_{tot} = A\gamma_A + A_{tot} \frac{\langle g, \gamma \rangle}{\gamma_A + \langle g, \gamma \rangle}$ and hence $A_{tot} \frac{\langle g, \gamma \rangle}{\gamma_A + \langle g, \gamma \rangle} = A_{tot} - A\gamma_A$, which is exactly the amount of amino acids that are used for enzyme synthesis, cf. p. 43. □

This corollary tells us that an enzyme constraint of the form (3.3), as applied in [Beg et al., 2007, Shlomi et al., 2011a, Schuster et al., 2011], corresponds to the case $G = [0,1]^n$ in our resource allocation model with the approximation in (3.8).

The constraint (3.3) is linear on every subset of $\mathbb{R}^n$ where the signs of the flux rates are not changing, because then, $\kappa(v)_j$ is constant and $|v|$ is linear as a function of v. These subsets are the orthants of $\mathbb{R}^n$. The whole set defined by (3.3) is a ball around zero in the 1-norm that is scaled componentwise by kc_j^+/γ_j and kc_j^-/γ_j in positive and negative direction, respectively. The constraint given by Corollary 3.9 is not linear without the approximation, because the bound depends on the amount A of free amino acids, which depends on $\langle g, \gamma \rangle$, i.e., the amount of amino acids used for enzyme production.

3.5 Optimal sequences of flux modes from the flux space

The goal is now to choose from the flux space of the resource allocation model several flux modes which produce the required output in minimal time when operating consecutively. This can be formulated as an optimisation problem. The question is whether the different flux modes can achieve a shorter production time than a single flux mode. It turns out that this depends on the convexity of the flux space, which was shown above to be determined by the space of possible gene expressions.

The question whether the efficiency can be improved by decomposing the output production into different flux modes, will now be formulated as an optimisation problem. In the previous chapter (p. 37) we introduced (OP1), an optimisation problem that minimises the time needed to produce certain required amounts of metabolites. This notion of efficiency will stay the same and we will just extend (OP1) by allowing the use of l different flux modes $v^1, \dots, v^l$, where $l \in \mathbb{N}$ is a fixed parameter. The requirement of metabolite production is assumed to be expressed by flux through certain target reactions $T \subset \{1, \dots, n\}$. For every target reaction $t \in T$ the requirement is given by $\Gamma_t > 0$ (cf. Sec. 2.3).

$$\begin{aligned} &\min \textstyle\sum_{k=1}^{l} \tau_k, \text{ subject to:} \\ &v^k \in \mathcal{F}^*, \quad k = 1, \dots, l \\ &\textstyle\sum_{k=1}^{l} \tau_k v_t^k \geq \Gamma_t \quad \forall t \in T \\ &\text{with variables:} \\ &\tau_k \in \mathbb{R}_{\geq 0},\ v^k \in \mathbb{R}^n, \quad k = 1, \dots, l \end{aligned} \tag{OP2a}$$

Note that the constraint $v^k \in \mathcal{F}^*$ is a short notation for $Sv^k = 0$ and $-A_{tot} k c_j^- p_j(g) \leq v_j^k \leq A_{tot} k c_j^+ p_j(g)$, $j = 1 \dots, n$, for some $g \in G$. In fact, $g \in [0,1]^n$ are also variables in this optimisation problem which are constraint by $g \in G$. A solution of (OP2a) consists of flux vectors $v^1, \dots, v^l$ and corresponding durations $\tau_1, \dots, \tau_l$, which in combination produce all required amounts Γ_t for target reactions $t \in T$. The total time $\sum_{k=1}^{l} \tau_k$ is minimised. The discussion here is restricted to the case where the target reactions produce some output, see Sec. 2.3. Therefore, all target reactions $t \in T$ are assumed to be irreversible. If for example an exchange reaction $t \in T$ was reversible, this would imply that in the summation $\sum_{k=1}^{l} \tau_k v_t^k$ the corresponding external metabolite can be both, produced and consumed. The net change is then constrained by Γ_t. This means that the external metabolite can be produced in one phase, be stored and also consumed in another phase. Such a scenario is realistic, sugar for example can be stored in the form of glycogen by many organisms. However, this aspect is neglected here. It would lead to different biological considerations and computational problems and should therefore be treated separately. Here we focus on the output of target metabolites and exclude reversible target reactions.

Equivalent formulation The bijection ϕ between the resource allocation flux space $\mathcal{F}^*$ and the polytope $\mathcal{F}$ (see Corollary 3.5) leads to an equivalent formulation of the optimisation problem. If the condition of Corollary 3.5 is fulfilled, i.e., if $p'(v) > 0$ for $v \in \mathcal{F} \setminus \{0\}$, we can substitute

$w^k := \phi^{-1}(v^k)$. This gives $\phi^{-1}(v^k) = (p'(v^k))^{-1} \cdot v^k$ for $v^k \neq 0$ and $\phi^{-1}(v^k) = 0$ otherwise. Since ϕ is a bijection, we have $v^k \in \mathcal{F}^* \Leftrightarrow w^k \in \mathcal{F}$. Substituting furthermore $\tau'_k := \tau_k p'(w^k)$ into (OP2a) gives the equivalent optimisation problem:

$$\begin{aligned}
&\min \textstyle\sum_{k=1}^{l} \tau'_k (p'(w^k))^{-1}, \text{ subject to:} \\
&w^k \in \mathcal{F}, \quad k = 1, \ldots, l \\
&\textstyle\sum_{k=1}^{l} \tau'_k w_t^k \geq \Gamma_t \quad \text{for } t \in T \\
&\text{with variables:} \\
&\tau'_k \in \mathbb{R}_{\geq 0},\ w^k \in \mathbb{R}^n, \quad k = 1, \ldots, l
\end{aligned} \tag{OP2b}$$

The products $\tau'_k w_j^k$ in the constraints of output production can be eliminated by using an equivalent formulation similar to (OP1b), p. 38. For readability we will mostly stick to the formulation (OP2a).

The number of flux modes in the optimal solution is bounded In the above optimisation problem (OP2a), p. 55, the number of flux vectors in the solution is fixed to l. In fact, l is just an upper bound, since some of the v^i might not be used, which is the case if the corresponding duration is $\tau_i = 0$. This bound seems to restrict the search space for an optimal solution, because the number l of different flux modes in (OP2a) is a priori not known. However, the following proposition shows that if the bound is fixed correctly there are always optimal solutions in the restricted search space.

Proposition 3.10. If (OP2a) is feasible for some $l \in \mathbb{N}$, there also exists an optimal solution using at most as many flux modes $v^1, \ldots, v^{|T|}$ as there are target reactions in T.

Proof. Assume there is an optimal solution consisting of flux modes $v^1, \ldots, v^l$ and associated durations $\tau_1^*, \ldots, \tau_l^*$ with $l > |T|$. Consider the following linear optimisation problem where the flux modes $v^1, \ldots, v^l$ are parameters and the only variables are the durations τ_k:

$$\min\{\sum_{k=1}^{l} \tau_k : \textstyle\sum_{k=1}^{l} \tau_k v_t^k \geq \Gamma_t,\ t \in T, \tau \geq 0\}.$$

By definition, $\tau^* = (\tau_1^*, \ldots, \tau_l^*)^\top$ is an optimal solution. From the theory of linear programming (see e.g. [Griva et al., 2009, Thm. 4.7, p. 121]), we know that the corresponding problem in standard form $\min\{\sum_{k=1}^{l} \tau_k : \sum_{k=1}^{l} \tau_k v_t^k - \tau' = \Gamma_t,\ t \in T,\ \tau, \tau' \geq 0\}$ has a so-called basic optimal solution $\bar{\tau}, \bar{\tau}' \geq 0$, for which the number of non-zero components $\bar{\tau}_k > 0$ is at most $|T|$. Such a solution $\bar{\tau}$ describes how to produce the demanded output in minimal time, using at most $|T|$ of the flux modes $v^1, \ldots, v^l$. □

It is hence enough to solve (OP2a) for the fixed value $l = |T|$ to find a solution which is optimal for all $l \geq |T|$. To obtain globally optimal solutions of the optimisation problem, we will therefore always solve the optimisation problem with fixed $l = |T|$. Unless stated otherwise, this result also holds for variations and extensions of (OP2a) which will be introduced.

Optimising over a convex flux space The optimisation of output production with a single flux mode given by (OP1) is an instance of (OP2a) with $l = 1$. Let $\tau(l)$ denote the objective value of (OP2a), i.e., the time needed for output production with l consecutive flux modes. This generalisation of (OP1) to (OP2a) with $l > 1$ is motivated by the hypothesis that the required time can be reduced by optimal resource allocation in different consecutive flux modes. Surely $\tau(1) \leq \tau(l)$, since every solution with one flux mode can be extended to a solution with l flux modes by setting $\tau_i = 0$ and $v^i = 0$, $i = 2, \ldots, l$. A reduction of production time is achieved if $\tau(l) < \tau(1)$. However, this is not always possible.

Proposition 3.11. The objective value $\tau(l)$ of (OP2a) cannot be strictly smaller than $\tau(1)$ if the flux space $\mathcal{F}^*$ is convex.

Proof. Assume that $\mathcal{F}^*$ is convex and $v^1, \ldots, v^l, \tau_1, \ldots, \tau^l$ is an optimal solution of (OP2a) with objective value $\tau(l) := \sum_{k=1}^{l} \tau_k$. The convex combination $v := \sum_{k=1}^{l} \frac{\tau_k}{\tau(l)} v^k$ is then an element of $\mathcal{F}^*$ and $\tau(l) \cdot v_t \geq \Gamma_t$ for the target reactions $t \in T$. This means that v, $\tau(l)$ is a solution to (OP2a) with only one flux mode and thus $\tau(1) = \tau(l)$. □

In particular, the linear enzyme capacity constraint (3.3) cannot explain a benefit of alterations in the flux distribution, since the resulting flux space is a convex polytope. Neither can any linear constraint provide such an explanation, because it preserves the convexity of the flux space. Also in our resource allocation model with convex G the flux space is convex. The geometry of $\mathcal{F}^* = \bigcup_{g \in G} \mathcal{F}_g$ depends on the one hand on G and on the other hand on the resource function. We will now consider the case $G \subset \{0, 1\}^n$ which gives a non convex flux space and can lead to an increase in efficiency if a sequence of different flux modes is used for the production of target metabolites

3.6 Binary gene expression

The model of resource allocation will be considered with binary gene expression. This choice is based on biological observations concerning the control of flux rates by gene expression. To obtain a biologically meaningful model, further restrictions on gene expression patterns should be imposed. The formulation of the corresponding optimisation problems is presented and we take a look at the geometry of the flux space.

3.6.1 Biological justification of the binary gene expression model

Modelling the gene expression by binary variables means that in the dynamic sequence the genes can be switched on or off and hence a distinct reaction is either blocked or maximally activated. If a gene is active, the precise bound on the corresponding reaction only depends on the turnover rates $kc_j^{\pm}$ and the distribution of resources among all activated reactions given by the resource allocation function. The level of gene expression cannot be perfectly adjusted to the flux rates of $v \in \mathcal{F}^*$ in this model. As a result, v does usually not use all available capacity, which means that only few bounds are rate limiting while most are strongly redundant (see Sec. 2.2.2). In contrast, imposing only the piecewise linear enzyme capacity constraint (3.3),

allows for perfect adjustment of the gene expression and resource allocation to a given flux mode such that all reactions are rate limiting. As shown by Corollary 3.9, the here discussed resource allocation model reduces to (3.3) if $G = [0,1]^n$.

From the biological point of view, the binary model of gene expression and the resulting coarse adjustment to the flux distribution is in concordance with the results of [Chubukov et al., 2013], which indicate that the gene expression only determines which parts of the metabolic network are activated and which are shut down. The precise level of gene expression was observed to be rarely in correlation with the flux rate of the corresponding reaction. Furthermore, it is known that usually only few reactions in a metabolic pathway are rate limiting. The implications and the eligibility of our model will be further discussed in Sec. 4.5.1.

3.6.2 Formalisation of binary gene expression

The restriction of the gene expression states to $G \subset \{0,1\}^n$ is simplifying the association of a suiting gene expression state with a given flux mode. If $v_j \neq 0$ then $g_j = 1$ is necessary, but the activation of further genes g_j with $v_j = 0$ would be a waste of resources. In the case $G = \{0,1\}^n$, the gene expression which enables the maximal flux through a flux mode v would hence be given by $g_j = 1 \;\Leftrightarrow\; v_j \neq 0$. If we consider a proper subset $G \subsetneqq \{0,1\}^n$, it might be unavoidable to activate genes $g_j = 1$, while $v_j = 0$. It is then preferable to activate genes which correspond to enzymes with small molecular weight γ_j, formally, the sum $\langle g, \gamma \rangle = \sum_{j\,:\,g_j=1} \gamma_j$ should be minimal.

In the resource allocation constraints (3.8) we set the value of the gene expression to $g_j = 1$. The fact that the reaction is blocked if $g_j = 0$ is then implemented by the additional constraint $g_j = 0 \Rightarrow v_j = 0$. The optimisation problem (OP2a), p. 55, is stated as follows (where $\gamma_A = 0$ if we use the approximated version of the resource allocation model and otherwise γ_A is the average weight of one amino acid).

$$
\begin{aligned}
&\min \sum_{k=1}^{l} \tau_k \text{ such that} \\
&Sv^k = 0,\ k = 1,\dots,l \\
&-A_{tot} k c_j^- \left(\gamma_A + \langle g^k, \gamma \rangle\right)^{-1} \leq v_j^k \leq A_{tot} k c_j^+ \left(\gamma_A + \langle g^k, \gamma \rangle\right)^{-1}, \\
&j = 1,\dots,n,\ \ k = 1,\dots,l \\
&g_j^k = 0 \Rightarrow v_j^k = 0, \quad j = 1,\dots,n,\ \ k = 1,\dots,l \\
&g^k \in G,\ k = 1,\dots,l \\
&\sum_{k=1}^{l} \tau_k v_t^k \geq \Gamma_t, \text{ for } t \in T,\ k = 1,\dots,l \\
&\text{with variables } \tau_k \in \mathbb{R}_{\geq 0},\ v^k \in \mathbb{R}^n,\ g^k \in \{0,1\}^n,\ k = 1,\dots,l.
\end{aligned}
\tag{OP3}
$$

The constraint $g \in G$ is only relevant if $G \subsetneqq \{0,1\}^n$ of course. In this case it is enforcing the activation of genes g_j while $v_j = 0$. In realistic models of transcriptional regulation of metabolism, such restrictions of the gene expression space to $G \subsetneqq \{0,1\}^n$ emerge naturally.

3.6.3 Realistic gene expression patterns

The individual metabolic genes in a cell cannot be arbitrarily activated or inactivated, because certain groups of genes are typically transcribed together. In order to determine biologically meaningful solutions, the space of gene expression patterns G thus has to be restricted to a proper subset of $\{0,1\}^n$. For the application to the core metabolic network (Chap. 4), the concept of **minimal flux modes** [Hoffmann and Holzhütter, 2009, Hoffmann et al., 2006] was adopted.

Minimal flux modes to define functional groups of genes Minimal flux modes (MFMs) represent minimal functional capabilities of the metabolic network. These flux modes fulfil a given function while using a minimal amount of resources. For example, we can consider a MFM $\bar{v}$ maximising the flux rate through one distinct target reaction $t \in T$, while consuming one of the available substrates (taken up by reaction u). If we use the resource allocation model given by (3.9), then $\bar{v}$ is an optimal solution of

$$\begin{aligned} &\max\ v_t \text{ s.t.} \\ &v \in \mathcal{F}^*,\ v_i = 0 \text{ for all uptake reactions } i \neq u \\ &g \in \{0,1\}^n \\ &\text{with variables } v \in \mathbb{R}^n, g \in \{0,1\}^n. \end{aligned} \tag{3.21}$$

The objective of using a minimal amount of resources is in this case formulated by $\max v_t$ which also depends on minimising the value $p'(v)$ of the resource allocation function (cf. (OP2b), p. 56). With the resource allocation model given by (3.8), the bounds of $\mathcal{F}^*$ on the target function t can be written as $-kc_t^- A_{tot} \leq \bar{v}_t\left(\gamma_A + \langle g, \gamma\rangle\right) \leq kc_t^+ A_{tot}$, and hence maximising v_t is equivalent to minimising the resource allocation term $\left(\gamma_A + \langle g, \gamma\rangle\right)$. The supports of the MFMs $\bar{v}^1, \ldots, \bar{v}^r$ will be denoted **minimal gene sets** (MGS) $\chi^1, \ldots, \chi^r$, i.e., $\chi^i := \operatorname{supp}(\bar{v}^i) := \{j : \bar{v}_j \neq 0\}$. The gene expression space G is defined by the assumption that these functional groups are regulated together. This means that G is the set of all g where all genes of one distinct MGS are concertedly activated or inactivated. Note also that one gene can be a member of several MGSs.

Formulation with linear constraints In order to include this additional constraint into the optimisation problem, we introduce binary variables b_s, $s = 1, \ldots, r$, where r is the total number of MFMs, to indicate whether the gene group χ^s is active or inactive. It follows that $g_j = 1$ if and only if there exists $s \in \{1, \ldots, r\}$ with $j \in \chi^s$ and $b_s = 1$. To formulate this as a linear constraint, let $\sigma_j = \{s \mid \bar{v}_j^s \neq 0\}$ be the set of MFMs using reaction j. Then we have to require that

$$\begin{aligned} g_j &\leq \sum_{s \in \sigma_j} b_s, \text{ for all } j = 1, \ldots, n, \text{ and} \\ b_s &\leq g_j, \text{ for all } s = 1, \ldots, r, \text{ and } j \in \chi^s. \end{aligned} \tag{3.22}$$

which is equivalent to

$$g_j \leq \sum_{s \in \sigma_j} b_s \leq |\sigma_j|\, g_j, \text{ for all } j = 1, \ldots, n.$$

Constraints on the sequence of gene expression states In the general optimisation problem (OP2a) the ordering of the different gene expression states $g^1, \dots g^l$ is arbitrary. From the biological point of view, it is however desirable to include also constraints on the ordered sequence $\vec{g} := (g^1, \dots, g^l)$ of gene expression states. This way, the possible dynamics of the gene regulatory network can be represented. The space of all admissible gene expression sequences of length l will be denoted $Dyn_l \subseteq G^l$ and (OP2a) is then generalised to

$$\begin{aligned} &\min \textstyle\sum_{k=1}^{l} \tau_k, \text{ subject to:} \\ &v^k \in \mathcal{F}_{g^k}, \quad k = 1, \dots, l \\ &(g^1, \dots, g^l)^\top \in Dyn_l \\ &\textstyle\sum_{k=1}^{l} \tau_k v_t^k \geq \Gamma_t \quad \forall t \in T \\ &\text{with variables:} \\ &\tau_k \in \mathbb{R}_{\geq 0},\ v^k \in \mathbb{R}^n,\ g^k \in \{0,1\}^n, \quad k = 1, \dots, l \end{aligned} \tag{3.23}$$

The constraint $g \in G$ in (OP2a) is just replaced with $(g^1, \dots, g^l)^\top \in Dyn_l$ to get the generalised form (3.23). Individual durations τ_i can take the value $\tau_i = 0$, which means that the corresponding state g^i is omitted. As a consequence, the set of admissible sequences Dyn_l is practically extended with all sub-sequences of all $\vec{g} \in Dyn_l$. Note that Prop. 3.10 is still valid, since it is independent of the order in the sequence of flux modes (and hence also in the sequence of corresponding gene expression patterns).

Once a space Dyn_l of admissible sequences is defined, a possible next step in the biological specification of the model could be to associate minimal delays $\delta(g, g')$ for switches between the states g and g' and enforce them by the constraints $\tau_i \geq \delta(g^i, g^{i+1})$, $i = 1 \dots, l-1$. If the optimisation problem (OP2a) is extended by such constraints, Prop. 3.10 need not be valid in general.

The space of monotone sequences A rather simple restriction of the space of sequences is given by the constraints that will be used in the application in Chap. 4, namely that genes can only be switched on, formally, $g^i \leq g^{i+1}$, $i = 1, \dots, l-1$.

3.6.4 Geometry of the flux space

The flux space represented as a union of polytopes In Sec. 3.3, the flux space defined by a general resource function p was introduced as $\mathcal{F}^* := \{v \in \mathbb{R}^n : Sv = 0,\ \exists g \in G \text{ s.t. } -A_{tot} k c_j^- p_j(g) \leq v_j \leq A_{tot} k c_j^+ p_j(g),\ j = 1 \dots, n\}$ which can also be given as $\mathcal{F}^* = \bigcup_{g \in G} \mathcal{F}_g$. The $\mathcal{F}_g$ are all closed and bounded polytopes by definition and in Sec. 3.4 we assumed that also $\mathcal{F}^*$ is closed. In the case of binary gene expression this assumption is surely satisfied, since we have a finite space G and $\mathcal{F}^*$ is hence a finite union of closed and bounded polytopes. The properties bounded and closed are preserved under finite unions, but $\mathcal{F}^*$ need not be convex, see Figs. 3.4 and 3.3.

If $v \in \mathcal{F}^*$ is an EM and $g \in \{0,1\}^n$ is the support of v then, by definition of EMs, v is up to scaling the only flux mode on the sub-network of all reactions j with $g_j = 1$. The polytope $\mathcal{F}_g$ is hence just a ray-segment including zero, i.e., $\mathcal{F}_g = \{\lambda \bar{v} : 0 \leq \lambda \leq 1\}$ for a maximal $\bar{v} \in M \subset \mathcal{F}^*$. In particular, it follows that $\mathcal{F}_g \nsubseteq \mathcal{F}_{g'}$ for any two EMs $g' \neq g$. In general we can have the situation that the polytope $\mathcal{F}_g$ is contained in $\mathcal{F}_{g'}$ for some $g' \neq g$. For example,

assume the reactions i and k are partially coupled (Def. 2.5). Let g, g' be identical except on i, say $g_i = 1$, $g'_i = 0$, and assume $g_k = 0$, $g'_k = 0$. Reaction k is blocked in g as well as in g' and due to the coupling reaction i cannot carry flux neither. Since the resource allocation terms differ, i.e., $\langle g, \gamma \rangle > \langle g', \gamma \rangle$, this construction results in flux spaces where the bounds in $\mathcal{F}_g$ are strictly smaller by a factor $\langle g', \gamma \rangle \cdot \langle g, \gamma \rangle^{-1} < 1$ than those of $\mathcal{F}_{g'}$ for all reactions except i, where the bounds are zero in $\mathcal{F}_{g'}$, but not in $\mathcal{F}_g$. However, reaction i cannot carry flux in neither flux space due to the coupling to the blocked reaction k and thus we have the strict inclusion $\mathcal{F}_g \subsetneqq \mathcal{F}_{g'}$.

Example of an EM that is not a vertex Due to their minimal support, EMs are "sticking out" in the flux space, see Fig. 3.4 or Fig. 3.3. While the EMs in Fig. 3.4 are also vertices (Def. 3.12) of $\mathcal{F}$, Fig. 3.3 gives an example of an EM (namely e^1) that is a local vertex (Def. 3.13) $v^* \in \mathcal{F}^*$ but $\phi^{-1}(v^*)$ is not a vertex in $\mathcal{F}$. Lets look at the following flux vectors of the network given in Fig. 3.3 a):

$$e^1 = (1, 0, 1, 1, 1, 0)$$
$$e^2 = (2, 1, 1, 2, 0, -1)$$
$$e^3 = (0, -1, 1, 0, 2, 1)$$

These are all the EMs of the network, as can be verified by hand. However, e^1 is also a convex combination of e^2 and e^3 and therefore it is not a vertex. The situation can be depicted in the projection onto the reactions 2 and 3, see Fig. 3.3 b).

3.7 Candidates for optimal solutions

As shown in Sec.3.4, the flux space maps bijectively to a linear flux space which is obtained by neglecting the limited resource allocation. This correspondence is used now to identify finite sets of elements which are sufficient to obtain optimal sequences. The analysis aims at binary gene expression, but is done in a more general setting. The key in the theoretical analysis is the support of the flux modes. The characterisation of the sufficient elements will be given in terms of modified variants of decomposability. The result that there is always an optimal sequence consisting only of flux modes from such a finite set depends crucially on the condition that the flux space contains zero, which excludes models with maintenance constraints.

For an LP over a polytope we know that if an optimal solution exists, there is also a vertex (Def. 3.12) that is an optimal solution. Similarly, we can characterise elements of $\mathcal{F}^*$, such that there is an optimal solution of (OP2a), p. 55, which consists only of such flux modes. In fact, the characterisation will be done in the associated polytope $\mathcal{F}$ and mapped to $\mathcal{F}^*$ with ϕ given by (3.14). The construction aims at the resource allocation model based on (3.10) and binary gene expression $G \subset \{0, 1\}^n$ as in (OP3), p. 58, but the mathematics just requires that the resource function p' given by (3.13) fulfils $p'(v) > 0$ for $v \in \mathcal{F} \setminus \{0\}$, so that ϕ is a bijection by Corollary 3.5 and furthermore that the gene expression state G is finite and that p' is monotone

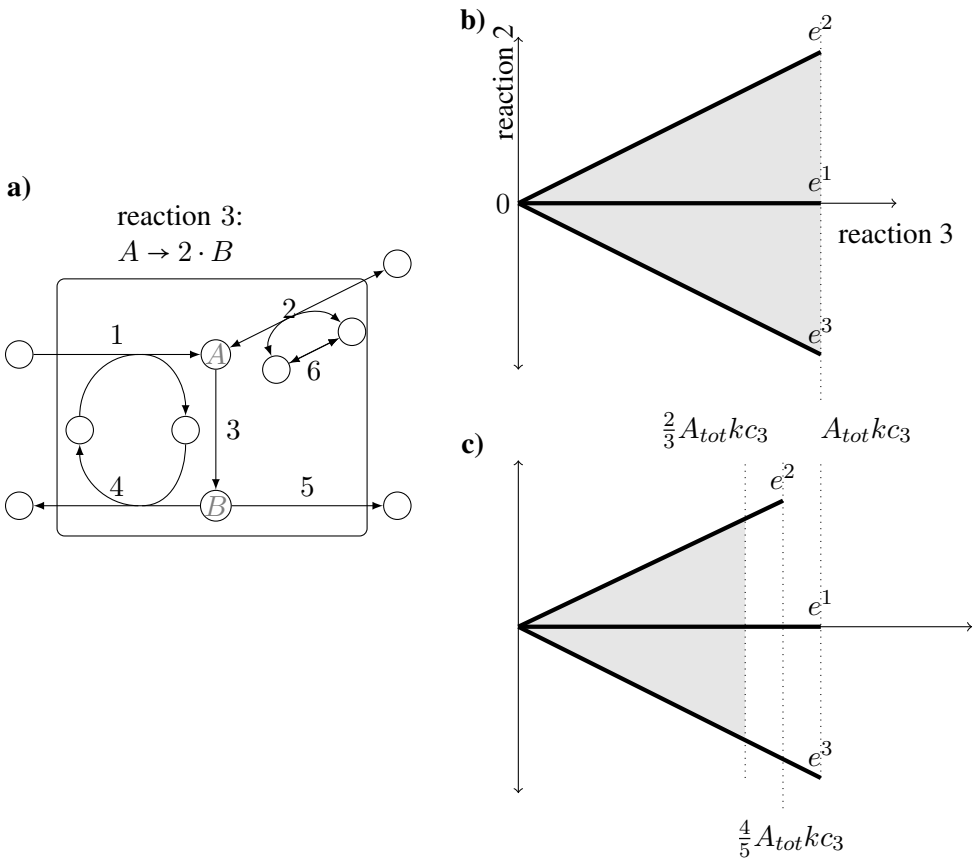


Figure 3.3: **a)** A network with six reactions. The four external metabolites outside the box are not constraint to be in steady state. Reaction 3 has the stoichiometry $A \to 2 \cdot B$, but all other stoichiometric coefficients are ± 1. We define the positive direction of reaction 2 to be the consumption of A. Only reaction 3 is bounded, namely by $A_{tot} kc_3^+$ in the polytope $\mathcal{F}$. This implies bounds on all other reactions of the network. The pathway using the reactions $\{1, 3, 4, 5\}$ gives an EM $e^1 = (1, 0, 1, 1, 1, 0)$. This EM is not a vertex, as is apparent in **b)** where the projection of $\mathcal{F}$ on the reactions 2 and 3 is shown. The bounds are not specified but the projection assumes that $ub_3 = A_{tot} kc_3^+ \leq 2\, ub_2 = 2\, A_{tot} kc_2^+$ and $ub_3 \leq -2\, lb_2$. For the resource allocation (approximated version of (3.8)), the molecular weights are set to $\gamma_i = \frac{1}{4}$, $i = 1, \ldots, 6$. The resulting flux space $\mathcal{F}^*$ gives the projection shown in **c)**. While the EMs e^1, e^2, e^3 are using $5, 4, 5$ reactions, respectively, all other flux modes are necessarily convex combinations of the EMs e^1, e^2, e^3 and it is straightforward to see that they must use all 6 reactions. The resulting bounds on reaction 3, given by the resource allocation model in case $4, 5$ or 6 reactions are activated, are indicated by the dotted lines.

in the support, i.e., $\mathrm{supp}(v') \subseteq \mathrm{supp}(v) \Rightarrow p'(v') \geq p'(v)$. By assumption we have $0 \in \mathcal{F}^*, \mathcal{F}$ due to the non-negativity of A_{tot}, $kc_j^{\pm}$ and the resource functions p and p'.

Definition 3.12. Let $F \subset \mathbb{R}^n$ be a polytope or, more general, a polyhedron. An element $v \in F$ is called a **vertex** of F if it is not a convex combination of other elements of F, i.e., there exist no $x, y \in F \setminus \{v\}$, such that $v = \lambda x + (1 - \lambda) y$, $\lambda \in [0, 1]$.

This definition is adopted from [Bertsimas and Tsitsiklis, 1997], where a vertex is defined as $v \in F$ such that there exists a $c \in \mathbb{R}^n$ with $c^\top x > c^\top v$ for all $x \in F \setminus \{v\}$ and the equivalence to the characterisation in Def. 3.12 is proven. The non-convex flux space of (OP2a) is a finite union of polytopes, $\mathcal{F}^* := \bigcup_{g \in G} \mathcal{F}_g$, as long as G is a finite set. Let v be a vertex of $\mathcal{F}_g$ and not be contained in any other $\mathcal{F}_{g'}$, $g' \neq g$. Since v is not contained in the finitely many $\mathcal{F}_{g'}$ there is some $\varepsilon > 0$, such that the ε-neighbourhood $\{x \in \mathcal{F}^* : \|x - v\| \leq \varepsilon\}$ of v in $\mathcal{F}^*$ contains only elements of $\mathcal{F}_g$ and v is thus not a convex combination of elements of this ε-neighbourhood.

Definition 3.13. A flux mode $v \in \mathcal{F}^* \setminus \{0\}$ is called a **local vertex**, if there is an $\varepsilon > 0$ such that for all $x^1, x^2 \in \mathcal{F}^* \setminus \{v\}$ and $\|x^i - v\| \leq \varepsilon$, $i = 1, 2$, there is no $\lambda \in [0, 1]$ with $v = \lambda x^1 + (1 - \lambda)x^2$.

Definition 3.14. A flux mode v from the polytope $\mathcal{F} = \{v \in \mathbb{R}^n : Sv = 0,\ -A_{tot}kc^- \leq v \leq A_{tot}kc^+\}$ is called **convexly decomposable**, if it is a convex combination $v = \lambda x^1 + (1 - \lambda)x^2$ with $x^i \in \mathcal{F} \setminus \{v\}$, such that $\operatorname{supp}(x^i) \subseteq \operatorname{supp}(v)$, $i = 1, 2$. By $\mathbf{\Omega}$ we denote the set of all elements which are not convexly decomposable and not zero.

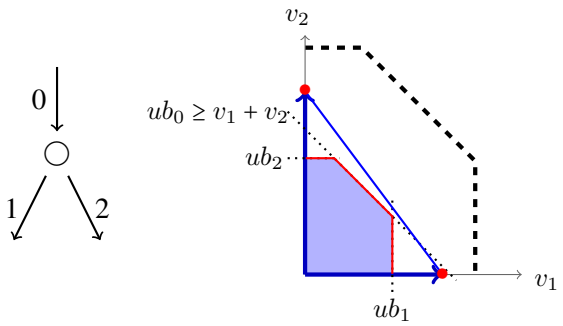


Figure 3.4: The projection of a flux space $\mathcal{F}^*$, corresponding to the network on the left, onto the reactions 1 and 2. The dotted lines show the upper bounds ub_0, ub_1, ub_2 in case all reactions are used. If only v_0, v_1 or only v_0, v_2 are used, the bounds are bigger and the maximal flux modes that only use one of the reactions 1, 2 can increase their flux. This leads to the disconnected set $M^* \subset \mathcal{F}^*$ of maximal elements (Def. 3.2) indicated in red. The two thick arrows correspond to the EMs of the network, namely $(1, 1, 0)$ and $(1, 0, 1)$, which are sticking out in $\mathcal{F}^*$ due to their smaller support. The dashed line indicates the projection of $M \subset \mathcal{F}$, the set of maximal elements of the polytope $\mathcal{F}$. It contains the projection of $\mathcal{F}^*$ (since $p'(v) \in [0, 1]$). All other vertices of $\mathcal{F}$ are mapped to local vertices of $\mathcal{F}^*$. The two vertices where both reactions 1 and 2 are used are at the same time convex combinations of the EMs in $\mathcal{F}^*$, as indicated with the blue line. However, locally they cannot be given as a convex combination.

Note that local vertices as well as not convexly decomposable elements are necessarily maximal elements (Def. 3.2) in their respective flux space. Since a local vertex v or a maximal element v is not 0 by definition, we can assume that v is contained in a ray segment $(0, v^*]$. In case $v \neq v^*$, it would be a convex combination of $(1 - \varepsilon)v$ and $(1 + \varepsilon)v$ which are both in $(0, v^*]$ for small enough ε. This is a contradiction and therefore we must have $v = v^*$. The same argument shows that also in the ray segment $(0, \bar{v}] \subset \mathcal{F}$ all elements $v \neq \bar{v}$ must be convexly decomposable. This shows that Ω, the set of all elements that are not convexly decomposable is contained in M.

The mapping ϕ is a bijection under the conditions given here by Lemma 3.5 and it will be used now to characterise elements in $\mathcal{F}$ and in $\mathcal{F}^*$.

Lemma 3.15. *If $\phi(v) \in \mathcal{F}^*$ is a local vertex then $v \in \mathcal{F}$ is not convexly decomposable.*

Proof. The contraposition will be shown. Assume $v \in \mathcal{F}$ is convexly decomposable, so there exist $x^1, x^2 \in \mathcal{F}$ as in Def. 3.14, in particular $v = \lambda x^1 + (1-\lambda)x^2$. We can assume that x^i are arbitrarily close to v, i.e., for $\varepsilon > 0$ and $i = 1, 2$, $\|x^i - v\| < \varepsilon$. This can always be achieved by replacing

$$\begin{aligned} x^1 \quad &\text{with} \quad y^1 := (\lambda+\delta)x^1 + (1-\lambda-\delta)x^2 \\ x^2 \quad &\text{with} \quad y^2 := (\lambda-\delta)x^1 + (1-\lambda+\delta)x^2 \end{aligned}$$

and choosing $\delta < \varepsilon(\|x^1 - x^2\|)^{-1}$. Then for $i = 1, 2$ we have $\|y^i - v\| = \|\delta(x^1 - x^2)\| < \varepsilon$ and $\frac{1}{2}y^1 + \frac{1}{2}y^2 = \lambda x^1 + (1-\lambda)x^2 = v$.

If we choose $\varepsilon < \min_{j, v_j \neq 0}(|v_j|)$, then $\operatorname{supp}(x^i) \supseteq \operatorname{supp}(v)$ as soon as $\|x^i - v\| < \varepsilon$, $i = 1, 2$. In the other direction we have by assumption $\operatorname{supp}(x^i) \subseteq \operatorname{supp}(v)$, so that in fact $\operatorname{supp}(x^i) = \operatorname{supp}(v)$, $i = 1, 2$. A direct consequence is $p'(v) = p'(x^i)$, $i = 1, 2$, due to the monotonicity of p' w.r.t. the support. From $v = \lambda x^1 + (1-\lambda)x^2$ we can therefore conclude

$$\begin{aligned} \phi(v) := p'(v)v =& \lambda p'(v)x^1 + (1-\lambda)p'(v)x^2 \\ =& \lambda p'(x^1)x^1 + (1-\lambda)p'(x^2)x^2 = \lambda\phi(x^1) + (1-\lambda)\phi(x^2). \end{aligned}$$

We saw that x^i can be arbitrarily close to v. But in fact we want to show that $\phi(x^i)$ is arbitrarily close to $\phi(v)$, which would imply that $\phi(v)$ is not a local vertex. This follows immediately, because $\|\phi(x^i) - \phi(v)\| = \|p'(x^i)x^i - p'(v)v\| = p'(v)\|x^i - v\|$, $i = 1, 2$. As we can choose x^i such that $\|x^i - v\| < \varepsilon \cdot (p'(v))^{-1}$, we get $\|\phi(x^i) - \phi(v)\| < \varepsilon$ for arbitrary $\varepsilon > 0$ and $i = 1, 2$. □

An essential ingredient in this proof was the equality $p'(v) = p'(x^i)$ which was deduced from the inclusion of the supports. We will introduce a notion of decomposability which refers directly to the resource allocation function p'. This will allow establishing a one-to-one correspondence between certain elements of $\mathcal{F}$ and the local vertices of $\mathcal{F}^*$.

Definition 3.16. A flux mode $v \in M \subset \mathcal{F}$ is called **p'-decomposable** if it is a convex combination $v = \lambda x^1 + (1-\lambda)x^2$ of $x^i \in F \setminus \{v\}$, such that $p'(x^i) \geq p'(v)$, $i = 1, 2$.

Analogously to Ω (Def. 3.14), we define $\boldsymbol{\Omega_{p'}}$ as the set of those elements of $\mathcal{F} \setminus \{0\}$ which are not p'-decomposable. Due to the monotonicity of p' w.r.t. the support, every convexly decomposable element is also p'-decomposable. A vertex of $\mathcal{F}$ is neither convexly decomposable nor p'-decomposable for any p'. Altogether we have the inclusions

$$\{\text{vertices } \neq 0 \text{ of } \mathcal{F}\} \subset \Omega_{p'} \subset \Omega \subset M \tag{3.24}$$

Also the non-decomposable elements (Def. 2.2) and the EMs (Def. 2.1) of the flux space $\mathcal{F}$ are elements of Ω as follows directly from the definitions. The elements of $\Omega_{p'}$ are now in a one-to-one correspondence to the local vertices given by ϕ, i.e., Lemma 3.15 can be generalised to

Proposition 3.17. Let p' be monotone in the support, i.e., $\operatorname{supp}(v') \subseteq \operatorname{supp}(v) \;\Rightarrow\; p(v') \geq p(v)$. Then $v \in \Omega_{p'}$ if and only if $\phi(v) \in \mathcal{F}^*$ is a local vertex.

Proof. To proof '$\Leftarrow$' we can adopt the argument of Lemma 3.15. Just replace the condition $\operatorname{supp}(x^i) \subset \operatorname{supp}(v)$ for convexly decomposability with $p'(x^i) \geq p'(v)$ for p'-decomposability.

Using the monotonicity of p' w.r.t. the support, gives $p'(x^i) \leq p'(v)$ in the ε-neighborhood of v and hence the equalities $p'(x^i) = p'(v)$, $i = 1, 2$.

To verify the other direction, assume $v \in \Omega_{p'}$, i.e., $v \in \mathcal{F} \setminus \{0\}$ is maximal and not p'-decomposable. To see that $v^* := \phi(v) = p'(v)v$ is then a local vertex of $\mathcal{F}^*$, suppose that for $x^*, y^* \in \mathcal{F}^*$ we have

$$\begin{aligned} \lambda x^* + (1-\lambda)y^* = v^* \quad &\Leftrightarrow \quad \lambda p'(x)x + (1-\lambda)p'(y)y = p'(v)v \\ &\Leftrightarrow \quad \lambda \frac{p'(x)}{p'(v)} p'(v)x + (1-\lambda)\frac{p'(y)}{p'(v)} p'(v)y = p'(v)v \\ &\Leftrightarrow \quad \hat{\lambda}x + \hat{\mu}y = v \end{aligned}$$

with $\hat{\lambda} = \lambda \frac{p'(x)}{p'(v)}$ and $\hat{\mu} = (1-\lambda)\frac{p'(y)}{p'(v)}$. Since $x, y \in \mathcal{F}$ and $\hat{\lambda}x + \hat{\mu}y = v \in \mathcal{F}$ is maximal, we must have $\hat{\lambda} + \hat{\mu} \geq 1$ (otherwise $\alpha v \in \mathcal{F}$ for an $\alpha > 1$). The case $\hat{\lambda} + \hat{\mu} > 1$ is only possible if w.l.o.g. $p'(v) < p'(y)$. In the complementary case $\hat{\lambda} + \hat{\mu} = 1$ we must have $p'(v) > p'(x)$ w.l.o.g., because v is not p'-decomposable by assumption. As a consequence, $\hat{\lambda} = \lambda \frac{p'(x)}{p'(v)} < \lambda$ and thus $\hat{\mu} = (1-\hat{\lambda}) > (1-\lambda)$ which is equivalent to $p'(v) < p'(y)$. Concluding, we have w.l.o.g. $p'(v) < p'(y)$ in any case and this implies, by the contrapositive of the monotonicity of p', that $\operatorname{supp}(v) \nsubseteq \operatorname{supp}(y)$. The support is clearly an invariant of the bijection ϕ (that is, $\operatorname{supp}(\phi(v)) = \operatorname{supp}(v)$) and we thus get $\operatorname{supp}(v^*) \nsubseteq \operatorname{supp}(y^*)$. As we saw in the proof of Lemma 3.15, the support of y^* is a superset of v^* as soon as $\|y^* - v^*\| < \min_{j \in \operatorname{supp}(v^*)}(|v_j^*|)$. Therefore, a convex combination $\lambda x^* + (1-\lambda)y^* = v^*$ with $x^*, y^* \in \mathcal{F}^*$ must fulfil $\|x^* - v^*\| \geq \varepsilon := \min_{j \in \operatorname{supp}(v^*)}(|v_j^*|)$ and this shows that v^* is a local vertex. □

Due to the inclusion of the vertices $\neq 0$ in $\Omega_{p'}$, (3.24), an immediate consequence is that every vertex $v \in \mathcal{F} \setminus \{0\}$ maps to a local vertex $\phi(v) := p'(v)v$ of $\mathcal{F}^*$, see Fig. 3.4. Also EMs are contained in Ω and Fig. 3.3 gives an example of an EM that is not a vertex in $\mathcal{F}$. Furthermore, Ω contains elements that are neither vertices nor EMs. This is the case for a flux mode $v \in \mathcal{F}$, if either it cannot be written as a convex combination of $x, y \in \mathcal{F} \setminus \{v\}$ or it has minimal support in the set $\{z \in \mathbb{R}^n : z = x + \alpha(v - x),\ \alpha \in \mathbb{R}\}$. This situation is illustrated in Fig. 3.5.

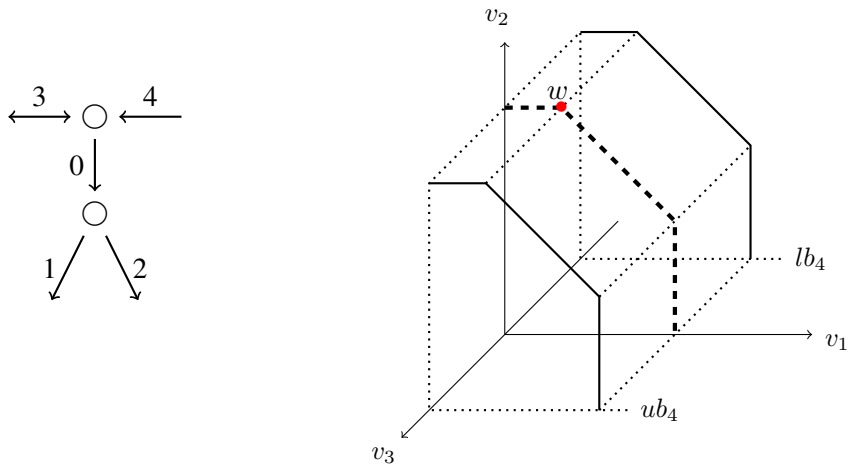


Figure 3.5: An example of an element $w \in \Omega$ that is neither an EM nor a vertex. The network is obtained from the example of Fig. 3.4 by adding the two reactions 3 and 4. The positive direction of 3 is going out of the network. Assuming that $ub_4 \geq ub_0 + ub_3$ and $-lb_3 \leq \min(ub_1, ub_2)$ the flux v_3 can vary freely between lb_3 and ub_3 as long as one of the bounds ub_0, ub_1 or ub_2 is attained, i.e., the set of maximal elements M in the shown projection to the reactions 1, 2 and 3 can be given as $M = M' \times [lb_3, ub_3]$, where M' is the set of maximal elements in the projection to reactions 1 and 2. The flux mode w, indicated by a red dot, has minimal support along the dimension of v_3, but it can only be written as a convex combination along this dimension.

3.7.1 Implications for optimal solutions

The set Ω of all not convexly decomposable elements gives the subset $\phi(\Omega)$ of $\mathcal{F}^*$. An optimal solution of (OP2a) can always be composed from elements of this subset. This means that in the optimisation the search space for the flux modes can be restricted a priori to $\phi(\Omega)$ instead of $\mathcal{F}^*$. The same result can be obtained for the potentially smaller set $\Omega_{p'}$, under an additional technical condition on p'. In the next section we will see that the set Ω can be generated by enumeration of EMs, which offers an alternative possibility to solve (OP2a).

Lemma 3.18. *Every element $v \in \mathcal{F}$ can be decomposed into $v = \sum_{i=1}^{k} \lambda_i w^i$ such that $\lambda_i \geq 0$, $\sum_{i=1}^{k} \lambda_i = 1$, $\operatorname{supp}(w^i) \subseteq \operatorname{supp}(v)$ and $w^i \in \Omega$, for $i = 1, \ldots, k$.*

Proof. Let $\overline{\operatorname{supp}}(v)$ be the set of indices j where $v_j = 0$. The intersection of the polytope $\mathcal{F}$ and the subspace $\{x \in \mathbb{R}^n : x_i = 0, i \in \overline{\operatorname{supp}}(v)\}$ is again a polytope and will be denoted $F_{v=0}$. Clearly $v \in F_{v=0}$. Since a polytope is the convex hull of its vertices [Bockmayr et al., 2001, p. 736], we can decompose $v = \sum_{i=1}^{k} \lambda_i w^i$, where w^i, $i = 1, \ldots, k$, are vertices of $F_{v=0}$ and $\sum_{i=1}^{k} \lambda_i = 1$. If there are $x, y \in \mathcal{F} \setminus \{v\}$ and $\lambda \in [0, 1]$ such that $w^i = \lambda x + (1 - \lambda) y$, then x and y cannot be both elements of $F_{v=0}$, because w^i is a vertex of $F_{v=0}$. From the definition, it is evident that $F_{v=0} = \{x \in \mathcal{F} : \operatorname{supp}(x) \subseteq \operatorname{supp}(v)\}$. The fact that x or y is not element of $F_{v=0}$ implies therefore that this is not a convex decomposition by Def. 3.14. We conclude that w^i is not convexly decomposable. □

Proposition 3.19. If (OP2a) has an optimal solution, then it has also an optimal solution that consists only of flux modes $\phi(w^i) \in \mathcal{F}^*$, with $w^i \in \Omega$.

Proof. Let $v^* \in \mathcal{F}^*$ be part of an optimal solution with the associated duration τ^*. We can assume w.l.o.g. that $\tau^* > 0$ and this implies for an optimal solution that $v^* \neq 0$. The flux mode can be mapped by the bijection ϕ^{-1} to $v := \phi^{-1}(v^*) \in \mathcal{F}$. Decomposing v as in Lemma 3.18 into $v = \sum_{i=1}^{k} \lambda_i w^i$ and mapping the decomposition back to $\mathcal{F}^*$ gives the elements $\phi(w^1), \ldots, \phi(w^k) \in \mathcal{F}^*$. Since we have $\mathrm{supp}(w^i) \subseteq \mathrm{supp}(v)$ and the resource function p' is monotone in the support, we get $p'(w^i) \geq p'(v)$. Multiplying the convex combination v with $p'(v)$ gives

$$v^* = \phi(v) = p'(v)\, v = \sum_{i=1}^{k} \lambda_i\, p'(v) \cdot w^i = \sum_{i=1}^{k} \lambda_i \frac{p'(v)}{p'(w)} p'(w) \cdot w^i = \sum_{i=1}^{k} \lambda_i \frac{p'(v)}{p'(w)} \cdot \phi(w^i).$$

Since $p'(w^i) \geq p'(v)$, $i = 1, \ldots, k$ we have $\sum_{i=1}^{k} (\lambda_i \frac{p'(v)}{p'(w)}) \leq 1$, replacing $\tau^* v^*$ by $\phi(w^i)$ with durations $\tau^* \cdot (\lambda_i \frac{p'(v)}{p'(w)})$ achieves exactly the same production of metabolites, while the total duration is not increased. This means that the new solution is not inferior and is thus optimal as well. □

Replacing the individual flux modes of an optimal solutions by decompositions into local vertices $\phi(w^1), \ldots, \phi(w^k)$ is blowing up the optimal sequence of flux modes. However, this new sequence can always be reduced to a subset of at most $|T|$ of these local vertices of $\mathcal{F}^*$ as follows from Prop. 3.10.

Under an extra condition on p', Lemma 3.18 holds also with p'-decomposability, i.e., every element $v \in \mathcal{F}$ can be decomposed into $v = \sum_{i=1}^{k} \lambda_i w^i$, $\sum_{i=1}^{k} \lambda_i = 1$, such that $p'(w^i) \leq p'(v)$ and $w^i \in \Omega_{p'}$, for $i = 1, \ldots, k$. The proof can be given similarly. We define $F_{\leq p'(v)} := \{x \in \mathcal{F} : p'(x) \leq p'(v)\}$, which need not be convex. Now we have to assume the extra condition that $F_{\leq p'(v)}$ is closed. Let $V := \mathrm{conv}(F_{\leq p'(v)})$. Since $F_{\leq p'(v)}$ is closed by assumption and bounded by definition, also V is closed and bounded. Let $X \subset V$ be the set of elements of V which cannot be written as convex combination of other elements of V. In general convex sets V these elements are called extreme points [Rockafellar, 1997, p. 162] and a closed and bounded V is the convex hull of its extreme points [Rockafellar, 1997, Corollary 18.5.1,p. 167], i.e., $\mathrm{conv}(X) = V$. In particular, $v \in F_{\leq p'(v)}$ is a convex combination $v = \sum_{i=1}^{k} \lambda_i w^i$, where $w^i \in X$, $i = 1, \ldots, k$. By definition of X and V we must have $X \subset F_{\leq p'(v)}$. Therefore, $w^i \in F_{\leq p'(v)}$ and by construction w^i is not p'-decomposable analogously to Lemma 3.18. Since $\Omega_{p'} \subset \Omega$ this is a slightly stronger result, given that p' fulfils the condition that $F_{\leq p'(v)}$ is closed.

3.7.2 Maintenance constraints

The so-called **maintenance reactions** (introduced on p. 28) in the metabolic network model represent vital cellular functions as e.g. energy production or protection from toxic substances, which have to be fulfilled permanently. In any flux mode a certain minimum flux rate $mb_j > 0$ through the maintenance reaction j is therefore required, giving the additional **maintenance constraints**

$$v_j^k \geq mb_j,$$

for each flux mode v^k, $k = 1 \ldots, l$ in a solution of the optimisation problem. To allow for a convenient vector notation, we set $mb_j = -\infty$ for all j which are not maintenance reactions. This defines a vector $mb \in (\mathbb{R} \cup \{-\infty\})^n$ and we can write the maintenance constraints as

$$v^k \geq mb. \tag{3.25}$$

These constraints exclude a neighbourhood of 0 from the flux space in $\mathbb{R}^n$. Lemma 3.15 and all following results are based on the bijective mapping ϕ, which is only a bijection if $0 \in \mathcal{F}$ and $p'(v) > 0$ for $v \in \mathcal{F} \setminus \{0\}$. If $0 \notin \mathcal{F}$, the results in this section are not valid anymore, see Fig. 3.6 for an illustration.

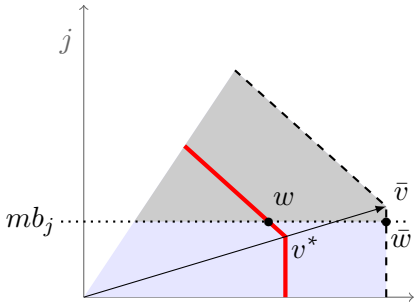


Figure 3.6: An illustration of the maintenance constraints. The flux space $\mathcal{F}$ is given by the shaded area bounded by the dashed line which indicates the maximal set M. The red full line gives M^*, the set of maximal elements of $\mathcal{F}^*$. The arrow indicates the ray segment $(0, \bar{v}] \subset \mathcal{F}$ including the ray segment $(0, v^*]$ of $\mathcal{F}^*$. The maintenance constraint $v_j \geq mb_j$ reduces the flux space to the darker shaded area above the dotted line which indicates the flux modes with $v_j = mb_j$. On the one hand, the maintenance constraint cuts away the vertex $v^* = \phi(\bar{v})$. On the other hand, new vertices $w \in \mathcal{F}^*$ and $\bar{w} \in \mathcal{F}$ emerge. The mapping ϕ is not a bijection and all results based on ϕ are not valid anymore, in particular the identification of local vertices with $\Omega_{p'}$ by Prop. 3.17. Also the construction of optimal solutions from elements of the image of Ω or $\Omega_{p'}$ under ϕ given in Prop. 3.19 is not possible anymore.

Another consequence of maintenance constraints is that the EMs of the flux cone need not coincide with the non-decomposable elements and the approach presented in the following section, where the set Ω is determined by enumeration of EMs, is hence not applicable in metabolic networks with maintenance constraints.

3.8 Finding solutions from pre-selected flux vectors

In the previous sections we identified the subsets Ω and $\Omega_{p'}$ of the flux space, such that an optimal sequence of flux modes can always be composed of flux modes from one of these subsets. The condition that zero is contained in the flux space was essential. For the case that the flux space is even a conic sprout, we will now show how enumeration of elementary modes can be used to generate Ω. A construction will be given which allows obtaining all not convexly decomposable elements from an enumeration of non-decomposable flux vectors of a modified network. Once the set Ω is given, an optimal sequence can be found by solving an LP.

3.8.1 Reducing the search space to a finite pre-selection

If we reduce the search space in the optimisation problem (OP2a), p. 55, to a finite subset, the problem is in fact reduced to an LP. The finite subset will be called a **pre-selection** $\mathscr{A} := \{x^1, x^2, \ldots, x^q\} \subseteq \mathcal{F}^*$. We consider the optimisation problem (OP2a) on the search space $\mathscr{A}$ instead of $\mathcal{F}^*$. Since $\mathscr{A}$ is a finite set, this can be equivalently written as the LP

$$\begin{aligned} &\min \sum_{i=1}^{q} \tau_i, \text{ such that} \\ &\sum_{i=1}^{q} \tau_i x_t^i \geq \Gamma_t \quad \text{for } t \in T \\ &0 \leq \tau \\ &\text{with variables } \tau \in \mathbb{R}^q. \end{aligned} \tag{LPps}$$

Theoretically, any number of entries τ_i might be non-zero. However, LP solvers usually finish the computation with simplex iterations and therefore they return a basic optimal solution τ containing at most $|T|$ non-zero entries (cf. Prop. 3.10). To compute an optimal solution allowing less than $|T|$ flux modes, i.e., $l < |T|$ in (OP2a), additional non-convex constraints would be required. A straightforward implementation would be to introduce binary variables $s_i = 1 \Leftrightarrow \tau_i > 0$ and the constraint $\sum_{i=1}^{q} s_i \leq l$. The resulting MILP with q binary variables is unlikely to be computationally tractable (cf. Sec. 6.2) and therefore, the pre-selection approach is not suited to compute optimal solutions of (OP2a) with $l < |T|$. Since the optimisation is performed over a subset $\mathscr{A} \subseteq \mathcal{F}^*$, the objective value of (LPps) is less or equal to the objective value of (OP2a). The goal is to construct a pre-selection such that the difference is as small as possible. In the best case the objective values are equal.

Definition 3.20. A pre-selection $x^1, \ldots, x^q \in \mathcal{F}$ is called **perfect**, if an optimal solution of (LPps) constitutes also an optimal solution of the optimisation problem (OP2a), p. 55.

3.8.2 Relationship to elementary modes

A direct consequence of Prop. 3.19 is that the set $\Omega \subset \mathcal{F}$ of not convexly decomposable elements gives a perfect pre-selection if we map it to $\mathcal{F}^*$ with the bijection ϕ. In this section we show how Ω can be computed by using EM enumeration. We are only dealing with the linear flux space $\mathcal{F} = \{v \in \mathbb{R}^n : Sv = 0,\ -A_{tot}kc_j^- \leq v_j \leq A_{tot}kc_j^+,\ j = 1, \ldots, n\}$ and will just write $lb_j = -A_{tot}kc_j^-$ and $ub_j = A_{tot}kc_j^+$, $j = 1, \ldots, n$. Every flux mode that is not decomposable in the usual sense (Def. 2.2) is in particular also not convexly decomposable. If the flux space $\mathcal{F}$ is a conic sprout, the non-decomposable elements and the EMs coincide in $\mathcal{F}$, see Prop. 2.3 and the equivalences on page 32. Using EM enumeration, we can then get the set of all non-decomposable elements. In general this is just a subset of Ω, but it turns out that for certain flux spaces the two sets coincide. For our resource allocation model given by (3.10), the flux space is surely a conic sprout, as noted after Corollary 3.5, and the here presented results can thus be applied.

Proposition 3.21. If all except one single reaction i have redundant bounds (see Sec. 2.2.2), then the not convexly decomposable elements coincide with the non-decomposable elements.

Proposition 3.22. If the flux space $\mathcal{F}$ is only bounded by the molecular crowding constraint (3.3), then the not convexly decomposable elements coincide with the non-decomposable elements.

For both propositions we only have to show that $v \in \mathcal{F}$ being not convexly decomposable implies that v is also non-decomposable, because the implication in the other direction always holds.

Lemma 3.23. *Let* $F := \mathcal{C} \cap \{v \in \mathbb{R}^n : \sum_{j=1}^n b_j|v_j| \leq 1\}$ *and* $F' := \mathcal{C} \cap \{v \in \mathbb{R}^n : \sum_{j=1}^n a_j v_j \leq 1\}$ *for some* $a, b \in \mathbb{R}^n$. *Every decomposable element in* F *or* F' *is also convexly decomposable.*

As a consequence, the non-decomposable and the not convexly decomposable elements coincide in F and in F'. For Prop. 3.21 we apply the lemma for F' with $a_i = (ub_i)^{-1}$, $a_j = 0$ for $j \neq i$ and for Prop. 3.22 we use F with $b_j = q_j(kc_j)^{-1}$, $j = 1, \ldots, n$, to express (3.3) (where $kc_j := \max(kc_j^+, kc_j^-)$).

Proof. If $v \in F$ is decomposable, there are $x, y \in F$ and $\lambda, \mu > 0$ such that $v = \lambda x + \mu y$ and $\operatorname{supp}(x), \operatorname{supp}(y) \subsetneq \operatorname{supp}(v)$. For the case of F we need to replace x, y with x' resp. y', such that $\operatorname{sign}(x'_j) = \operatorname{sign}(y'_j) \in \{0, +, -\}$ for all $j = 1, \ldots, n$. Therefore, consider for some $\delta > 0$ the elements $x' := \delta x + (1-\delta)v$ and $y' := \delta y + (1-\delta)v$. With $\lambda' := \lambda\left(\delta + (1-\delta)(\lambda+\mu)\right)^{-1}$ and $\mu' := \mu\left(\delta + (1-\delta)(\lambda+\mu)\right)^{-1}$ we have $\lambda' x' + \mu' y' = v$ and we can choose δ such that x' and y' lie in an ε-ball around v. For small enough ε, i.e., $\varepsilon < \min_{j \in \operatorname{supp}(v)}(|v_j|)$, every w with $\|w - v\| \leq \varepsilon$ fulfils $\operatorname{supp}(w) \supset \operatorname{supp}(v)$ and we can hence assume that $\operatorname{supp}(x'), \operatorname{supp}(y') = \operatorname{supp}(v)$ which is equivalent to $x'_j = 0 \Leftrightarrow v_j = 0 \Leftrightarrow y'_j = 0$ for all $j = 1, \ldots, n$. As a consequence, x' and y' also have no opposing signs $x'_j = -y'_j \neq 0$, because this would imply that on the one hand $v_j \neq 0$ and on the other hand that a convex combination $w = \alpha x' + (1-\alpha)y'$ with $w_j = 0$ exists, which is a contradiction, since w is contained in the convex ε-ball around v.

The following calculation is given for the case of F, but holds similarly for F' with $a \in \mathbb{R}^n$ instead of b and without application of the absolute value. Assume that $v \in M$, i.e., v is maximal according to Def. 3.2, otherwise it would already be convexly decomposable, see (3.24). By scaling x', y' and adjusting the scalars $\lambda', \mu' \geq 0$ accordingly, we can also assume that $x', y' \in M$. For $z = x',\ y',\ v$ we thus have $\sum_{j=1}^n b_j|z_j| = 1$. Due to the identical signs of x'_j, y'_j we can conclude that

$$\begin{aligned} \lambda' + \mu' &= \lambda' \sum_{j=1}^n b_j|x'_j| + \mu' \sum_{j=1}^n b_j|y'_j| \\ &= \sum_{j=1}^n b_j|\lambda' x'_j + \mu' y'_j| \\ &= \sum_{j=1}^n b_j|v_j| \ = 1. \end{aligned} \tag{3.26}$$

We conclude that $\lambda' x' + \mu' y' = v$ is a convex decomposition of v. □

Optimal solutions with molecular crowding If the flux space is a conic sprout, the non-decomposable elements coincide with the EMs of the network. In this case, the EMs give a pre-selection according to Prop. 3.22. In other words, (OP2a) has an optimal solution that consists only of EMs. In fact one single EM suffices, because the flux space of molecular crowding is convex (see Prop. 3.11). The molecular crowding constraint (3.3) can be directly derived

from the limited enzyme amount, $\sum_{j=1}^{n} q_j E_j \leq 1$, and the bounds (3.2), which depend linearly on the enzyme amounts for the individual reactions, see page 41. Furthermore, we saw in Corollary 3.9 that the resource allocation model reduces to (3.3) in case $G = [0,1]^n$. In fact, bounding the total enzyme amount and implementing a linear relationship between individual enzyme amounts and the bounds on the flux rates was shown to always result in EMs as optimal solutions for maximisation of distinct fluxes, independently of the precise form of the reaction kinetics [Müller et al., 2014, Wortel et al., 2014].

3.8.3 Extending the set of elementary modes yields a pre-selection

If the flux space is not of the kind of F or F' in Lemma 3.23, we will see that this can be enforced by extending the metabolic network. Projecting the non-decomposable elements of the extended network onto the original one yields exactly those elements which are not convexly decomposable there, that is, we get exactly Ω. If the flux space of the original network is a conic sprout, then this also holds for the extended network and hence its EMs coincide with its non-decomposable elements. Therefore, based on this construction we can apply EM enumeration in the extended network to obtain Ω in the original network.

Construction of the extended network As usual, the reactions of the original network are indexed by $J = \{1, \ldots, n\}$. Let $J^+ = \{j^+(1), \ldots, j^+(k)\}$ be the subset of reactions whose upper bound is not redundant and $J^- = \{j^-(1), \ldots, j^-(h)\}$ those, whose lower bound is not redundant and not 0 (lower bounds $lb_j = 0$ can be omitted here, because they will be represented in the irreversibility constraints $v_j \geq 0$ for $j \in J^{irr}$). A reaction j can have non-redundant upper and lower bounds and in this case $j \in J^+ \cap J^-$. The unique non-redundant reaction in the extended network will be a new artificial reaction called **unibound**. It has the bounds $lb_{uni} = 0$ and $ub_{uni} = 1$ and will substitute the bounds for all other reactions, because they will be directionally coupled to the unibound reaction. This coupling is established by the constraints

$$\begin{aligned} v_j &\leq ub_j \cdot \tilde{v}_{uni}\,, \text{ for } j \in J^+ \\ v_j &\geq lb_j \cdot \tilde{v}_{uni}\,, \text{ for } j \in J^- \end{aligned} \tag{3.27}$$

where $\tilde{v}_{uni}$ is the flux rate through the unibound reaction. To implement these constraints in the metabolic network, some more artificial metabolites and reactions will be added.

To define the extension formally, the stoichiometric matrix $\widetilde{S}$ of the extended network must be defined. Let S be the stoichiometric matrix of the original network. The stoichiometric coefficients of the unibound reaction and the artificial metabolites are given by a vector $b \in \mathbb{R}^{k+h}$ with all non-redundant bounds as entries, i.e.,

$$b_i := \begin{cases} ub_{j^+(i)}, & i \leq k \\ -lb_{j^-(i-k)}, & i > k \end{cases}$$

The consumption and production of the artificial metabolites by the reactions J^+ and J^- is given by the $(k+h) \times n$-matrix A, with $A_{i,j} := \begin{cases} -1, & i \leq k \text{ and } j = j^+(i) \\ 1, & \text{if } i > k \text{ and } j = j^-(i) \\ 0, & \text{otherwise} \end{cases}$. The stoichiometry

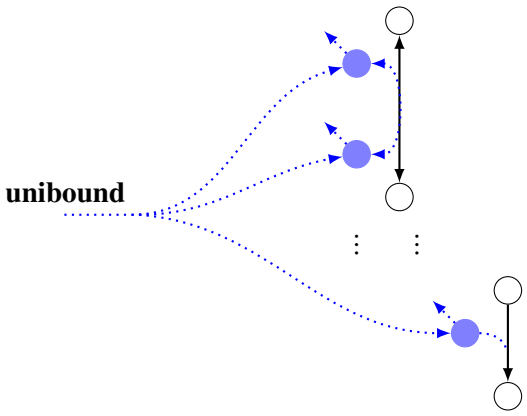


Figure 3.7: The extension of the network. The new metabolites are the filled circles and the new reactions are dotted as well as the modifications of original reactions. The original reactions which have non-redundant bounds are modified to additionally consume an artificial metabolite which is exclusively produced by the unibound reaction. In case the lower as well as the upper bound of a reversible reaction are non-redundant, one artificial metabolite is added for each direction. Depending on the direction, the one or the other metabolite will be consumed. The bounds on the unibound reaction are $lb_{uni} = 0$ and $ub_{uni} = 1$ and its stoichiometry implements the original bounds of the non-redundant reactions. A flux of $\tilde{v}_{uni} = 1$ produces the artificial metabolite(s) for the positive (and negative) direction of reaction j with stoichiometric coefficient ub_j (resp. $-lb_j$). All bounds except ub_{uni} are hence redundant. The new artificial metabolites are also subject to the steady-state assumption. To allow that the non-redundant reactions have fluxes below the bounds given by the flux rate of the unibound reaction, so-called **leak reactions** are added to let the artificial metabolites drain off. The leak reactions as well as the unibound reaction are irreversible.

of the extended network is then given by the $(m+k+l)\times(n+k+l+1)$ stoichiometric matrix

$$\widetilde{S} := \begin{pmatrix} S & 0 & 0 \\ A & b & -Id \end{pmatrix},$$

where Id is the identity matrix in $k+h$ dimensions. The last l columns define the **leak reactions** which consume the artificial metabolites. The flux space of the extended network is given by

$$\widetilde{\mathcal{F}} = \{\tilde{v} \in \mathbb{R}^{n+k+l+1} : \widetilde{S}\tilde{v} = 0,\ \tilde{v}_j \geq 0 \text{ for } j \in J^{irr} \text{ or } j > n,\ \tilde{v}_{uni} \leq 1\}.$$

An element $\tilde{v} \in \widetilde{\mathcal{F}}$ can be written as

$$\tilde{v} = \begin{pmatrix} v \\ \tilde{v}_{uni} \\ \tilde{v}_L \end{pmatrix}, \tag{3.28}$$

where $v \in \mathbb{R}^n$, $\tilde{v}_{uni}$ is the flux rate of unibound and $\tilde{v}_L$ are the flux rates of the leak-fluxes, i.e., L is the index set of the leak reactions, $L = \{n+2, n+3, \ldots, n+1+k+l\}$.

Consider the flux cone defined by $\widetilde{S}\tilde{v} = 0$ and $\tilde{v}_j \geq 0$ for the irreversible reactions in the original network and the artificial reactions which are all irreversible, i.e.,

$$\widetilde{\mathcal{C}} := \{v \in \mathbb{R}^{n+k+l+1} : \widetilde{S}\tilde{v} = 0,\ \tilde{v}_j \geq 0 \text{ for } j \in J^{irr} \text{ or } j > n\}.$$

The flux space $\widetilde{\mathcal{F}}$ is obtained from $\widetilde{\mathcal{C}}$ by adding the single bounding constraint $\tilde{v}_{uni} \leq 1$. The corresponding halfspace (cf. p. 15) $H_{uni} = \{v \in \mathbb{R}^{n+k+l+1} : v_{uni} \leq 1\}$ clearly contains an ε-ball around 0 in $\widetilde{\mathcal{C}}$. This means that $\widetilde{\mathcal{F}}$ is a conic sprout of $\widetilde{\mathcal{C}}$ (Def. 2.7).

3.8.4 Projecting the extended flux space onto the original flux space

The constraints $\widetilde{S}\tilde{v} = 0$ are equivalent to $Sv = 0$ and

$$Av + b\,\tilde{v}_{uni} = \tilde{v}_L \quad \Leftrightarrow \quad \begin{cases} -v_j + ub_j\,\tilde{v}_{uni} = \tilde{v}_{L(j)},\ j \in J^+ \\ v_j - lb_j\,\tilde{v}_{uni} = \tilde{v}_{L(j)},\ j \in J^-. \end{cases}$$

Since $\tilde{v}_L \geq 0$ and $0 \leq \tilde{v}_u \leq 1$, this implies the original bounds for all non-redundant reactions, i.e., $v_j \leq ub_j$ for $j \in J^+$ and $v_j \geq lb_j$ for $j \in J^-$. This gives the implication $\tilde{v} \in \widetilde{\mathcal{F}} \Rightarrow v \in \mathcal{F}$ for $v, \tilde{v}$ as in (3.28). Therefore, we can define the projection $\pi\colon \widetilde{\mathcal{F}} \to \mathcal{F}$ by projecting $\tilde{v} \in \widetilde{\mathcal{F}} \subset \mathbb{R}^{(n+k+h+1)}$ onto the first n coordinates, that is, onto v. To see that π is surjective, a lifting $\psi\colon \mathcal{F} \to \widetilde{\mathcal{F}}$ will be given, such that $\pi(\psi(v)) = v$. Let $v \in \mathcal{F}$, then the lifted element is $\tilde{v} = \psi(v) := \begin{pmatrix} v \\ 1 \\ b + Av \end{pmatrix}$. By construction, $\tilde{v}$ is an element of $\widetilde{\mathcal{F}}$. Note that an element $\tilde{v} \in \widetilde{\mathcal{F}}$ is maximal if and only if $\tilde{v}_{uni} = 1$. As a consequence, the image of the lifting is exactly the set $\widetilde{M}$ of maximal elements in $\widetilde{\mathcal{F}}$ (Def. 3.2). On the other hand, if v is maximal in $\mathcal{F}$ then $\tilde{v}_{uni} = 1$ must hold. The projection of an element that is not maximal in $\widetilde{\mathcal{F}}$ is therefore not maximal in $\mathcal{F}$.

Since unibound is the only reaction with non-redundant bounds in the extended network, the non-decomposable elements are exactly the not convexly decomposable elements $\widetilde{\Omega} \subset \widetilde{\mathcal{F}}$ according to Prop. 3.21. The projection π turns out to establish a one-to-one correspondence between $\widetilde{\Omega} \subset \widetilde{\mathcal{F}}$ and $\Omega \subset \mathcal{F}$. In case the flux space is a conic sprout (Def. 2.7), this allows generating Ω by enumerating the EMs in the extended network and projecting them.

Proposition 3.24. The not convexly decomposable elements in the extended network project onto the not convexly decomposable elements in the original network, i.e., $\pi(\widetilde{\Omega}) = \Omega$.

Proof. The notation in this proof is according to (3.28). To get the inclusion $\Omega \subseteq \pi(\widetilde{\Omega})$, we show that every element that is convexly decomposable in $\widetilde{\mathcal{F}}$ is projected to a convexly decomposable element in $\mathcal{F}$. If $\tilde{v} \in \widetilde{\mathcal{F}}$ is not maximal then the projection $v = \pi(\tilde{v})$ is also not maximal and hence convexly decomposable as noted before, p. 63. Therefore it just remains to check maximal elements. Let $\tilde{v} = \lambda\tilde{x} + (1-\lambda)\tilde{y}$ be a convex decomposition of a maximal element, i.e., $\operatorname{supp}(\tilde{x}), \operatorname{supp}(\tilde{y}) \subset \operatorname{supp}(\tilde{v})$. Then the elements $\tilde{x}$ and $\tilde{y}$ are also maximal due to Lemma 3.25 below. As a consequence, there are elements $x, y \in \mathcal{F}$, such that $\tilde{x} = \psi(x)$ and $\tilde{y} = \psi(y)$, because the range of ψ are the maximal elements in $\widetilde{\mathcal{F}}$. Since ψ is by construction injective, it follows that $x \neq y$. Since the projection is linear, we have

$$\lambda x + (1-\lambda)y = \lambda\pi(\psi(x)) + (1-\lambda)\pi(\psi(y)) = \pi(\lambda\tilde{x} + (1-\lambda)\tilde{y}) = \pi(\tilde{v}) = v.$$

The projection clearly preserves the inclusions of the supports, so we have $\operatorname{supp}(x), \operatorname{supp}(y) \subset \operatorname{supp}(v)$ and v is thus convexly decomposable.

In the other direction we will show that convexly decomposable elements are lifted to convexly decomposable elements. This implies that a not convexly decomposable element in $\widetilde{\mathcal{F}}$

must be projected onto a not convexly decomposable element in $\mathcal{F}$ and hence $\pi(\tilde{\Omega}) \subseteq \Omega$. Let $v = \lambda x + (1-\lambda)y$ be a convex decomposition in $\mathcal{F}$, i.e., $\mathrm{supp}(x), \mathrm{supp}(y) \subset \mathrm{supp}(v)$. It is straightforward to verify that we have the corresponding convex combination $\tilde{v} = \lambda\tilde{x} + (1-\lambda)\tilde{y}$ of the lifted elements. Clearly, $\tilde{x} \neq \tilde{y}$ is implied by $x \neq y$. It just remains to show that $\mathrm{supp}(\tilde{x}), \mathrm{supp}(\tilde{y}) \subseteq \mathrm{supp}(\tilde{v})$. Since $\tilde{x}_u = \tilde{y}_u = \tilde{v}_u = 1$ and $\mathrm{supp}(x), \mathrm{supp}(y) \subset \mathrm{supp}(v)$ by assumption, it holds for the first $n+1$ reactions. For the last reactions, i.e., the leak reactions $l \in L$, it suffices to verify that the leak fluxes satisfy the implications $\tilde{v}_l = 0 \Rightarrow \tilde{x}_l = 0,\ \tilde{y}_l = 0$. If $\tilde{v}_l = 0$ is given, just note that the leak-fluxes are positive, i.e., $\tilde{x}_l, \tilde{y}_l \geq 0$. Together with $\tilde{v}_l = \lambda\tilde{x}_l + (1-\lambda)\tilde{y}_l$, $0 < \lambda < 1$, this implies $\tilde{x}_l, \tilde{y}_l = 0$. □

Lemma 3.25. *If v is a maximal element of the polytope F, then a convex combination $v = \sum_{i=1}^{k} \lambda_i w^i$, $\lambda_i \geq 0$, $\sum_{i=1}^{k} \lambda_i = 1$ implies that $w^1, \ldots, w^k$ are also maximal elements.*

Proof. Assume that the conditions of the lemma are given but not all w^i are maximal. Then, w.l.o.g. w^1 is not maximal and $\alpha w^1 \in F$ with $\alpha > 1$. Let $v' := \lambda_1(\alpha w^1) + \sum_{i=2}^{k} \lambda_i w^i$. This is a convex combination of elements of F and thus $v' \in F$. An alternative representation of v is given by

$$\alpha^{-1}v' + (1-\alpha^{-1})\sum_{i=2}^{k}\lambda_i w^i \quad = \quad \lambda_1 w^1 + \alpha^{-1}\sum_{i=2}^{k}\lambda_i w^i + (1-\alpha^{-1})\sum_{i=2}^{k}\lambda_i w^i \quad = \quad v.$$

The sum of the coefficients is

$$\alpha^{-1} + (1-\alpha^{-1})\sum_{i=2}^{k}\lambda_i \quad = \quad \alpha^{-1} + (1-\alpha^{-1})\cdot(1-\lambda_1) \quad = \quad 1 - \lambda_1 + \alpha^{-1}\lambda_1,$$

but this is strictly smaller than 1, since $\alpha^{-1} < 1$. We conclude that v is not maximal contrary to the assumption. □

3.9 Discussion

Summary. In this chapter we presented a mathematical model of resource allocation and an optimisation problem to find sequences of flux modes for most efficient production of metabolic output. The optimisation problem was considered in a very general form as well as in the special case of binary gene expression. All flux modes of the sequence are chosen from the same flux space, which is determined by the resource function representing the resource allocation model and by the gene expression state. With our derived resource allocation model and completely free gene expression $g \in [0,1]^n$, the resource allocation constraints reduce to one single constraint which coincides with the molecular crowding approach. If $G \subset [0,1]^n$ is convex, the problem is still rather simple, because we know then that also the flux space is convex and as a consequence, a sequence cannot be more efficient in output production than an optimal single flux mode. However, with binary gene expression the flux space is in general not convex as is illustrated by examples. The theoretical analysis is then focused on this binary gene expression model. Only few aspects concerning the biological meaning and justification of our modelling approach are mentioned, but these questions will be thoroughly discussed in Chap. 4. Since the number of binary gene expressions is finite, the flux space is the union of finitely many polytopes. The comparison of this flux space with the polytope that is obtained by neglecting the limited enzyme resources leads to a characterisation of so-called perfect pre-selections. These

are finite subsets of the flux space which are sufficient to constitute an optimal sequence for any demand Γ of output production. Due to the assumption that the resource function is monotone in the support of the flux modes, the decompositions play an important role and it turns out that a perfect pre-selection is given by the set of not convexly decomposable elements. This is a notion that was is introduced here and includes non-decomposable elements and elementary modes (EMs). We construct then an extended flux space whose EMs project to exactly the not convexly decomposable elements in the original flux space. This allows for using EM enumeration to generate a perfect pre-selection.

The detailed analysis of the flux space and optimal sequences was concentrated on finite gene expression spaces G. For infinite but non-convex G, the properties of the flux space and the question if a perfect pre-selection for optimal sequences can be found are not investigated. Also the possible differences between Ω and $\Omega_{p'}$ are not further examined. Since we require p' in Sec. 3.7 to be monotone in the support of the flux modes, it must be constant on all flux modes with a fixed support vector. Under this condition, the resource allocation model is therefore very close to the special case of binary gene expression, even if G is infinite. The monotonicity of p' in the support is thus implementing the assumed insensitivity of the metabolic reactions to the precise levels of gene expression.

The essential results of this chapter are based on a class of resource allocation models which give a flux space that is a conic sprout, i.e., it resembles locally at 0 the underlying flux cone. In particular the existence and the construction of the perfect pre-selection depends on this property. However, further biologically motivated constraints can destroy this property as is e.g. the case for maintenance constraints.

Practical implications. The approach to reduce the search space to a perfect pre-selection is interesting in general, because the MIQCP optimisation problem (p. 15) is reduced to an LP. In Chap. 6 the possibility to construct pre-selections for more general optimisation problems is investigated, which might not always give optimal solutions but only approximations. The theoretical analysis in this chapter yielded a construction of a perfect pre-selection, which means that an optimal solution is always obtained. This construction can be realised with help of EM enumeration as long as the extended network (which is enlarged by a factor of at most three but usually much smaller) is not too large. Once the set of not convexly decomposable elements has been determined, optimal solutions of (OP2a) can be computed for different objectives Γ and different model parameters $kc_j^{\pm}$, γ_j by solving a rather simple LP. This procedure of constructing the extended network, enumerating EMs, projecting into the original network and solving (LPps) with this pre-selection was applied to the core metabolic network presented in Chap. 4 without the maintenance constraints. An important requirement for this method is that the flux space is a conic sprout. If this is not the case, for example because the model contains maintenance constraints, one can try to restore this property by approximating the maintenance constraint $mb_j \leq v_j$ by a constraint $a^\top v \geq 0$ which couples the maintenance reaction to some uptake or output reactions that carry flux in every relevant flux mode. For example this constraint could be $\sum_{t\in T} v_t \leq cv_j$ enforcing a flux through j as soon as any of the target reactions carries flux. Since we optimise for fast output production, only those flux modes are relevant where this sum is rather big. Such a work around can unfortunately not be developed into a sound method for implementing approximated maintenance constraints in general, since the dependence on some reference reactions is a strong restriction, but it might be helpful in certain cases where the maintenance constraints can be well approximated.

Concluding, we can say that in basic models the question of optimal sequences can be answered by finite subsets of the flux space and practically solved by EM enumeration. A model is basic if it consists only of the steady-state assumption, irreversibilities of reactions and bounds on the flux rates which might be given by complex resource allocation models, but do not affect the flux space at 0. For a more general class of models, which need not to satisfy the requirements for the results in this chapter, some useful approaches for the practical computation of pre-selections are presented in Chap. 6.

4 Switching metabolic pathways as a means to optimise output production

4.1 Overview

The model of resource allocation that was proposed and analysed in the previous chapter will now be applied. This whole chapter is based on a project that was carried out in collaboration with Hermann-Georg Holzhütter and Sascha Bulik [Palinkas et al., 2015], who provided also the core metabolic network model given by Fig. 4.3 and Tab. 4.1. The motivation for the development of the resource allocation model and the computational method was to elucidate the control and impact of metabolic alterations, and to test whether switching between different metabolic phases can lead to a more efficient production of the required output than a single flux mode. The optimisation problem (OP2a), p. 55, based on the resource allocation model given by (3.7) will be solved for $l = 1, 2, \ldots, |T|$ with additional maintenance constraints and restrictions on the gene expression states. At first, only a single flux mode is allowed to fulfil the requirements, then the number of different flux modes that can be used consecutively is increased. Comparing the objective values gives us the gain in time achieved by operating in different metabolic phases. Also a reference case where all genes are activated is considered in order to determine the parameter A_{tot} of total amino acid amount. A minimal example will serve for analysis of the dependence of the gain on parameters in the resource allocation model and output requirements. Depending on the parameters, switching between different flux modes is mostly beneficial, but in certain cases the performance of a single flux mode is superior. The method is then applied to a core metabolic network of the central carbon metabolism, where a significant gain in time is achieved by switching. Repeating the computations with perturbed parameters of the resource allocation model shows that this gain is a robust feature of the model. We also analyse the trade-off between the capacity for output production against energy production and protection from reactive oxygen species. The energy that is required for switching the flux modes is taken into account a posteriori. The benefit achieved by switching turns out not to be significantly affected by these additional requirements.

4.2 Illustration with a minimal example

The presented minimal example will serve to illustrate how a sequence of different flux modes can increase the efficiency of output production in our resource allocation model. It allows describing analytically how the different parameters, i.e., the bounds and the output demand, determine the performance of a single flux mode and of a sequence of flux modes.

The minimal example is a simplistic metabolic network comprising three irreversible reactions (see Fig. 4.1). The network has one internal metabolite X, one uptake flux v_0, and two target fluxes v_1 and v_2, yielding the target metabolites P_1 and P_2. The corresponding required fluxes through the target reactions $T = \{1, 2\}$ are Γ_1 resp. Γ_2. We presuppose two MGSs (see Sec. 3.6.3), $\chi_1 := \{g_0, g_1\}$ and $\chi_2 := \{g_0, g_2\}$. The two associated MFMs are $w_1 = (v_0, v_1, 0)$ with $v_0, v_1 > 0$ and $w_2 = (v_0, 0, v_2)$ with $v_0, v_2 > 0$. As the maximal number of different flux modes required to minimise the production time of the demanded output cannot be larger than the number of different target metabolites (Prop. 3.10), the maximal number of different metabolic phases for this example is two and we thus have to compare four possible strategies, shown in Fig. 4.1. Strategy A defines the reference case, where all genes are constantly active. The other strategies assume that during production of the metabolic output the network switches between two phases. Strategy B consists in producing the two relevant products successively: First, the demanded amount of product P_1 is produced while the pathway for the production of P_2 is switched off. Then, the demanded amount of product P_2 is produced while the pathway for the production of P_1 is switched off. The switch between these two metabolic phases requires the complete degradation of the enzymes constituting the P_1-synthesizing pathway. Since rapidly proliferating cells (without S0-phase of the cell cycle) continuously accumulate biomass during growth without any significant degradation of proteins [Roostalu et al., 2008]. Strategy B should realistically apply to non-proliferating cells. In strategies C and D, during the first metabolic phase only one product is produced, and in the second metabolic phase both products are produced simultaneously. Also here switching between the two metabolic phases requires a partial degradation of enzymes of the initially active pathway in order to allocate protein to the second pathway. For this simple system an analytical solution of the optimization problem can be given.

4.2.1 Analysis of the minimal example

Strategy A: Single flux mode For strategy A, the reference case, the demanded output $\Gamma > 0$ is produced by a single flux mode v. We assume w.l.o.g. that $\tau v_1 = \Gamma_1, \tau v_2 = \Gamma_2$ and set $r := \Gamma_2/\Gamma_1 = v_2/v_1$. Using the steady-state assumption $v_0 = v_1 + v_2$, we obtain $v = (v_0, v_1, v_2)^\top = v_1 \cdot (1 + r, 1, r)^\top$. Thus, there is only the unknown v_1, which has to be maximised in order to minimise $\tau(1) = \Gamma_1/v_1$. For $v > 0$, we get the gene expression state $g = (1, 1, 1)^\top$. Setting the expression efficiencies to $\eta_j = 1$, $j = 1, 2, 3$, the upper bounds on the fluxes according to the resource allocation model (3.7) are given by

$$ub_j := A_{tot}\, kc_j^+ \frac{1}{\gamma_A + \langle g, \gamma \rangle} = A_{tot}\, kc_j^+ \frac{1}{\gamma_A + \gamma_0 + \gamma_1 + \gamma_2}, \text{ for } j = 0, 1, 2.$$

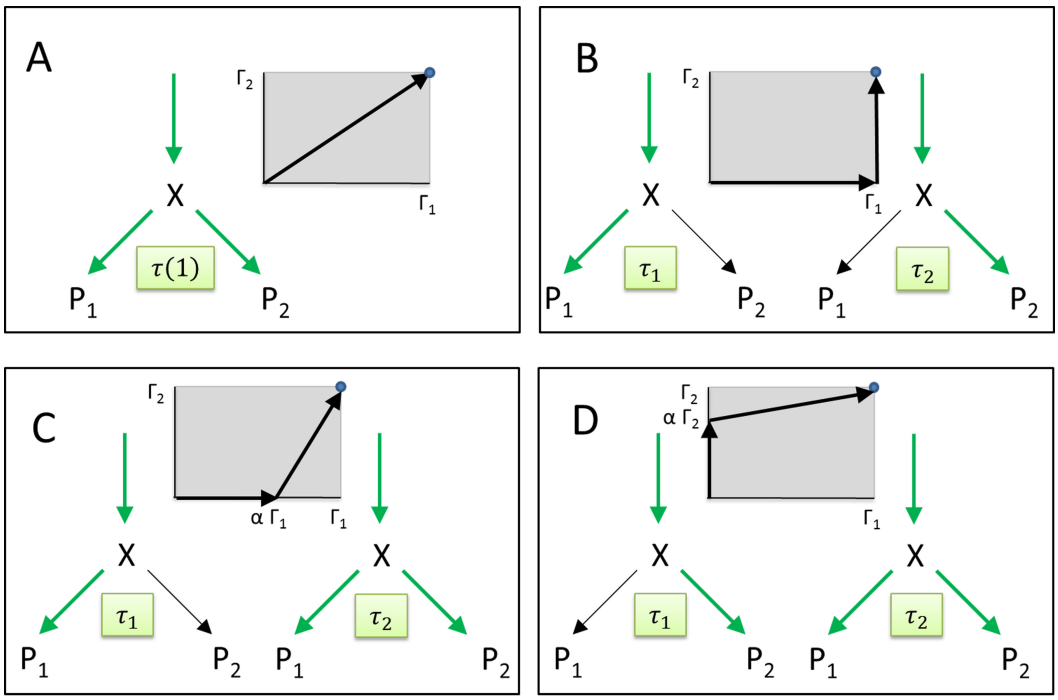


Figure 4.1: **Simplistic metabolic network with two target fluxes.** (Drawing provided by Hermann-Georg Holzhütter) Strategy A: All genes are constantly active, the demanded metabolic output is generated during the time interval τ_0 by a single flux mode composed of the two MFMs (see Sec. 3.6.3) w^1 and w^2. Strategy B: The two minimal gene sets are separately active, during the first time interval τ_1 only the demanded amount of product P_1 and in the second time interval τ_2 only the demanded amount of P_2 is produced. Strategy C: During the initial time interval τ_1 only the minimal gene set χ_1 is active and only a certain fraction $\alpha < 1$ of the demand for P_1 is produced. Thereafter, the second minimal gene set is additionally activated so that the products P_1 and P_2 are produced simultaneously. Strategy D: During the initial time interval τ_1 only the minimal gene set χ_2 is active, thereafter the second minimal gene set is additionally activated so that the products P_1 and P_2 are produced simultaneously. The grey-shaded panels illustrate the proportions in which the demanded amounts Γ_1 and Γ_2 of the two output metabolites are produced in strategies A-D.

Maximizing v_1 under the constraint $v \leq ub$, we obtain $v_1 = \min(ub_0/(1+r), ub_1, ub_2/r)$ or equivalently

$$\tau(1) = \frac{\Gamma_1}{v_1} = \max\left(\frac{\Gamma_1+\Gamma_2}{ub_0}, \frac{\Gamma_1}{ub_1}, \frac{\Gamma_2}{ub_2}\right). \tag{4.1}$$

Strategy B: Switching between two MinModes Next we consider the case where the two minimal gene sets χ_1, χ_2 are separately activated in two time intervals with flux modes w^1, w^2. Here w^1 is only producing the target metabolite P_1 and w^2 only P_2. Applying the steady state condition, we get $w^1 = (w_0^1, w_0^1, 0)^\top, w^2 = (w_0^2, 0, w_0^2)^\top$. For w^1, we have the upper bounds

$$ub_0^1 := A_{tot}\, kc_0^+ \frac{1}{\gamma_A+\gamma_0+\gamma_1}, \qquad ub_1^1 := A_{tot}\, kc_1^+ \frac{1}{\gamma_A+\gamma_0+\gamma_1}, \qquad ub_2^1 = 0,$$

whereas as for w^2 we get

$$ub_0^2 := A_{tot}\, kc_0^+ \frac{1}{\gamma_A + \gamma_0 + \gamma_2}, \qquad ub_1^2 = 0, \qquad ub_2^2 := A_{tot}\, kc_2^+ \frac{1}{\gamma_A + \gamma_0 + \gamma_2}.$$

Maximizing w_0^1 resp. w_0^2 under the constraint $w^1 \leq ub^1$ resp. $w^2 \leq ub^2$ yields

$$w^1 = \min(ub_0^1, ub_1^1)\begin{pmatrix}1\\1\\0\end{pmatrix} \text{ and } w^2 = \min(ub_0^2, ub_2^2)\begin{pmatrix}1\\0\\1\end{pmatrix}.$$

For the durations, we get

$$\tau_1 = \frac{\Gamma_1}{\min(ub_0^1, ub_1^1)} = \max\left(\frac{\Gamma_1}{ub_0^1}, \frac{\Gamma_1}{ub_1^1}\right) \text{ and } \tau_2 = \frac{\Gamma_2}{\min(ub_0^2, ub_2^2)} = \max\left(\frac{\Gamma_2}{ub_0^2}, \frac{\Gamma_2}{ub_2^2}\right). \tag{4.2}$$

Whether or not the solution using w^1 and w^2 outperforms the single flux vector v, i.e., whether or not $\tau_1 + \tau_2 < \tau(1)$ depends on the demand Γ and the upper bounds ub, ub^1, ub^2. We discuss two cases in more detail. First suppose ub_0 is small, such that $\tau(1) = (\Gamma_1 + \Gamma_2)/ub_0$ and $ub_0 < ub_1^1, ub_2^2$. It follows that $\tau_1 < \Gamma_1/ub_0$, $\tau_2 < \Gamma_2/ub_0$ and hence $\tau_1 + \tau_2 < \tau(1)$. In other words, switching from w^1 to w^2 is more efficient than the single flux mode v. Second, assume ub_0 is large, such that w.l.o.g. $\tau(1) = \Gamma_1/ub_1 \geq \Gamma_2/ub_2$. In particular, by (4.1) we have $(\Gamma_1 + \Gamma_2)/ub_0 \leq \Gamma_i/ub_i$, which implies $\Gamma_i/ub_0 \leq \Gamma_i/ub_i$, for $i = 1, 2$. Since $ub_0 \geq ub_i \Leftrightarrow ub_0^i \geq ub_i^i$ and using (4.2), we get $\tau_i = \Gamma_i/ub_i^i$, for $i = 1, 2$. The switching solution thus has the duration $\tau_1 + \tau_2 = \Gamma_1/ub_1^1 + \Gamma_2/ub_2^2$. As long as Γ_2/ub_2^2 is not extremely small, this will be larger than $\tau(1) = \Gamma_1/ub_1$, the duration of the single mode solution. Taking a closer look at the ratio Γ_1/Γ_2, we observe that a smaller value of Γ_1 and a larger value of Γ_2 are favourable for the single mode solution. On the one hand, increasing Γ_1 by a factor $c > 1$ increases also $\tau(1)$ by c, whereas $\tau_1 + \tau_2$ increases by a strictly smaller factor (as long as $\Gamma_2 > 0$). On the other hand, decreasing Γ_2 has no effect on $\tau(1)$, while the duration of the switching solution is decreased. We conclude that the single mode solution performs best compared to the switching solution, i.e., $\tau(1)/(\tau_1 + \tau_2)$ is minimal, if we have equality in our assumption, i.e., $\Gamma_1/ub_1 = \Gamma_2/ub_2$, or equivalently $\Gamma_1/\Gamma_2 = ub_1/ub_2$.

Example With $A_{tot} = 1.8 \cdot 10^6$, equal expression efficiencies ($\eta_j = 1$, $j = 0, 1, 2$) and molecular weights ($\gamma_j = 60,000$, $j = 0, 1, 2$) and catalytic constants $kc_0^+ = 10\,h^{-1}, kc_1^+ = kc_2^+ = 1\,[h^{-1}]$, using the approximation in (3.7), i.e., neglecting $\gamma_A = 126$ in the denominator, the upper boundaries of strategies A and B read:

$$\begin{array}{lllll}
(A) & ub_0 = 100, & ub_1 = 10, & ub_2 = 10 & \text{for} \quad v^1 = (v_0^1, v_1^1, v_2^1)^\top \text{ (reference case)} \\
(B) & ub_0^1 = 150, & ub_1^1 = 15, & ub_2^1 = 0 & \text{for} \quad w^1 = (w_0^1, w_1^1, 0)^\top \\
 & ub_0^2 = 150, & ub_1^2 = 0, & ub_2^2 = 15 & \text{for} \quad w^2 = (w_0^2, 0, w_2^2)^\top
\end{array}$$

For the production of $\Gamma_1 = 10$ units of P_1 and $\Gamma_2 = 50$ units of P_2 a production time of $\tau = 10$ [h] is needed in case (A) and only $\tau = 6.67$ [h] in case (B). Note that for this concrete example, the concept of perfect gene regulation [Beg et al., 2007] would allow an even shorter production time of $\tau = 2.2$ [h] to be achievable with a single optimal flux mode v. As we just saw above, this v is fixed up to scaling by some $\lambda > 0$, in this case $v = (v_0, v_1, v_2)^\top = \lambda \cdot (6, 1, 5)^\top$ [(mol/gDW)/h].

Perfect gene regulation would mean that all enzymes work at their maximal capacity, i.e., $ub = \lambda \cdot (6, 1, 5)^\top$ [(mol/gDW)/h]. With the given values for the molecular weights and the catalytic constants this is achieved by choosing the gene expressions $g \in [0, 1]^3$ to be $g_0 = 0.12$, $g_1 = 0.2$ and $g_2 = 1$. The resulting upper bounds are then $ub = (ub_0, ub_1, ub_2)^\top = 4.545 \cdot (6, 1, 5)^\top = v$.

We will now have a look at the effect that the parameters Γ and kc^+ have on the performance of the strategies A and B. For $\Gamma_1 + \Gamma_2 = 100$ units, the optimal solution with strategy B always needs $\tau_1 + \tau_2 = 6.67$ h, whereas the time needed with a single flux mode (strategy A) varies between $\tau(1) = 5$ [h] in the case $\Gamma_1 = \Gamma_2 = 50$ units to $\tau(1) = 9$ [h] for $\Gamma_1 = 90$, $\Gamma_2 = 10$ units. When we fix $\Gamma_1 = \Gamma_2 = 50$ units and vary instead the catalytic constants kc_1^+, kc_2^+, strategy B becomes preferable if the constants are sufficiently different. For example, with $kc_1^+ = 0.5$ h^{-1} and $kc_2^+ = 5$ h^{-1} we get for strategy A $ub_1 = 5$ and $ub_2 = 50$ [(mol/gDW)/h] and therefore $\tau(1) = 10$ [h]. For strategy B the bounds are increased by the factor $3/2$ or set to zero and they allow production of Γ in a time of $\tau_1 + \tau_2 = 6.67$ [h] + 0.67 [h] = 7.33 [h] only.

Strategies C or D

Between these two extreme strategies to produce all demanded output on the one hand in a single flux mode and on the other hand with two flux modes, of which each produces only one output, there are the strategies C and D. Strategy A can be seen as the limit case of strategy C or D, when τ_2 goes to zero. On the other side strategy B is then the limit case of C or D, when v_1^2 resp. v_2^2 vanishes. In general, these mixed strategies can significantly increase the efficiency compared to Strategy B, as is illustrated in Fig. 4.2.

4.3 A core metabolic network of the central carbon metabolism

As a biologically meaningful application of our approach, a core metabolic network model is introduced. Based on parameters from the literature, we estimate the minimum total protein amount required to fulfil maintenance and biomass production and compute minimal functional flux modes and gene sets. We also take a look at the balance between the costs for biomass production and maintenance constraints. After this, the performance of sequences and single flux modes can be compared in the next section.

A core metabolic network of the cellular carbon metabolism is given by the reaction scheme Fig. 4.3. It comprises as main metabolic pathways glycogenesis, glycolysis and gluconeogenesis, the pentose phosphate cycle composed of the oxidative and non-oxidative branch, the synthesis of triglycerides and the oxidative energy metabolism. The citric acid cycle, the respiratory chain and the synthesis of free fatty acids and triglycerides are only represented by lumped overall reactions. The considered final output of the network is the production of four macromolecules which are central for maintaining the integrity of the cell and which in dividing cells have additionally to accumulate in the growth phase (G1 of the cell cycle) before cell division: Synthesis of glycogen (an important carbohydrate store), nucleic acids (RNA + DNA), triglycerides (an important energy store), and proteins. The cellular network can exchange oxygen, carbon dioxide and the metabolites glucose and lactate with the environment. The rate of the membrane transporters for glucose and lactate is subject to the same resource allocation

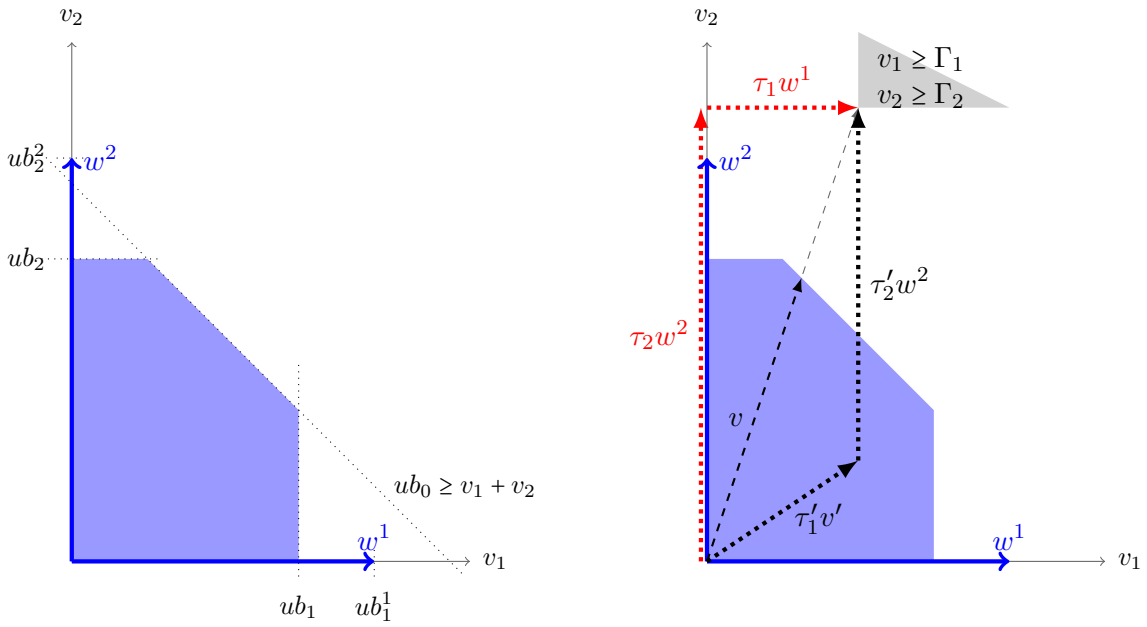


Figure 4.2: Projection of the flux space of the minimal example on the reactions 1 and 2. The grey region defines the demand. We consider the network with $A_{tot} = 40$, $\gamma_0 = 2$, $\gamma_1 = \gamma_2 = 1$ and use the approximated version of (3.7) without γ_A. The turnover numbers of the reactions are $kc^+ = (5, 3, 4)^\top$ and $kc^- = (0, 0, 0)^\top$. The bounds are then $ub = A_{tot} kc^+ \frac{1}{4} = (50, 30, 40)^\top$ if all three reactions are used. For the case that only two reactions (reactions 0 and 1 or reactions 0 and 2) are used, we have $ub^1 = ub^2 = \frac{4}{3} ub = \frac{1}{3} (200, 120, 160)^\top$. The demand is $\Gamma_1 = 20$, $\Gamma_2 = 60$. The optimal single flux mode to satisfy the demand is $v = (50, 12.5, 37.5)^\top$, with the duration $\tau = 1.6$ (dashed arrow). Using strategy B, we get the two MFMs $w^1 = 40 \cdot (1, 1, 0)^\top$, $w^2 = \frac{160}{3} \cdot (1, 0, 1)^\top$ and a solution with durations $\tau_1 = 0.5$ and $\tau_2 = \frac{9}{8} = 1.125$, summing up to 1.625 (red dotted arrows). An optimal solution is the combination of $v' = (5, 3, 2)^\top$ with $\tau_1' = \frac{2}{3}$ and the MFM $w^2 = \frac{160}{3} \cdot (1, 0, 1)^\top$ with $\tau_2' = \frac{7}{8} = 0.875$, summing up to a total time of 1.542 (black dotted arrows).

constraints as all other enzymatic reactions. Only the exchange of oxygen and carbon dioxide as well as the artificial reactions ATPase and GSHox which represent energy consumption and glutathione reduction, respectively, are excluded from the resource allocation model.

Some of the metabolic objectives of a network have to be permanently fulfilled during the whole life cycle of a cell, and thus cannot be temporarily switched off. These requirements are implemented by maintenance constraints, see Sec. 3.7.2. For the metabolic network of Fig. 4.3, the maintenance reactions are ATPase and GSHox, producing the metabolites ATP and GSH, respectively. The anti-oxidant GSH protects the cell from reactive radicals and has to be continuously replenished from GSSG. Furthermore, besides the ATP consuming processes utilised by reactions that explicitly occur in the network, a certain fraction of ATP is continuously utilised (termed +ATP utilisation in Table 4.1) to maintain essential cellular processes as, for example, active membrane transport or cell motion. Table 4.1 quantifies for an average human cell type the demanded output and the brutto reactions relating the metabolites produced in the network to the output of macromolecules. The prescribed fluxes through the target reactions GSSG reduction and surplus ATP production convert into the quantities of ATP and GSH that have to be

obligatorily produced in any time interval, implemented by (3.25), page 67.

Formulating the optimisation problem The optimisation problem to minimise the production time for the four target metabolites glycogen, protein, nucleic acids and lipids (triglycerides) thus has to be solved by taking into account the possible consumption of the two alternative substrates glucose and lactate and the two indispensable maintenance reactions ATP consumption and GSSG reduction (see Table 4.1). We used here the basic optimisation problem (OP3), p. 58, with the resource allocation function p defined by (3.10) and extended it with two maintenance constraints, namely $v_j^k \geq mb_j$ for $j \in \{\text{ATPase}, \text{GSHox}\}$ and $k = 1, \ldots, l$. The space of available gene expression states $G \subsetneqq \{0,1\}^n$ is constituted by combinations of minimal gene sets (MGS) which are based on minimal flux modes (MFM) computed for the production of the output under different conditions as described in Sec. 4.3.1 below. This restriction of the gene expression space is implemented by linear integer constraints as explained in Sec. 3.6.3. The resulting optimisation problem will further on just be called $(\mathscr{P}_0)$. In Sec. 4.4.1 the sequence of gene expression states will be additionally constrained to be monotone, which is also implemented by linear integer constraints, see Sec. 3.6.3. The optimisation problem with this additional restriction will be called $(\mathscr{P}_1)$ in this chapter. For all computations these optimisation problems are solved with the Gurobi Optimizer 5.6 (`http://www.gurobi.com`) via Matlab. All computations were carried out with and without the approximation in the resource allocation model (3.10). The numerical results were identical in all cases. The eligibility of the approximation is hence confirmed and we do not have to distinguish between the original and the approximated version in this chapter.

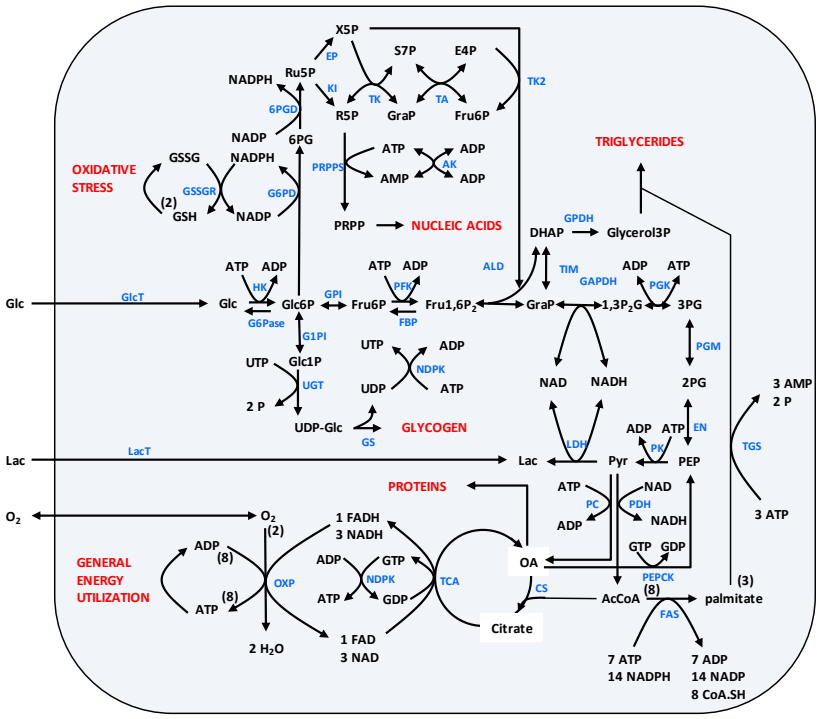


Figure 4.3: **Scheme of the network of the central carbon metabolism, with glucose or lactate as substrates.** (Drawing provided by Hermann-Georg Holzhütter.) The metabolic output, i.e., production of triglycerides, nucleic acids, proteins and glycogen are shown in red as well as the permanent energy consumption (ATPase) and oxidative stress (GSHox). The metabolites are Lactate (Lac), Glucose (Glc), Glucose-6-*P* (Glc6P), Fructose-6-*P* (Fru6P), Fructose-1,6-bisphosphate (Fru1,6P2), Glucose-1-*P* (Glc1P), UDP-Glucose (UDP-Glc), Dihydroxyacetone phosphate (DHAP), Glyceraldehyde phosphate (GraP), 1,3-Bisphosphoglycerate (1,3P2G), 3-Phosphoglycerate (3PG), 2-Phosphoglycerate (2PG), Phosphoenolpyruvate (PEP), Pyruvate (Pyr), Oxalacetate (OA), Acetyl-Coenzym-A (ACoA), Gluconate-6-*P* (6PG), Ribulose-5-*P* (Ru5P), Xylulose-5-*P* (X5P), Ribose-5-*P* (R5P), Sedoheptulose-7-*P* (S7P), Erythrose-4-*P* (E4P) and Phosphoribosyl pyrophosphate (PRPP).

Metabolic output of the reaction network			
Target metabolite	biomass reaction for producing 1 unit of target molecule	demanded units	flux units Γ_t
Glycogen	50,000 UDP-Glc → 1 glycogen + 50,000 UDP	$5.6 \cdot 10^{-6}$ mmol/gDW [1)]	0.28
Triglycerides	3 palmitate + 1 glycerol-3-P + 3 ATP → 1 triglyceride + 3 AMP	0.2 mmol/gDW [2)]	0.2
Proteins	400 OA + 1600 ATP → 1 protein + 400 AMP + 1200 ADP	0.016 mmol/gDW [3)]	6.4
Nucleic acids	RNA: 3000 PRPP + 21.000 ATP → 1 RNA + 21.000 ADP DNA: $6{\cdot}10^9$ PRPP + $42{\cdot}10^9$ ATP → 1 DNA $+42 \cdot 10^9$ ADP	$4.48{\cdot}10^{-5}$ mmol/gDW [4)] $5.3{\cdot}10^{-13}$ mmol/gDW [4)]	0.13 + 0.032 = 0.162
GSSG reduction	1GSSG + NADPH → 2 GSH + NADP	0.002 (mmol/gDW)/h [5)]	
+ATP utilisation	ATP → ADP	5 (mmol/gDW)/h [6)]	

1. Average number of glucose moieties in a glycogen molecule = $50,000$. Average MW of glycogen = 70 [μg/mg] protein [Rousset et al., 1981].
2. 0.167 [g/gDW] lipid [Shlomi et al., 2011b]. Average MW per triglyceride = $176 + 42\,n$ [Da] with n = length of fatty acids. With n = 16 (palamitate) MW = 848 [Da].
3. 0.78 [g/gDW] protein [Shlomi et al., 2011b]. Average size of protein = 400 amino acids. Average MW per amino acid = 126 [Da]. As amino acids are not included into the network model the metabolite oxaloacetate (OA) involved in the transamination of many amino acids is used here as a place holder, i.e., the consumption of amino acids for protein synthesis equals the consumption of OA.
4. DNA: 0.0103 [g/gDW] [Shlomi et al., 2011b]. Length DNA (double strand) = $6 \cdot 10^9$ nucleotides. Average MW of single nucleotide = 325 [Da]. RNA: 0.0437 [g/gDW] [Shlomi et al., 2011b]. Average length RNA = 3000 nucleotides.
5. GSSG reduction rate in erythrocytes representing a cell type with a high oxidative load [Schuster and Holzhütter, 1995].
6. Value chosen such that 40 % of total ATP utilisation is spent on active membrane e processes (predominantly Na-K-ATPase).

Table 4.1: Demand of output metabolites and associated energy consumption. The amounts in the third column are based on literature. In the fourth column, the corresponding flux units through the target reactions are given, they define the demand vector Γ for the computations. These values are obtained by scaling the demanded units according to the stoichiometry of the network model. For example, the target reaction for glycogen synthesis consumes 1 UDP-Glc which gives a factor of $50,000$ for the demanded units and we get $50,000 \cdot 5.6 \cdot 10^{-6} = 0.28$ [mmol/gDW]. Since the nucleic acid synthesis is subsumed in one reaction in our model, the sum of RNA and DNA demand gives the value of Γ. For the maintenance reactions ATPase and GSHox, minimal flux rates are given which need to be attained permanently.

In Tab. 4.1 the values of Γ_t are given in the rightmost column for the four target reactions $t \in T$ representing synthesis of glycogen, triglycerides, proteins and nucleic acids. The overall reactions GSHox and ATPase are defined as maintenance reactions, the minimum maintenance flux rates are 0.002 [(mmol/gDW)/h] for GSHox and 5 [(mmol/gDW)/h] for ATPase.

4.3.1 Determination of minimal gene sets

For each of the two substrates glucose or lactate, we computed 6 minimal gene sets (MSGs), each corresponding to a minimal flux mode (MFM) producing the maximal amount of one of the 4 target metabolites or maximising one of the two maintenance fluxes (see Sec. 3.6.3). The maintenance constraints are not imposed in these computations. The MFMs were obtained by (3.21), that is, maximising the flux through one of the target reactions or maintenance reactions over the flux space given by the steady-state assumption and the bounds (3.7) of the resource allocation model. The single uptake reaction u refers either to glucose or to lactate uptake. To these 12 MGSs we added another group corresponding to 8 MFMs which maximise the flux through the four target reactions yielding glycogen, protein, nucleic acids and triglycerides while maintaining the indispensable target fluxes. This means that the maintenance constraints $v_j \geq mb_j$ for $j \in \{\text{GSHox, ATPase}\}$ were added to the optimisation (3.21) which was then carried out as before for the four target reactions and one of the substrates. The resulting 20 MGSs were used to constrain the gene expression states, i.e., the simultaneous activation and inactivation of genes when solving the optimisation problem. These constraints were implemented by linear constraints as described in Sec. 3.6.3.

4.3.2 Specification of turnover rates and molecular masses

Numerical values for the turnover rates $kc_j^{\pm}$ and molecular weights γ_j were taken from the BRENDA data base [Schomburg et al., 2002, `http://www.brenda-enzymes.org/`].

Molecular weights The diffusion transport of O_2 and CO_2 as well as the maintenance reactions GSHox and ATPase and the synthesis reactions of the target metabolites are excluded from the resource allocation model by setting the molecular weights to $\gamma_j = 0$ (cf. p. 45). These reactions were furthermore assumed not to be rate limiting and the corresponding turnover numbers therefore set to ∞. The upper bounds of these reactions are then ∞, independently of the gene expression state. For the reversible reactions O_2 and CO_2 the lower bounds are $-\infty$, but for the irreversible synthesis reactions as well as for GSHox and ATPase the lower bounds are 0. All other reactions of the network are subject to the resource allocation model. Their molecular weights could be obtained from BRENDA. The molecular masses assigned to lumped reactions were taken as the sum of the molecular masses of the involved individual enzymes.

Turnover numbers Transport rates for glucose and lactate were taken from [Goodyear et al., 1991, Hertz and Dienel, 2005], respectively. For the overall reaction 'citric acid cycle' (TCA), we took the turnover number of the rate limiting enzyme isocitrate dehydrogenase. The fatty acid synthesis, FS, was assigned a turnover number of 43 s^{-1}, according to [Cox and Hammes, 1983] and oxidative phosphorylation was assigned a turnover rate of 80 s^{-1}, which is the minimum of the values for the individual reactions, as retrieved from BRENDA. For Glucose-6-P-dehydrogenase we assigned a turnover rate of 14 s^{-1} according to [Topham et al., 1986]. The few reactions where no turnover number was found were assumed not to be rate limiting for

the network, the respective fluxes are hence unbounded. Formally, kc_j^+ is set to ∞ for these reactions j and if j is reversible $kc_j^- = -\infty$, otherwise $kc_j^- = 0$.

4.3.3 Specification of the reference case

To define the reference case where the network operates with a single stationary flux mode, in which all genes are constantly active and thus all enzymes are constantly expressed, the value of the total mass of amino acids, A_{tot}, was chosen such that the minimal time for the production of the demanded metabolic output given in Table 4.2 was 8 hours. The times needed for biomass production decrease monotonously with increasing A_{tot}. A value of 8 [h] for the biomass duplication time lies between doubling times reported for yeast cells (1.25-2 hours) and cancer cells in culture derived from metastatic tumors (~ 24 hours). Slightly different values of A_{tot} were obtained depending on the availability of the substrates glucose and lactate (see Table 4.2). The critical amount of amino acids A_{tot}^{crit} which is just sufficient to produce the flux through the two maintenance reactions (GSHox and ATPase) without required production of the four biomass components is about $A_{tot}^{crit} = 0.023$ [mg/gDW]. This means that 20% to 27% of the total available amino acids pool has to be spent on pathways enabling these two permanently active maintenance reactions (see Table 4.2). Under challenging conditions, the flux through the maintenance reactions may even increase, e.g. if the cell is exposed to a higher osmotic pressure that necessitates the activation of membranous ATP-depending ion pumps to preserve the osmotic equilibrium, or at a higher load of reactive oxygen species (peroxides) enhancing the utilisation of GSH. In such situations, at a fixed value of A_{tot}, an increase of the flux through the maintenance reactions can result in a slower production of the biomass components. The increase in production time under higher maintenance requirements revealed a weak and linear sensitivity to an increase of GSH oxidation minimum flux, increasing the production time only marginally. Higher permanent ATP demand has no effect as long it does not cross a value of circa 11 [(mmol/gDW)/h]. However, as this threshold is surpassed the production time increases drastically (see Fig. 4.4).

Computation of protein amounts Formally, A_{tot}^{crit} is defined as the minimal protein amount α that enables a flux mode $v \in \mathcal{F}^*$ with maintenance constraints $v_j \geq mb_j$ for $j \in \{\text{GSHox, ATPase}\}$. All genes are assumed to be activated, hence the resource allocation term is fixed to a constant, $p_j(v) = r := \left(\sum_{j=1}^n \gamma_j\right)^{-1}$. Therefore, $v \in \mathcal{F}^*$ is equivalent to $Sv = 0$ and $\alpha \cdot kc_j^- r \leq v \leq \alpha \cdot kc_j^+ r$. By substituting $w := \alpha^{-1} v$, these constraints are equivalent to $Sw = 0$, $kc_j^- r \leq w \leq kc_j^+ r$ and $\alpha \cdot w \geq mb$. This gives the following optimisation problem, whose objective value is A_{tot}^{crit} by definition:

$$\begin{aligned} &\min \alpha, \text{ subject to:} \\ &Sw = 0 \\ &kc^- r \leq w \leq kc^+ r \\ &\alpha \cdot w \geq mb \\ &\text{with variables: } \alpha \in \mathbb{R}_{\geq 0},\ w \in \mathbb{R}^n \end{aligned} \tag{4.3}$$

In fact, a close look at (OP3), i.e., $(\mathscr{P}_0)$ without maintenance constraints, reveals that the durations τ_k depend linearly on A_{tot}. Let the optimal solution $v^1, \ldots v^l$, $\tau_1, \ldots, \tau_l$ of (OP3), p. 58, be given with any $A_{tot} > 0$. For a different $A'_{tot} > 0$, we get an optimal solution by scaling the durations with A_{tot}/A'_{tot}, as follows directly by substituting in the optimisation problem. To

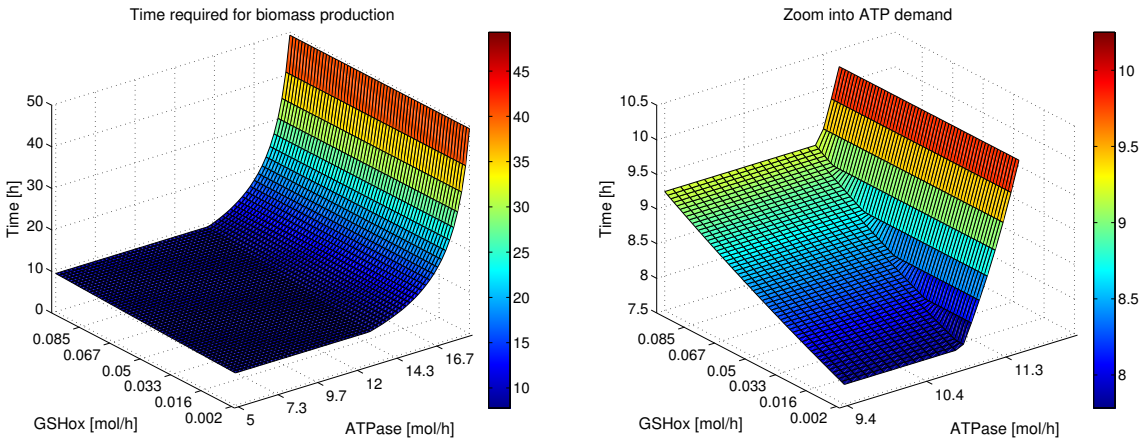


Figure 4.4: **Dependence of the minimal biomass production time in the reference case (all genes active) on the magnitude of the flux through the maintenance reactions.** The fluxes through the maintenance reactions were increased. GSHox flux was increased up to 50-fold and ATPase flux 4-fold of their normal values. The surface starts at the bottom with the minimal production time of 8 [h] with all genes active and maintenance demand of 0.002, 5 [(mmol/gDW)/h], see Table 4.1. Only if ATP consumption by the maintenance reaction is increased by a factor ≥ 2.5, the minimal production time is prolonged. This is due to the fact that fulfilment of the metabolic objectives requires ATP production in all metabolic phases. As long as the responsible reactions are not rate limiting, i.e., their catalysing enzymes do not operate at the upper flux bound, the rate of ATP synthesis can be increased to balance the additional ATP demand of the maintenance reactions up to an increase to the 2.5-fold of the normal. Below this threshold, only GSSG reduction acts as a bottleneck for biomass production. With 15 [(mmol/gDW)/h] of ATP consumption the time for production becomes 13.1 [h] and if the consumption rate tends towards 18.7 [(mmol/gDW)/h], the total available amount of amino acids has to be allocated to the ATP-producing flux mode and *de novo* production of biomass is not possible anymore. In contrast, variations of the flux through the GSH oxidase reaction have only little impact on the minimal production time. An even 5-fold higher rate of GSH oxidation prolongs the minimal production time by only 0.06 h.

compute A_{tot}, we also have to minimise a protein amount. The requirements are in this case not only the fulfilment of the maintenance $v \geq mb$, but also that the production of the demanded output is achieved in a given time, namely $\tau = 8$ [h]. Formally, $\tau \cdot v_t = 8v_t \geq \Gamma_t$, for $t \in T$. The substitution of w as above gives $w \geq \tau^{-1}\Gamma$. So A_{tot} is the objective value of the following optimisation problem, where $\tau^{-1} = \frac{1}{8}$.

$$
\begin{aligned}
&\min \alpha, \text{ subject to:} \\
&Sw = 0 \\
&kc^{-}r \leq w \leq kc^{+}r \\
&\alpha \cdot w \geq mb \\
&\alpha \cdot w_t \geq \tau^{-1}\Gamma_t \text{ for } t \in T \\
&\text{with variables: } \alpha \in \mathbb{R}_{\geq 0},\ w \in \mathbb{R}^n
\end{aligned}
\tag{4.4}
$$

Note that the products of the variables $\alpha \cdot w_t$ in (4.3) and (4.4) can be eliminated as explained on page 38.

To compute the required time if the protein amount is fixed to A_{tot} and the maintenance bounds are increased, see Fig. 4.4, the roles of τ and α are interchanged in (4.4): The duration τ is the variable and is minimised (or, equivalently, τ^{-1} is the variable and is maximised), while α is a fixed parameter.

substrate	A_{tot}^{crit} in mg/gDW	A_{tot} in mg/gDW
glc and lac	0.023	0.087
glc	0.023	0.115
lac	0.024	0.087

Table 4.2: Minimal total molecular mass of amino acids A_{tot} required to produce the metabolic output within 8 [h] at different conditions of substrate supply. A_{tot}^{crit} represents the minimal total molecular mass of amino acids required for only fulfilling the maintenance constraints. The computations were done under the reference conditions where all genes are activated.

4.4 Performance of optimal sequences compared to single flux modes

Applying the resource allocation model and the optimisation problems developed in the previous chapter with the parameters presented and derived in the previous section, we will now analyse the performance of sequences of different flux modes in producing metabolic output. The results show that switching yields a significant gain in production time under all substrate conditions. To make sure that this is not the result of opportune parameters, the robustness of these results is confirmed by repeating the computations with sampled parameters. Furthermore, we estimate the energy cost for switching in order to verify that this cost is not cancelling the achieved gain in efficiency.

After the computation of the total amount of amino acids A_{tot} and the MGS, we investigated whether switching between different sets of active genes may significantly reduce the minimal production time of the metabolic output. This was done by solving the optimisation problem $(\mathscr{P}_0)$, that is, (OP3), p. 58, with additional maintenance constraints and the gene expression space G generated by the MGS. Furthermore, all computations were also carried out for $(\mathscr{P}_1)$ which is obtained by adding the constraint that genes can only be switched on from one phase to the next. The numerical values of the expression efficiencies were put to $\eta_j = 1$ for all enzymes, $j = 1, \dots n$. Hence, the abundance of enzymes according to expression (3.7) is only controlled by the number and molecular masses of active genes in the model. According to Prop. 3.10, an optimal solution of the minimisation problem does not require more phases than there are different target reactions, i.e., $l \leq |T|$ can be assumed. Thus we fixed the maximal number of phases where different sets of genes are active to $l = 4$ and solved the optimisation problem for an increasing number of phases, $l = 1, 2, 3, 4$. Note that the case $l = 1$ is not identical with the reference case. Solving the optimisation problem with $l = 1$ allows inactivating parts of the

network, whereas all genes are active in the reference case. By $\tau(l)$ we denote the total time needed with an optimal solution using at most l phases. As before, τ_k denotes the duration in the k-th phase of a solution. The times $\tau(1)$ are notably below 8 [h], see Tables 4.3 and 4.4. The minimal production times obtained when the number of possible phases was increased stepwise from $l = 1$ to $l = 4$ are depicted in Table 4.3. The flux distributions of the optimal solutions with $l = 4$ are shown in Figs. 4.6 to 4.8 for the three different substrate conditions. In Tab. 4.3 we see that the most significant drop of the minimal production time was already obtained by allowing two phases, i.e., switching the network once between two different steady states. A larger number of phases resulted only in a marginal further improvement. The precise dependence of the gain on the number of switched genes can be determined by another computational approach which is presented in Sec. 6.7, see Fig. 6.4 therein. If both substrates are available, the best solution with a minimal production time $\tau = 4.78$ [h] was obtained with three switches between $l = 4$ phases. The relative proportions of the biomass components produced within the four phases are shown in in Fig. 4.5 and the corresponding flux modes are visualised in Fig. 4.6.

The sequence of phases is arbitrary. However, since the degradation of proteins and the *de novo* synthesis of mRNA and proteins consumes energy and other metabolic resources, we ordered them such that the cost for newly synthesised protein in the transitions between the different activity states in consecutive time intervals becomes minimal (the cost for synthesis was computed according to (4.5) in Sec. 4.4.2 below).

The results of the computations, i.e., the optimal solutions of $(\mathscr{P}_0)$ show that the rate of synthesis of the four biomass components should vary in different phases. In particular, the synthesis of glycogen and nucleic acids (dominated by the *de novo* synthesis of DNA) is predicted to occur only in short phases, whereas the production of the more abundant components (lipids and proteins) occurs in more than one phase. The optimal solutions depend critically on the availability of substrates. If, for example, only glucose is available, the total time interval during which the genes related to protein synthesis are active is longer than in a situation where both glucose and lactate can be used (see Fig. 4.5 A,B). Interestingly, if the two substrates glucose and lactate are both available, they are used differently within the four phases. During the synthesis of nucleic acids and glycogen, both substrates are used in parallel, but in the last phase, where the majority of protein is synthesised, lactate serves as the only substrate.

4.4.1 Minimising biomass production time by successive activation of genes

The optimal solutions in the preceding section were obtained by allowing genes and the related enzymes to be switched on and off in different metabolic phases. This is an unlikely situation in rapidly dividing cells, which run quickly through the G1-phase of the cell cycle without a notable degradation of proteins [Roostalu et al., 2008]. To account for this situation, all calculations were repeated with the additional constraint that genes can only be progressively turned on. Nevertheless, a partial degradation of enzymes of the preceding phase is still required. This is because the resource allocation model distributes in every phase all available proteins among the active reactions and degradation is hence necessary to make amino acids available for the synthesis of the additionally activated enzymes. The constraints $g^i \leq g^{i+1}$ for $i = 1, \ldots, l-1$ are added to $(\mathscr{P}_0)$ and we obtain the optimisation problem $(\mathscr{P}_1)$. In fact, it would make no difference if the monotonicity was imposed on the MGS activation instead of the gene activation. The constraints $g^i \leq g^{i+1}$, $i = 1, \ldots, l-1$ are equivalent to the similar constraints $b^i \leq b^{i+1}$, $i = 1, \ldots, l-1$, where $b^i \in \{0,1\}^s$ are the indicator variables of the MGS $\chi^1, \ldots, \chi^s$ which are

Switching MGS On and Off									
sub-strates	l	$\tau(l)$ [h]	τ_1 [h]	τ_2 [h]	τ_3 [h]	τ_4 [h]	# gene switches	# active genes	
glc, lac	1	7.344					0	41	(i)
	2	5.394	2.815	2.580			10	33, 43	(ii)
	3	4.788	2.434	0.177	2.177		28 (14+14)	31, 41, 35	(iii)
	4	4.774	2.451	2.090	0.099	0.134	31 (12+10+9)	31, 35, 37, 40	(iv)
glc	1	7.513					0	41	(v)
	2	5.330	2.113	3.217			15	38, 33	(vi)
	3	4.812	3.217	1.439	0.157		15 (9+6)	33, 32, 38	(vii)
	4	4.794	0.095	0.057	1.425	3.217	17 (6+2+9)	36, 34, 32, 33	$(viii)$
lac	1	7.344					0	41	(ix)
	2	5.929	0.706	5.222			6	41, 35	(x)
	3	5.857	5.222	0.202	0.433		8 (2+6)	35, 37, 39	(xi)
	4	5.848	3.052	2.161	0.202	0.433	10 (2+2+6)	33, 35, 37, 39	(xii)

Table 4.3: MGS can be switched on and off from one phase to the next. Times of optimal solutions with $1, 2, 3$ and 4 phases. The flux modes of solutions (iv), $(viii)$ and (xii) are visualised in the network in Figs. 4.6 to 4.8.

active in the i-th phase, see Sec. 3.6.3. The results are listed in Table 4.4. The gain, i.e., the relative reduction of biomass production times was only marginally lower than in the preceding computations, where gene sets could also be switched off in subsequent phases. Remarkably, although the additional constraint that genes can only be turned on results in a smaller total number of gene switches (see Tables 4.3 and 4.4), the predicted four phases of biomass production are very similar to those obtained if genes are allowed to be turned on and off (see Fig. 4.5).

Switching MGS only On									
sub-strates	l	$\tau(l)$ [h]	τ_1 [h]	τ_2 [h]	τ_3 [h]	τ_4 [h]	# gene switches	# active genes	
glc, lac	1	7.344					0	41	(i)
	2	5.394	2.580	2.815			10	43, 33	(ii)
	3	4.850	2.053	0.180	2.618		18 (6+12)	37, 43, 31	(iii)
	4	4.847	2.006	0.089	0.131	2.621	18 (2+4+12)	37, 39, 43, 31	(iv)
glc	1	7.513					0	41	(v)
	2	5.729	2.512	3.217			10	43, 33	(vi)
	3	5.180	1.782	0.181	3.217		16 (6+10)	37, 43, 33	(vii)
	4	5.170	0.086	0.086	1.780	3.217	12 (2+6+4)	41, 43, 37, 33	$(viii)$
lac	1	7.344					0	41	(ix)
	2	5.929	0.706	5.222			6	41, 35	(x)
	3	5.885	0.433	0.230	5.222		8 (2+6)	39, 41, 35	(xi)
	4	5.876	0.433	0.230	3.044	2.169	12 (2+8+2)	39, 41, 33, 35	(xii)

Table 4.4: Times of optimal solutions with $1, 2, 3$ and 4 consecutive flux modes. Calculated with the additional constraint that genes can only be turned on, but not deactivated from one interval to the next. Solutions (iv),$(viii)$ and (xii) are visualised in the network in Figs. 4.9 to 4.11.

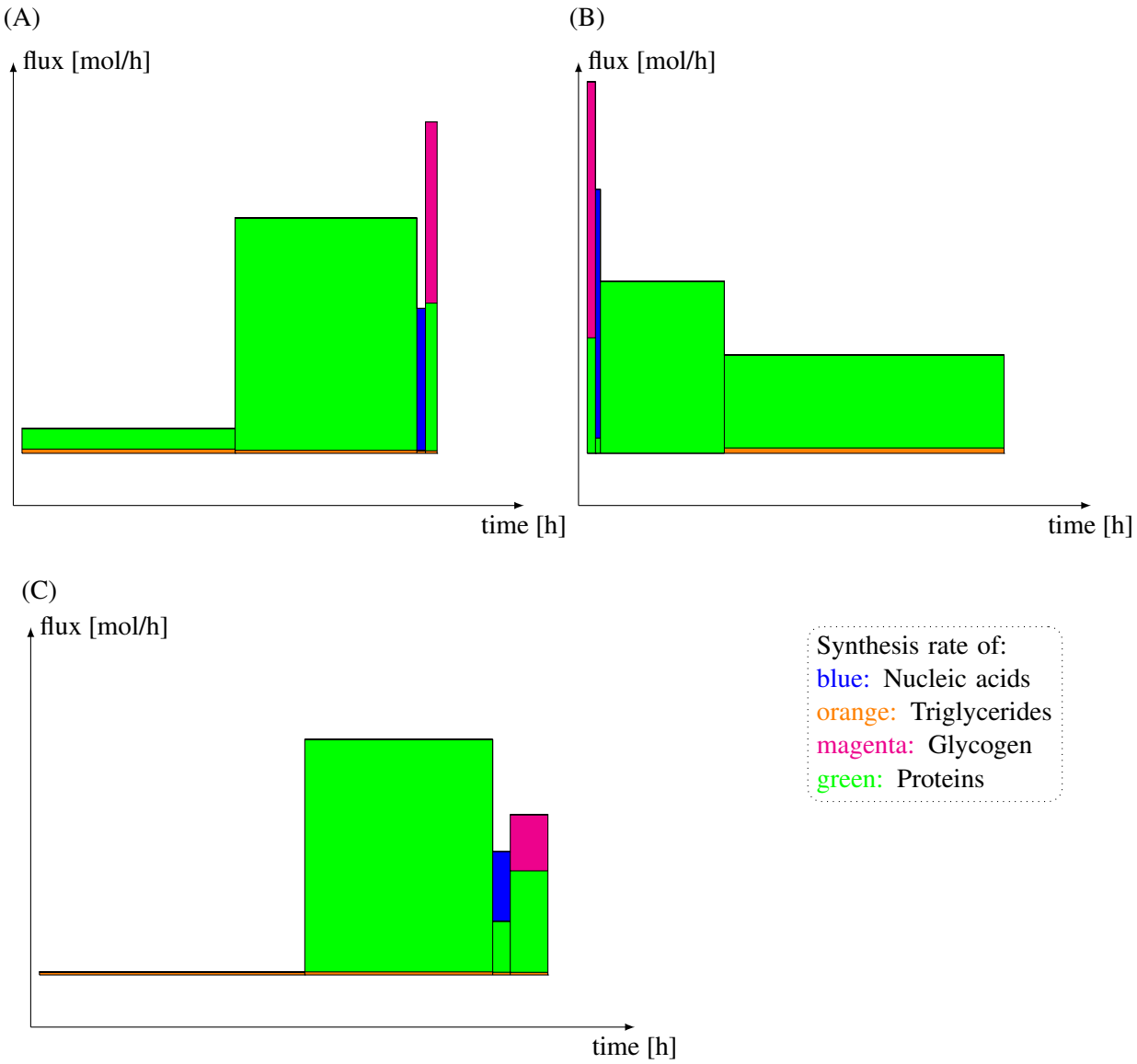


Figure 4.5: Flux rates through the biomass producing target reactions within various phases in the solution of the optimisation problem with $l = 4$ different phases, visualising the solutions (iv), $(viii)$ and (xii) from Tab. 4.3. (A) Glucose and lactate are available substrates, (B) Glucose is the only substrate, (C) Lactate is the only substrate. The size of the coloured areas corresponds to the amount of biomass components produced in the respective time interval. Due to the high demand for protein synthesis and the high cost of fatty acid synthesis, the solutions are dominated by these requirements. Fatty acid synthesis is limited to relatively small rates and must therefore be active most of the time, i.e., permanently if lactate is available (A and C) and 67% of the time if only glucose is available (B).

4.4.2 Estimating the cost for switching

On the one hand, the above results show that switching between different phases of the metabolic network allows for a faster production of all required output metabolites. On the other hand,

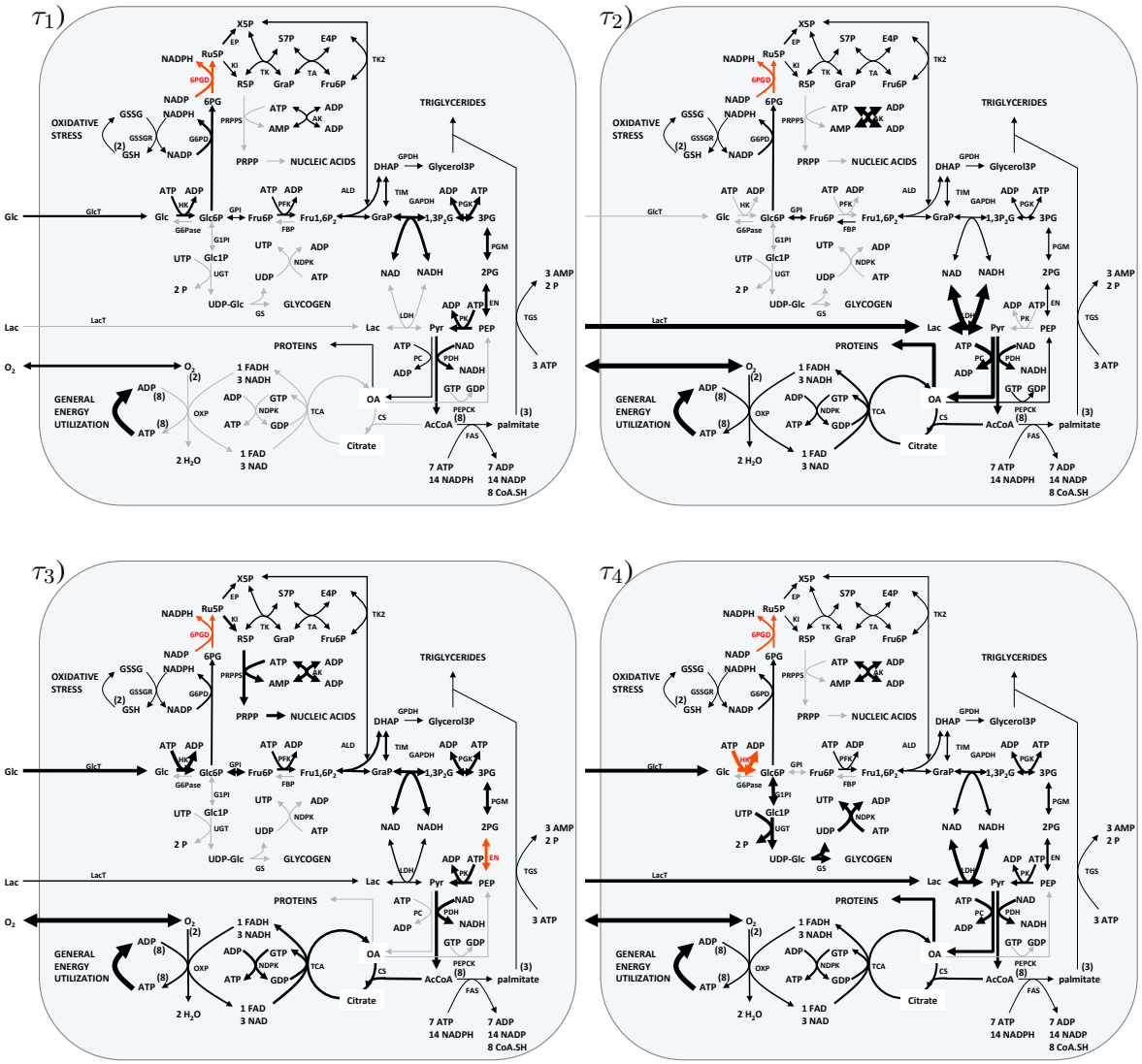


Figure 4.6: Steady state flux distributions of the four different phases of biomass production. The shown flux modes correspond to the optimal solution (iv) in Table 4.3. Active reactions are indicated by black arrows, where the thickness indicates the flux rate. For the identification of rate-limiting reactions, those fluxes which coincide with their upper bound are marked red.

the synthesis and degradation of proteins are consuming ATP. This additional energy demand is not considered in the present method. However, the computations of Fig. 4.4 showed that the capacity for output production is only reduced when the ATP demand is increased 2.5 fold at least. Although these computations were done for the reference case, it is likely that the optimal solutions from Tabs. 4.4 and 4.4 are not sensitive to the additional ATP demand neither. To verify this presumption, we added the additional ATP demand for switching between the different phases a posteriori to the maintenance constraint $v_{\text{ATP}} \geq mb_{\text{ATP}}$.

Derivation of switching cost We estimate the cost of enzyme synthesis, but the energy that is used for degradation of enzymes is not explicitly considered in our calculations. However, since the whole available protein pool A_{tot} is utilised in every phase, degradation and synthesis of enzymes sums up to zero for every switch and the degradation is hence reflected in the synthesis of enzymes. To determine the amount of ATP that is consumed for the synthesis, we used an estimated cost of 5 ATP per amino acid. For one protein of enzyme j we then have a cost of $5 \cdot (\gamma_j$ [Da] $/126$ [Da]) molecules of ATP, where 126 [Da] is the average molecular weight of a single amino acid. Given an optimal solution consisting of l phases, let $E_j^k := A_{tot} \frac{1}{\gamma_A + \langle g, \gamma \rangle}$ be the enzyme concentration for reaction j in the k-th phase. The corresponding bounds are then given by $ub_j^k = E_j^k kc_j^+$, $lb_j^k = -E_j^k kc_j^-$ and the change in enzyme concentration for reaction j in the k-th switch, $k = 1, \ldots, l-1$, is $E_j^{k+1} - E_j^k$. We obtain these enzyme amounts as $E_j^k = ub_j^k / kc_j^+$ for $j = 1, \ldots, n$, $k = 1, \ldots, l$ and the total amount of enzymes, which are newly synthesised in the transition from phase k to $k+1$ is then given by

$$\sum_{j=1}^{n} \max(\Delta^k ub_j / kc_j^+, 0), \quad k = 1, \ldots, l-1$$

where the $\Delta^k ub := ub^{k+1} - ub^k$ and the maximum is taken to neglect the negative values that correspond to degradation of enzyme proteins. Altogether we have the following calculation for the cost of switching from phase k to phase $k+1$ denoted C^k for $k = 1, \ldots, l-1$:

$$C^k := 5 \cdot (\gamma_j / 126) \cdot \sum_{j=1}^{n} \max(\Delta^k ub_j / kc_j^+, 0) \tag{4.5}$$

The value of C^k depends on the gene expression states g^k, g^{k+1} in the two phases, because they determine the values of the upper bounds. Hence, (4.5) gives a cost function for the enzyme synthesis that depends on the switch in gene expression given by the pair g^k, g^{k+1}.

Impact on the required production time. The additional ATP demand was integrated into the model by increasing the permanent maintenance constraint by an amount ϱ [mol/h]. The new maintenance bound is then $mb'_{ATP} := mb_{ATP} + \varrho$ and ϱ is chosen such that during the total duration $\tau = \sum_{k=1}^{l} \tau_k$ the total switching cost is accumulated, i.e., $\tau\varrho = \sum_{k=1}^{l-1} C^k$ or, equivalently, $\varrho = \tau^{-1} \cdot \sum_{k=1}^{l-1} C^k$. For the optimal solutions that were obtained for the proliferating and non-proliferating case (Tabs. 4.4 and 4.5, respectively) the increase of the maintenance bound was always below 10% and the optimal solutions remained virtually unchanged (production time increased by less than 1%). Compared with the gain in time that is achieved by switching, we can conclude that the additional cost for switching is marginal and does not impair the benefit of using different flux modes. It is hence justified to neglect these costs in the computational procedure for this metabolic network model.

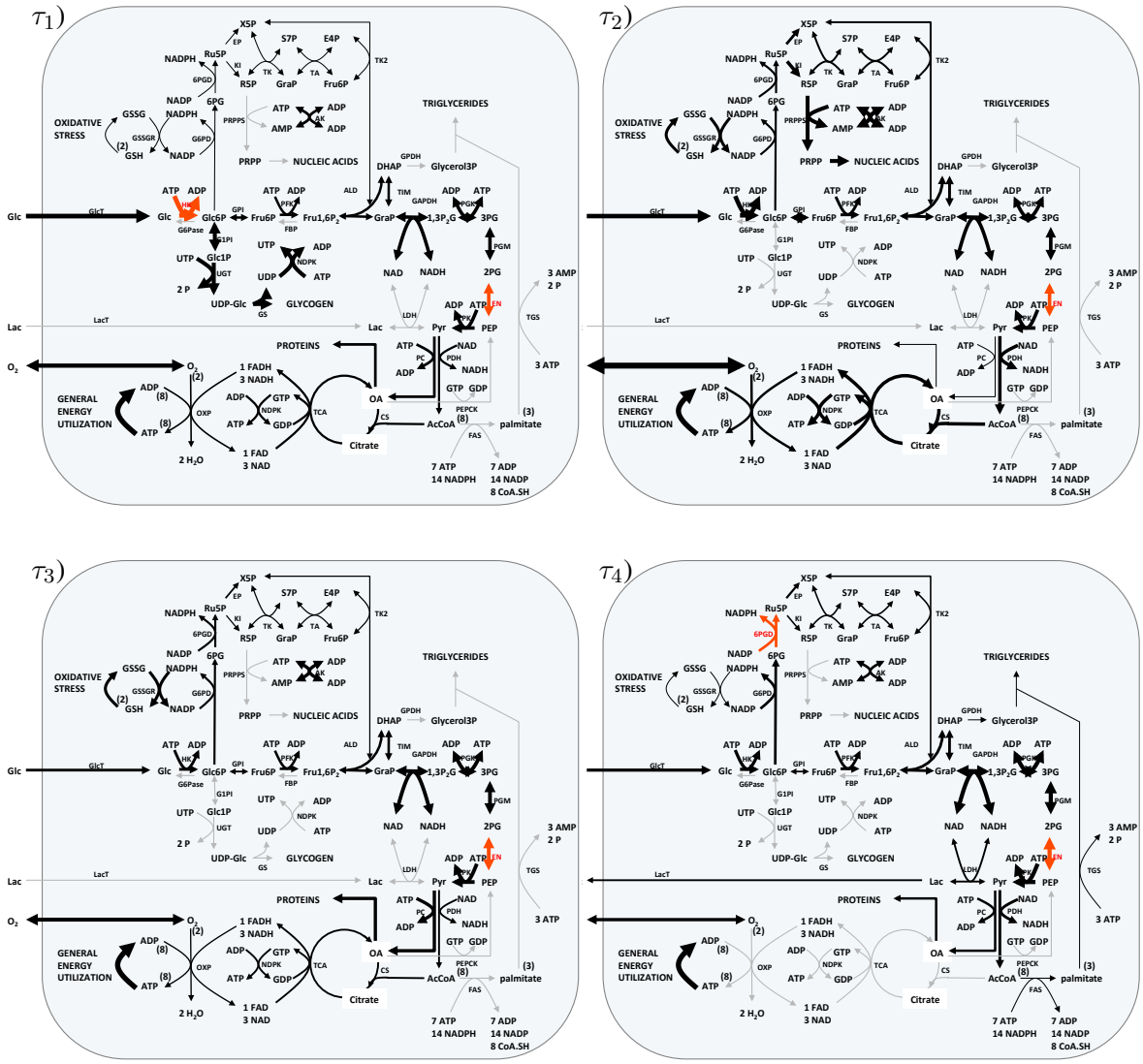


Figure 4.7: Glucose as substrate. The shown flux modes correspond to solution $(viii)$ in Table 4.3, see also Fig. 4.5. Note that lactate is secreted in the last phase. This is likely to be caused by a need of NAD which is provided by LDH during lactate production.

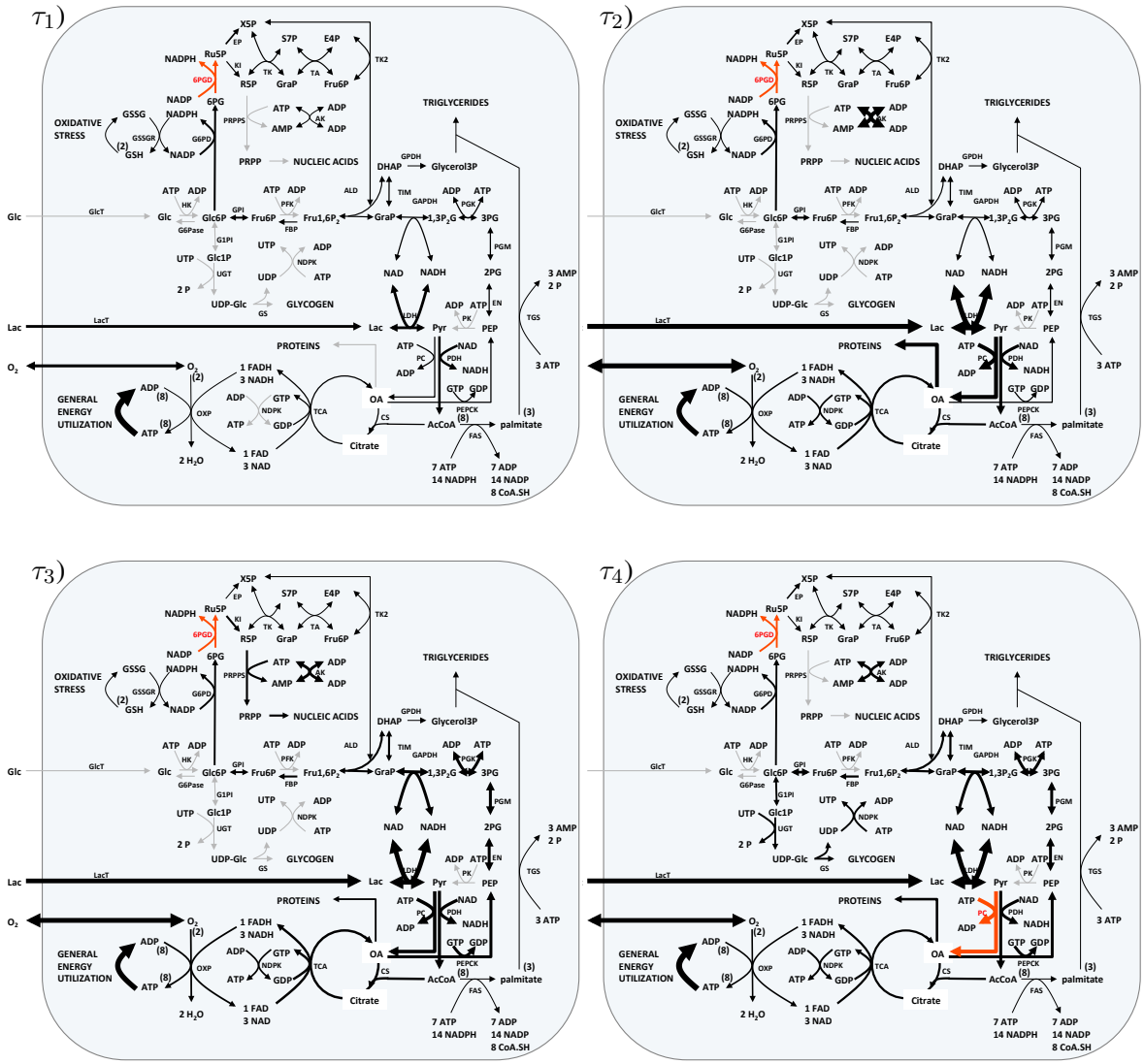


Figure 4.8: Lactate as substrate. The shown flux modes correspond to solution (xii) in Table 4.3, see also Fig. 4.5.

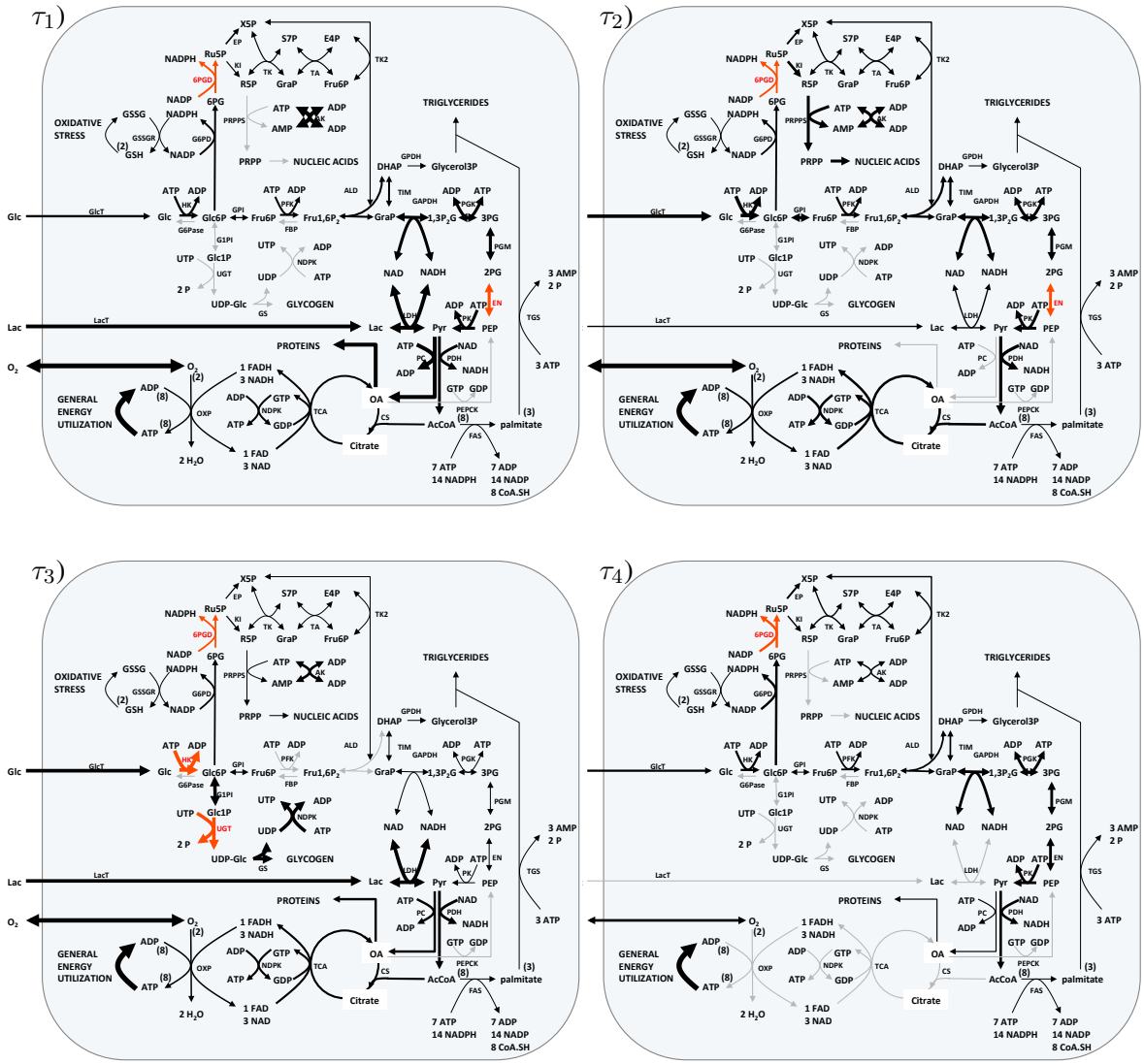


Figure 4.9: Glucose and lactate as substrates. The shown flux modes correspond to solution (iv) in Table 4.4.

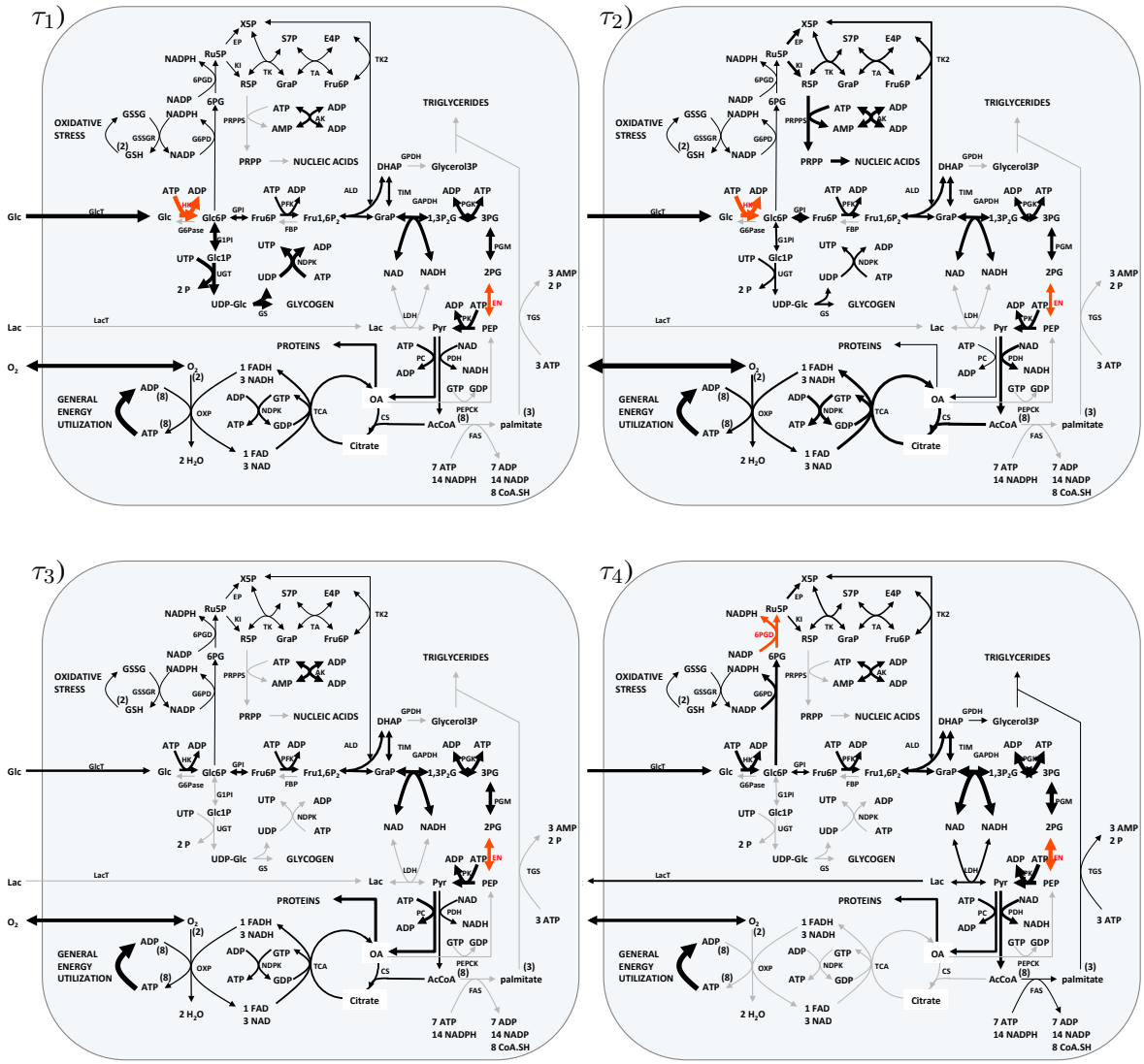


Figure 4.10: Glucose as substrate. The shown flux modes correspond to solution $(viii)$ in Table 4.4.

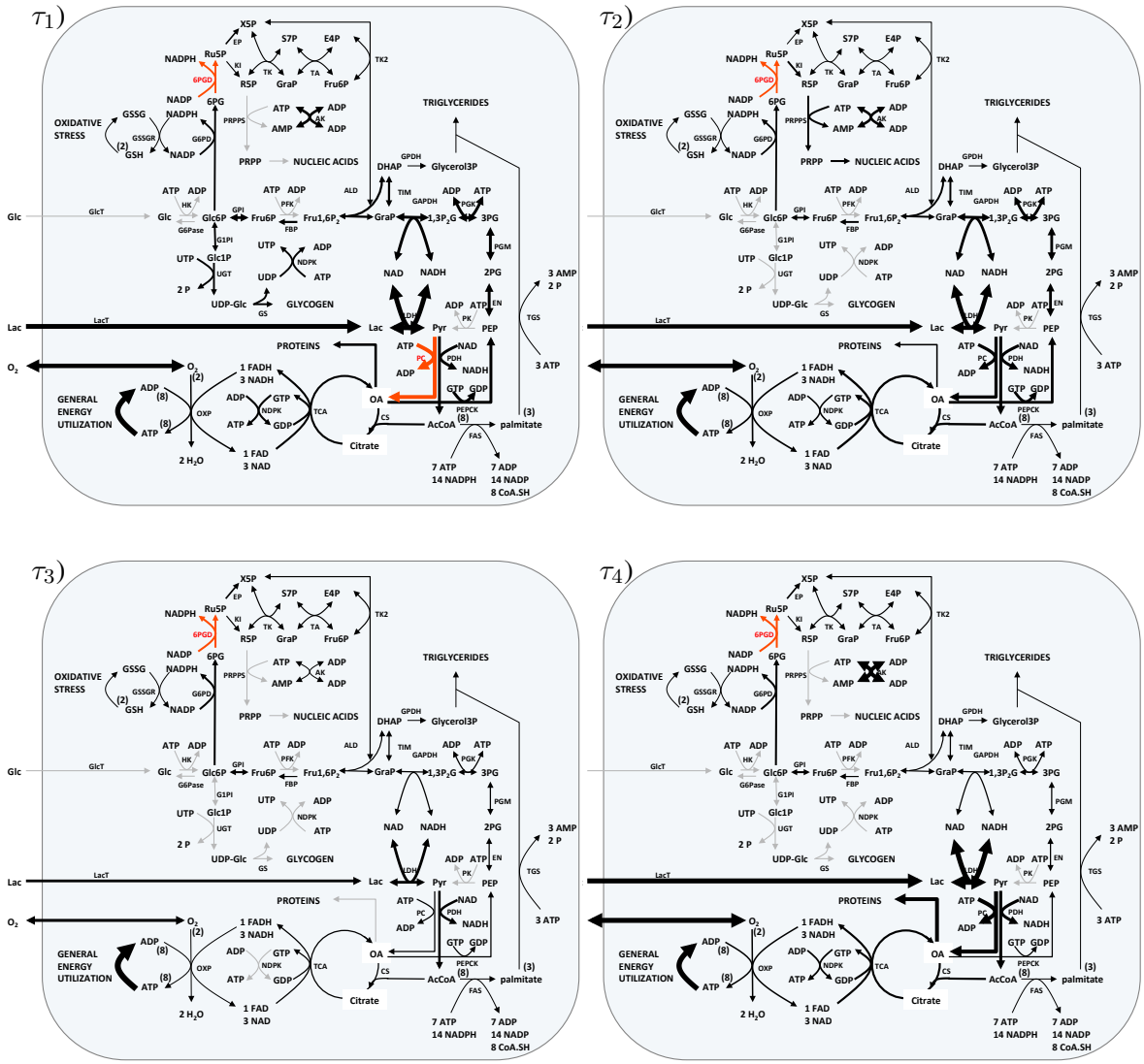


Figure 4.11: Lactate as substrate. The shown flux modes correspond to solution (xii) in Table 4.4.

4.4.3 Robustness of optimal solutions against random variations of model parameters

The computations presented above were carried out under the assumption that the expression efficiencies of all enzymes are equal ($\eta_j = 1$ for all j). However, expression efficiencies may considerably vary as convincingly demonstrated by the generally poor correlation between mRNA and protein levels (see e.g. [Maier et al., 2009]). Therefore, in order to exclude that the results obtained in the preceding sections critically depend on the choice of the expression efficiencies, we randomly varied their numerical values from a lognormal distribution (see p. 107), i.e., $\eta_j \sim ln\mathcal{N}(0, 0.36^2)$. As in the preceding sections, we determined for each sample the minimal value of A_{tot} with which the demanded metabolic output can be accomplished within 8 [h] in the reference case (see Fig. 4.12). As expected, variations of the expression efficiencies had a

large impact on A_{tot} with deviations of up to a factor of 5 from the unperturbed value A_{tot} = 0.087 [mg/gDW].

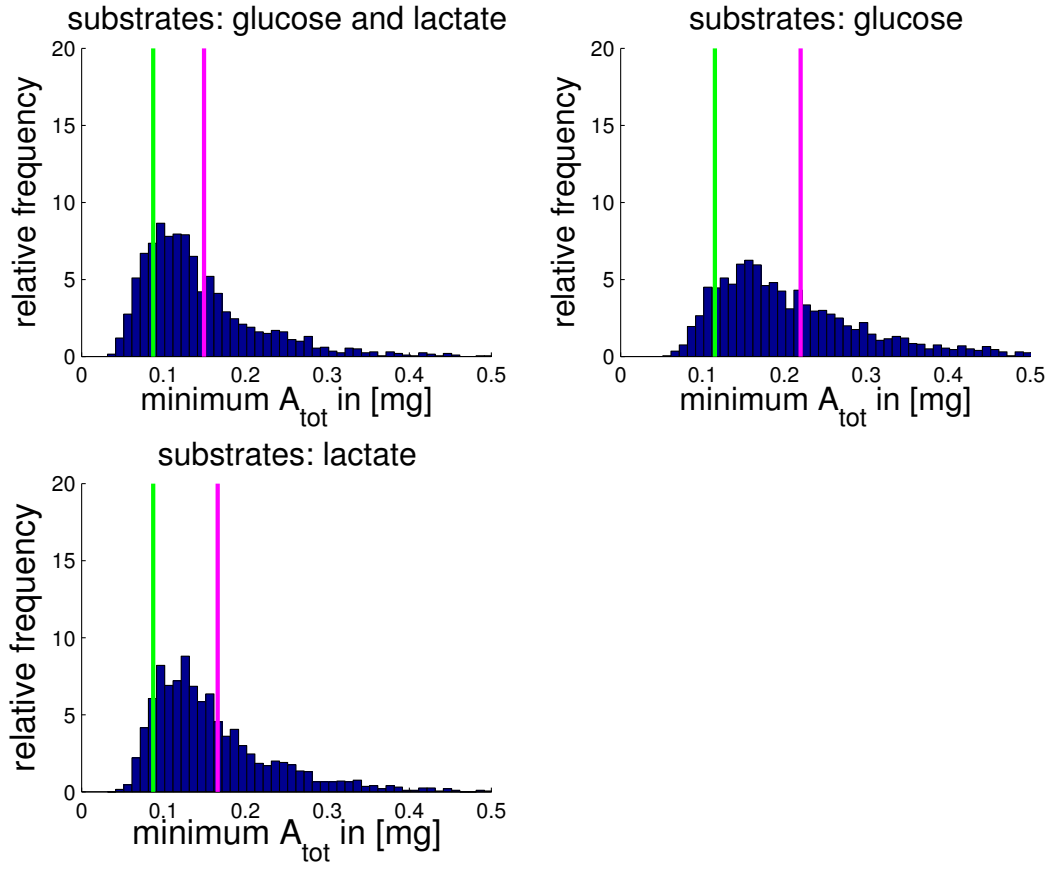


Figure 4.12: Impact of variations in the enzyme expression efficiencies on the minimal total amino acid mass A_{tot} required to produce the metabolic output within 8 [h] in the reference case (all genes active, no switch). The frequency distributions are the outcome of 2000 computations of A_{tot} where the values of the expression efficiencies were sampled from the lognormal distribution $\eta_j \sim ln\mathcal{N}(0, 0.36^2)$. The unperturbed values, i.e., $\eta_j = 1$ for all reactions j, are marked in green. The mean of the samples is indicated in magenta. Intriguingly, the mean of the samples is significantly above the value of A_{tot} in the unperturbed model. This can be explained by the nature of perturbed flux spaces and the corresponding optimal solutions, which will be examined in Chap. 5.

Gain in production time For each sample of the parameters and the corresponding protein amount A_{tot} enabling output production within 8 [h], we proceeded by solving $(\mathscr{P}_0)$ as well as $(\mathscr{P}_1)$ for $l = 1, 2, 3, 4$ different phases and different combinations of glucose and lactate as available substrates. Fig. 4.13 shows the range of obtained times $\tau(l)$, $l = 1, 2, 3, 4$ using both substrates.

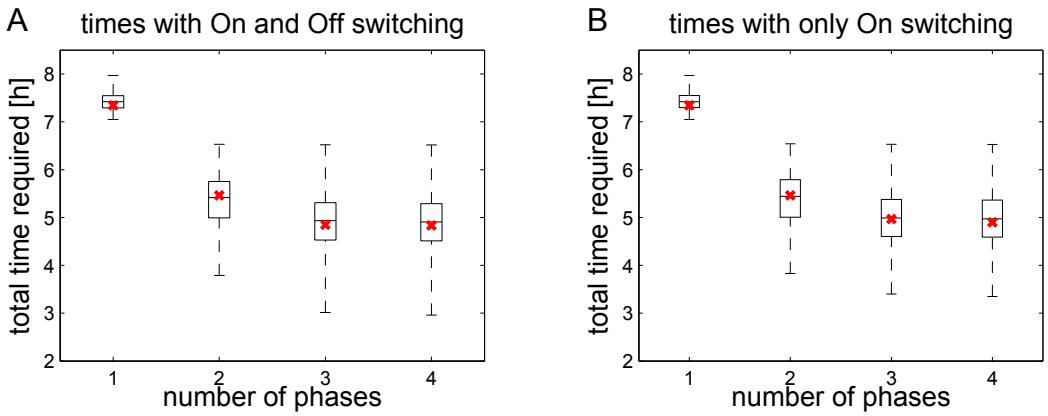


Figure 4.13: Possible ranges of the gain obtained with different numbers of metabolic phases and glucose and lactate as allowed substrates. (A) Genes can be switched on and off, (B) Genes can only be switched on. The red mark represents the reference case ($\eta_j = 1$ for $j = 1, \ldots, n$)

Remarkably, for all random samples the strategy of alternating gene switches resulted in a reduction of the total biomass-production-time. In the other substrate conditions, only glucose or only lactate, we get virtually the same outcome as shown in Fig. 4.13 for both substrates. Altogether, the results coherently validate that a gain in production time is achieved independently of the expression efficiencies. Thus, the findings of the preceding section, according to which a variable use of selected gene sets in different metabolic phases allows a reduction of the biomass production time, turn out to be robust against variation of enzyme expression efficiencies. Also the observation that the largest gain is achieved by allowing two different phases and the extension to three and four phases only brings minor gains, is reconfirmed.

4.5 Discussion

The general approach that was developed in Chap. 3 was applied here with a binary gene expression space $G \subset \{0,1\}^n$ to a minimal illustrative example and to a core metabolic network. The analytical examination of the minimal example as well as the computational results for the core network showed that the efficiency of cells to accomplish their metabolic output can be increased by variable usage of genes and related metabolic enzymes. The main conclusion is that switching between different metabolic phases could be a regulatory mechanism that allows for faster production of metabolic output. Temporal gene expression is mainly known and modelled as a means to adapt the organism to changes in the environment. In contrast, the results here are obtained from a model with constant environment and thus they suggest that temporal gene expression could also have emerged as a strategy to produce metabolic output fast and efficiently. The minimisation of production time is in particular advantageous for rapidly dividing cells. These are separately taken into account by $(\mathscr{P}_1)$, where complete degradation of enzymes is disallowed.

4.5.1 Biological assumptions and model building

As follows from Prop. 3.11 and Prop. 3.7, a gain in production time can only be achieved by switching if the gene expression cannot be perfectly adjusted to a flux distribution that is optimal for the metabolic requirements. Such a precise adaptation of the gene expression of metabolic enzymes to flux requirements seems unlikely for several reasons. First, this assumption would entail that all enzymes are saturated, i.e., operate at their maximal capacity (v_{max}, cf. p. 40). In contrast, all metabolic pathways studied so far have in common that only few enzymes (mostly only a single one) are rate limiting, while the others are not saturated with their ligands, and therefore possess a large overcapacity compared to the actual flux they carry. Second, an investigation of the control of metabolic flux in the model bacterium *Bacillus subtilis* by quantifying fluxes, transcripts and metabolites in eight metabolic states enforced by different environmental conditions revealed that for the majority of enzymes in central metabolism, enzyme concentrations were insufficient to explain the observed fluxes [Chubukov et al., 2013]. In line with this finding, a suboptimal control of gene expression was reported to be widespread in bacteria [Price et al., 2013]. Third, from an evolutionary point of view, a perfect economical allocation of proteins to the enzymes and transporters of the cellular reaction networks is unlikely to occur, as biological systems that perform multiple tasks face the fundamental trade-off that a given phenotype cannot be optimal at all tasks [Shoval et al., 2012].

The strategy of using different phases for the production of the different output metabolites allows producing more or less of certain outputs simply by shifting the time points of switching from one phase to the subsequent. Formally, every combination of outputs, given by $\Gamma \in \mathbb{R}^{|T|}_{\geq 0}$ can be achieved by adjusting the time points of switching between $v^1, \ldots, v^l$, as long as $\Gamma \in \mathrm{cone}(v^1_T, \ldots, v^l_T)$ (where v^k_T denotes the restriction to the target reactions $t \in T$). On the other hand, if we assume that the flux rates are precisely controlled by gene expression, the adaptation of the flux distribution would require exact readjustment of most of the gene expression values. However, this is highly unrealistic, since it involves many reactions and associated metabolic genes (some dozens in our core metabolic network) and only if the expression of all metabolic genes is correctly changed, the new required flux distribution can be obtained. In fact, the flexibility to change the proteome is limited by the structure of the regulatory network of the cell [Chubukov et al., 2012]. In contrast, readjustment of the durations of the phases in the here conjectured strategy can be achieved by shifting the time point of few signals only (four in our core network). These considerations show that the alteration between different metabolic phases is theoretically an efficient and also flexible manner of producing metabolic output. The results of this chapter are thus proposing a viable strategy for coordinated output production and are giving a theoretical proof of concept by estimating a significant gain in efficiency that is achieved by this strategy.

4.5.2 Possible explanation of metabolic cycling

In a non-dividing ("resting") cell, the metabolic output does not lead to an accumulation of the total biomass but instead serves to replenish the permanent loss of biomass components by various degradation and damage processes (lipid oxidation, proteolysis, glycogen consumption, RNA degradation). Based on the observations here, we can hypothesise that the phases that achieve an efficient output production are repeated periodically to assure permanent restoration of the biomass composition of the cell. As explained in the preceding paragraph this production strategy can always adapt to changes in the demand for the different biomass components by ad-

justing the durations of the individual phases. Formally, let $\varepsilon < 1$ denote the fraction of biomass components that is lost in a given time and that needs to be restored within this time period (= homeostasis) to prevent severe cell damage or loss of cell functions. Then the results here predict that switching between different metabolic phases permits a faster reproduction of the demand $\varepsilon\,\Gamma$ than achievable with a constant metabolic steady state. If this scenario repeats periodically, the outcome is a so-called metabolic cycling where each cycle produces only a certain fraction $\varepsilon\,\Gamma$ such that, without permanent loss, after $1/\varepsilon$ cycles the complete biomass would have doubled. From this consideration one may conclude that the economic usage of the protein pool that can be allocated to metabolic enzymes is one possible explanation for the periodic switches between distinct metabolic states observed in various non-proliferating cell types as, for example, yeast cells in batch cultures [Silverman et al., 2010, Slavov et al., 2011]. In mammals, such oscillators have been found in almost all peripheral cell types [Nagoshi et al., 2005, Schibler and Naef, 2005]. Ex vivo experiments have clearly demonstrated that such cell-intrinsic oscillators may work independently from circadian changes of the cell's environment (e.g. variations in the concentration of nutrients or hormones) and phases of the cell cycle [Slavov et al., 2011].

4.5.3 Computational approach

The computational tools that were applied in this chapter are all based on the constraint-based modelling approach. The whole computational procedure to obtain the results consists of many separated steps, all of which include optimisation problems. The steps are executed in the following order:

- Solving the reference case to determine the total protein amount A_{tot} as well as A_{tot}^{crit}.
- Computing the required production time for varied maintenance bounds in order to analyse the sensitivity.
- Generating the MFMs as optimal flux modes for the production of individual target metabolites. The corresponding MGSs are used to defined the constraints on $G \subset \{0,1\}^n$.
- Solving the optimisation problems $(\mathscr{P}_0)$ and $(\mathscr{P}_1)$ for the production of demanded output with $l = 1, 2, 3$ or 4 different phases.
- Repeating the previous steps (step 2 is not necessary) for sampled enzyme efficiencies to check for robustness.

The introduction of maintenance constraints is strongly increasing the computational complexity of all these problems. However, for this core metabolic network all optimisation problems are still tractable. In principle, the computational procedure from this chapter can also be applied to any other metabolic network that is given with the required biological information about demanded metabolic output and enzyme parameters. To make the approach tractable for larger networks, alternative computational methods will be presented in Chap. 6.

4.5.4 Outlook

The method that was presented in Chap. 3 and applied in this chapter was motivated by the hypothesis that optimal resource allocation is not static, but switches between different pathways. Hence it offers an explanation for alterations in metabolism which are not triggered by external

signals and occur in a constant environment. The mathematical analysis showed that the explanation of the alterations in the model relies on the assumption of binary gene expression, or at least on a non-convex gene expression space. This means that the enzyme concentrations cannot be adjusted individually to the exact amount necessary for a given metabolic steady state. Several arguments were given to justify this assumption biologically. Also the flexibility of the metabolic network and the underlying regulatory network to readjust to changes in demand or to other conditions was discussed as an advantage of the strategy of sequential phases. This leads directly to the question, which design principles can explain the functionality of metabolic networks. The present work argues for a design principle where the metabolic network only uses a limited number of metabolic steady states, for which well suiting enzyme concentrations can be established by gene regulation. These steady state flux modes are not perfectly suited for the typical metabolic requirements, but switching between them and adjusting the duration of each flux mode allows controlling the fulfilment of the individual biological functions, in particular biomass production. This aspect was not formally discussed here, but a biological discussion and mathematical analysis would be an interesting direction to further explore the interplay of metabolism and gene regulation.

5 Effects of perturbations of bounding constraints

5.1 Overview

In Sec. 4.4.3 of the previous chapter, the robustness of the results was analysed w.r.t. the unknown enzyme efficiencies η_j, $j = 1, \ldots, n$. The resource allocation model was given by (3.7), i.e., $v_j \leq ub_j := g_j k c_j^+ A_{tot} \frac{\eta_j}{\sum_i g_i \gamma_i \eta_i}$, where the enzyme efficiencies were all set to $\eta_j = 1$, $j = 1, \ldots, n$. To examine robustness, the η_j were then sampled from a lognormal distribution. This can be seen as perturbing the original bounds lb, ub where $\eta_j = 1$, $j = 1, \ldots, n$. The results in Fig. 4.12 contain the surprising detail that in average the perturbed models require a significantly larger amount of amino acids to achieve the output production in the reference time of 8 [h]. In this chapter we will give strong evidence that this phenomenon is caused by effects of the constraint-based representation of the flux space. On the one hand, this evidence is based on the observation that other possible explanations can be ruled out. On the other hand, we can observe and rigorously explain the same effect for a simpler case of a linear flux space, where perturbations of the bounding constraints lead to deteriorated objective values in optimisation. Another perspective on the perturbations, which might help to understand the origin of these effects, can be obtained by iteratively perturbing the bounds and observing the asymptotic behaviour as the number of iterations goes to infinity. The results are proven formally as qualitative statements for perturbations that are distributed with an unbiased expectation as e.g. the normal distribution. To also examine the case of lognormally distributed perturbations, where the expectation is biased, as well as to quantify the effects, the last section examines them empirically. This empirical analysis confirms the significance of the effects, also for lognormally distributed perturbations. Furthermore it helps to understand which structural properties of the network representation are determining the effects of perturbations. These observations elucidate how the representation of the model, in particular the constraints that bound the flux rates, gives rise to these unexpected side effects of perturbations.

5.2 Properties and variants of perturbations

Perturbations of parameters in the model description, as the bounds in this case, can be modelled by random variables. The natural choice is the normal or lognormal distribution. Their disadvantage are the long tails that represent biologically infeasible perturbations. Cutting off these unrealistic parts of the distribution might be a solution, but it also changes the expectation value.

5.2.1 Notation and setting

An introduction to random variables and related notions is given in the Mathematical Preliminaries, p. 16. The formal treatment in this chapter will be much simplified by the assumption that the metabolic network contains only irreversible reactions, i.e., all reversible reactions were split into two irreversible reactions. This formal modification has no impact on the calculations, all results hold equally for networks with reversible reactions. The only difference would be that formulations and calculations would have to distinguish between the positive and the negative bounds.

With this convention the bounds on flux modes in the network are simply given by $0 \le v \le ub$ and only the bounds ub are perturbed. Since we exclude blocked reactions from the network analysis, $ub_j > 0$ for all $j = 1, \ldots, n$. The perturbation of the bound ub_j is described by a random variable $B_j \colon \Omega \to \mathbb{R}$, where Ω is the sample space (see p. 16 for details). The values of B_j are called the samples and denoted $\widetilde{ub}_j := B_j(\omega)$, $\omega \in \Omega$. The whole vector of perturbed bounds is denoted $\widetilde{ub}$ and

$$\widetilde{\mathcal{F}} := \{v \in \mathbb{R}^n \ : \ Sv = 0, \ 0 \le v \le \widetilde{ub}\}$$

is the resulting flux space. All B_j are considered to be independent random variables. This might not be realistic if two random variables B_j, $B_{j'}$ refer to two directions of one reversible reaction which was split up and both directions are catalysed by the same enzyme. But to facilitate the formal argumentation, this aspect is not always taken into account in the calculations. The results do not depend on the assumption that *all* perturbations are independent, although the interplay of *several* independent perturbations is essential.

5.2.2 The distribution of the perturbations

Usually, perturbations are of interest because the parameters that lead to the bounds ub are not known exactly. In particular, they typically fluctuate from cell to cell. Therefore, it can be important to validate results that are obtained by a model with distinct bounds $ub \in \mathbb{R}^n$. The given bounds ub are just candidates for the real values that limit the metabolic flux rates in the cell. By applying perturbations to these bounds we can explore the metabolic network with comparable candidates $\widetilde{ub}$ which might be as realistic as the original bounds ub. Which distribution of the perturbations is appropriate in the mathematical model, depends on the distribution of the error due to uncertainty or the distribution of the fluctuations. If the distribution is not known, there might still be some qualitative information available that is helpful in making a choice. As an example consider the case that the fluctuations in enzyme amounts are supposed to be modelled. The underlying kinetics of synthesis and degradation of the enzymes has to be considered then. Stochastic fluctuations in reaction kinetics are described by the chemical master equation (see Sec. 8.2). In the simplest case, where the concentration just depends on synthesis and degradation, the molecule amount is Poisson distributed around the equilibrium (see p. 167). A good approximation of the Poisson distribution is given by the normal distribution [DasGupta, 2010, Thm. 10.5, p. 229]. For modelling fluctuations in the enzyme concentrations by perturbing the bounds, the normal distribution is hence well qualified. To take into account that the decrease of a reaction bound can at most lead to a bound at zero, the lognormal distribution is a natural candidate to be used instead of the normal distribution. In this case the perturbation is given by multiplication with a lognormally distributed factor. In this chapter we will restrict the analysis of explicit distributions of perturbations to these two. Their most important properties

are shortly introduced below. The analytical observations are stated for general perturbations which are not biased for an increasing or decreasing perturbation. This means that the expectation is the original value, i.e. $E(B_j) = ub_j$. The normal distribution fulfils this requirement, but the lognormal does not. Only the empirical analysis will reveal that the formally discussed effects occur in the case of lognormal distributions as well. The median is an alternative measure of bias and by this measure the normal as well as the lognormal distribution are unbiased, that is, the median of B_j is ub_j in both cases.

Normal distribution For ub_j the normally distributed perturbation is given by $B_j \sim \mathcal{N}(\mu, \sigma^2)$. This notation means that B_j is a **normally distributed** random variable with mean μ and variance σ^2, i.e., B_j has the probability density function (PDF)

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

For B_j the mean is $\mu = ub_j$, equivalently we can define $B_j := X_j + ub_j$, where $X_j \sim \mathcal{N}(0, \sigma^2)$, because then $B_j \sim \mathcal{N}(ub_j, \sigma^2)$ by definition. The cumulative distribution function (CDF) of a normally distributed random variable is given by

$$F(x) = \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{-(x-\mu)^2}{\sqrt{2\sigma^2}}\right)\right),$$

where erf denotes the **error function** $\operatorname{erf}\colon \mathbb{R} \to (0,1)$ given by $\operatorname{erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)dt$ [Fristedt and Gray, 1997]. The range of $B_j \sim \mathcal{N}(\mu, \sigma^2)$, that is, the support of the PDF is $\mathbb{R}$. The expectation is given by the parameter μ, and coincides with the median.

Lognormal distribution A lognormally distributed perturbation is defined by multiplying with a sampled factor, i.e., $B_j := Y_j\, ub_j$, where Y_j is a **lognormally distributed** random variable, $Y_j \sim log\mathcal{N}(\mu, \sigma^2)$. The lognormal distribution can be defined via the normal distribution, i.e.,

$$Y_j = \exp(X_j) \text{ with } X_j \sim \mathcal{N}(\mu, \sigma_j^2).$$

The range of Y_j or, equivalently, the support of the PDF is hence $\mathbb{R}_{>0}$. Here we will use lognormal distributions with $\mu = 0$, as a consequence the median is $\exp(\mu) = \exp(0) = 1$. The expectation is $\exp(\mu + \sigma^2/2)$, which is larger than the median.

Cutting off the distributions The support of the PDF of the normal distribution is $\mathbb{R}$, which means that the samples x of $X \sim \mathcal{N}(\mu, (\sigma^2))$ can be arbitrarily small or large. As a consequence, the perturbed bound can be zero or even switch its sign. In the model scenario this has no biological meaning. Therefore it is reasonable to cut off these perturbations, i.e., define $B_j := \max((ub_j + X_j), 0)$. As a consequence, we would have $E(B_j) > ub_j$, while the median stays ub_j. With the lognormal distribution we are not confronted with this problem, since the range of $Y_j \sim log\mathcal{N}(\mu, \sigma^2)$ is $\mathbb{R}_{>0}$ and thus this is also the range of $B_j := Y_j\, ub_j$, because $ub_j > 0$. Another important difference between the normal and lognormal distributions is the relation between the probabilities of increasing and decreasing perturbations. In the lognormal distribution, $\widetilde{ub}_j \geq \lambda ub_j$ and $\widetilde{ub}_j \leq \lambda^{-1} ub_j$, $\lambda > 0$, have the same probability. With the normal distribution the probability for $\widetilde{ub}_j \leq \lambda^{-1} ub_j$ is strictly larger. Furthermore, in the case of normal

distribution, the deviations are relatively larger for small bounds, while the lognormal perturbations are obtained by multiplying ub_j with the sample and hence the dispersion is weighted by the magnitude of the bound ub_j. The normal distribution with positive cut off and the perturbation by a lognormal distribution have in common that $E(B_j) > ub_j$, while the median is exactly ub_j. For the lognormal perturbations, a cut off from above seems reasonable, since the probability for large samples is higher than in the normal distribution. An increase by the factor $a > 1$ has probability

$$P(B_j \geq ub_j a) = P(Y_j \geq a) = P(\exp(X_j) \geq a) = P(X_j \geq log(a))$$

with the lognormal perturbation $B_j = ub_j Y_j = ub_j \exp(X_j)$. The same increase has probability $P(ub_j + X_j \geq ub_j a)$ with the normal perturbation $B_j = ub_j + X_j$. The larger a, the higher becomes the probability of a lognormal perturbation reaching a compared to the probability of a normal perturbation reaching a.

These different variants of distributions with cut off will not be considered in the analysis in this chapter, instead we will focus on the normal and lognormal distribution. The formal discussion will show that the observed effects do not depend on the exact form of the distribution but mainly on the expectation value and the support.

5.3 The core metabolic network revisited

> Apart from the results of perturbations of the enzyme efficiencies presented in the previous chapter, see Fig. 4.12, further computations with different distributions of the perturbations were performed. However, the results did not change qualitatively, thus suggesting that the effect of the shifted average is not caused by a certain feature of the distribution, but emerges from the mathematical structure of the flux space representation.

The perturbations that were applied to the core metabolic network in Sec. 4.4.3 did not directly perturb the bounds, but were a bit more intricate. The parameters $\eta_j = 1$ are perturbed by a lognormal distribution to $\tilde{\eta_j} \sim log\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.36$. The η_j occur as a factor of the turnover numbers, but also in the penalty term $\sum_{j=1}^{n} \gamma_j g_j \eta_j$ of the resource allocation. Perturbing η_j by Y_j is in fact equivalent to perturbing the turnover numbers $kc_j^{\pm}$ as well as the molecular weight γ_j by Y_j (cf. p. 45). Since the expectation of the lognormal distribution is above 1, also this penalty term is increased in expectation by these perturbations, i.e. $E(\sum_{j=1}^{n} \gamma_j g_j \tilde{\eta}_j) > \sum_{j=1}^{n} \gamma_j g_j \eta_j$. This is of course a possible explanation for the decrease in performance of the model as observed in Fig. 4.12. Therefore, the computations for the statistics of Fig. 4.12 were repeated without perturbing the penalty term, i.e., $\widetilde{ub}_i = g_i kc_i^{+} A_{tot} \frac{\tilde{\eta}_i}{\sum_j g_j \gamma_j \eta_j} = ub_j Y_j$. However, the results did not exhibit a significant difference to the results of Fig. 4.12. Also statistics with normally distributed perturbations with different parameters, with or without cut off, applied to η_j or only to $kc_j^{\pm}$, gave in all cases the qualitatively identical result of decreased performance, i.e., a significantly higher mean value of amino acid amount A_{tot} required to achieve output production in the reference time. This means that at constant A_{tot}, the mean value of required minimal production time over the perturbed models would have increased, see p. 87. We conclude that this phenomenon is most likely not caused by the distribution of the individual perturbations,

but seems to be a property of the resulting flux space and the optimisation that is performed over this flux space. Such properties will be uncovered in the succeeding section for linear flux spaces and LP optimisation. The close relation between the flux space of our resource allocation model and the polytope given by the turnover numbers alone (see Corollary 3.5) strongly suggests that the effect observed in linear flux spaces and in LPs is also occurring in optimisation problems of the kind (OP3), p. 58, and that this is what caused the shifted average in Fig. 4.12. However, a rigorous treatment of the non-linear case will not be given here.

5.4 Analytical results concerning the flux space and optimisation

5.4.1 Effects on the flux space

It turns out that even if the expectation value of each perturbation B_j of a single bound ub_j is again the original bound, i.e., $E(B_j) = ub_j$, the length of any ray segment in the flux space is getting smaller in expectation. This result is based on the observation that the minimum of several independent random variables with the same support is strictly smaller than the expectation of one of them, as illustrated in Fig. 5.1.

Setting In the introduction different distributions were discussed for modelling perturbations. Concluding, we can say that the normal and lognormal distribution are well suited, but it might be necessary or at least reasonable to restrict the range by a cut off. Some distributions of the perturbations that were mentioned have an expectation value that is in fact larger than the unperturbed bound, i.e., $E(B_j) > ub_j$. In this section, however, we require that the distributions satisfy $E(B_j) = ub_j$, $j = 1, \ldots, n$. This leads to the question whether the observations also hold in other cases, e.g. for lognormal perturbations, where $E(B_j) > ub_j$. This question will be answered affirmatively by the empirical analysis in the succeeding section. Another important condition in this section is that the PDFs of the perturbations B_j of the individual bounds all have the same support $(a, b) \subset \mathbb{R}$. The open interval $(a, b) \subset \mathbb{R}$ can be unbounded, i.e., $a = -\infty$ as well as $b = \infty$ is allowed. Furthermore, we assume that the PDF and the CDF of the perturbations B_j are continuous functions to avoid technical hindrances (this implies that the support of the PDF is an open set, in our case here an open interval by assumption). We are dealing in this chapter with the flux space

$$\mathcal{F} := \{v \in \mathbb{R}^n : Sv = 0,\ 0 \le v \le ub\} = \mathcal{C} \cap \prod_{j=1}^{n} [0, ub_j],$$

with the underlying flux cone $\mathcal{C} = \{v \in \mathbb{R}^n : Sv = 0,\ v \ge 0\}$. The perturbation only affects the bounds ub and thus not the flux cone. We assume $\infty > ub_j > 0$ for all $j = 1, \ldots, n$. However, we will discuss later the consequences of unbounded reactions, i,e., j with $ub_j = \infty$ on the effects of perturbations.

A fixed ray in the flux space In Chap. 3 we saw how a flux space can be completely described in terms of ray segments, see Lemma 3.3. We will now see that every ray segment of $\mathcal{F}$ becomes

shorter in expectation as the flux space is perturbed to $\widetilde{\mathcal{F}}$. A ray is fixed by a normalised vector from the flux cone, i.e., $s \in \mathcal{C}$ with $\|s\|_2 = \sqrt{\sum_{j=1}^{n} s^2} = 1$. The distance from 0 to the boundary of the flux space is given by the 'radius' $r(s) \geq 0$, this scalar is defined by $r(s) \cdot s \in \mathcal{F}$ and $\lambda\, s \notin \mathcal{F}$ for any $\lambda > r(s)$ (in other words $r(s) \cdot s \in M$, according to Def. 3.2). In fact, $r(s)$ depends only on the bounds, it can be given as

$$r(s) := \min_{i \in \mathrm{supp}(s)} \left(\frac{ub_i}{s_i} \right). \tag{5.1}$$

The minimum is taken over the support of s, see Fig. 5.1 A. Since $ub_j > 0$ and $s_j > 0$ for all $j \in \mathrm{supp}(s)$, we have $r(s) > 0$. For a perturbed flux space $\widetilde{\mathcal{F}}$ we denote the corresponding radius for a given normalised $s \in \mathcal{C}$ by $\tilde{r}(s) = \min_{i \in \mathrm{supp}(s)} \left(\widetilde{ub}_i \cdot s_i^{-1} \right)$. This minimum is taken over perturbation samples $\widetilde{ub}_j = B_j(\omega)$, it can be interpreted as the random variable

$$Z := \min_{i \in \mathrm{supp}(s)} \left(B_i s_i^{-1} \right). \tag{5.2}$$

Intuitively, it is compelling that the expectation of Z is smaller than the expectations of all $B_i s_i^{-1}$. It will be formally proven in Lemma 5.2 and leads to the following statement:

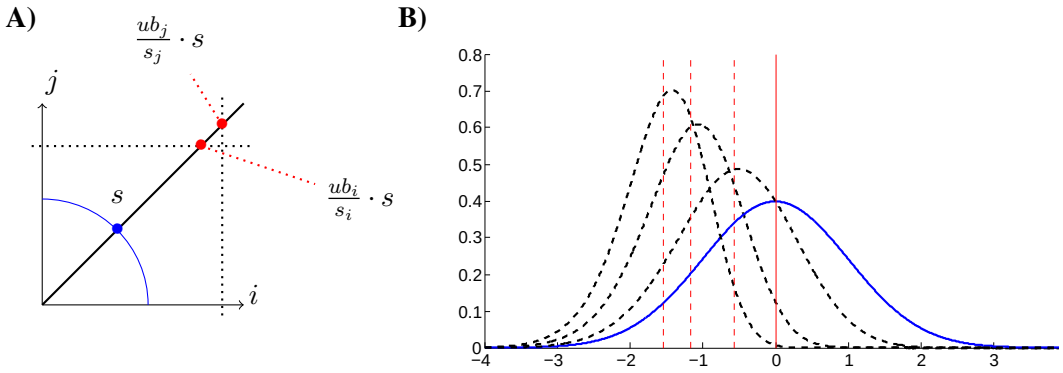


Figure 5.1: **A)** s is a normalised flux mode. The scaling of this flux modes is bounded by $r(s) := \min(\frac{ub_j}{s_j}, \frac{ub_i}{s_i})$. **B)** The PDF of the standard normal distribution is shown in blue and in dotted lines the PDF of the minimum of $2, 5, 10$ standard normally distributed random variables, respectively. Red vertical lines indicate the corresponding expectation values.

Proposition 5.1. If the bounds ub_j of the flux space $\mathcal{F}$ are perturbed according to independent random variables B_j, such that $E(B_j) = ub_j$ and the PDFs of the B_j all have the same support $(a, b) \subset \mathbb{R}$, then $E(Z) < r(s)$ for every s with more than one non-zero entries $s_i > 0$.

Before the proof is given, note that the condition that s has more than one non-zero entries is not a restriction for a metabolic network. If the cone $\mathcal{C}$ contained an s with a single non-zero s_j this would mean that reaction j is completely independent of the steady-state assumption and of the rest of the network. In particular, the flux space $\mathcal{F}$ could be seen as the product of the flux space of the network that results from the deletion of j and the interval $[lb_j, ub_j]$. If such a reaction exists in the network it can be discarded.

Proof. When s has more than one non-zero entries $s_i > 0$, the random variable Z defined in (5.2) is the minimum over at least two random variables $B_i s_i^{-1}$. Furthermore, we can exclude cases where two bounds refer to the same reversible reaction that was split up. In this case s would contain a futile cycle which can be subtracted from the flux mode. So we assume that all B_i, $i \in \mathrm{supp}(s)$ refer to genuinely different reactions and are independent random variables. The scalar multiples $B_i s_i^{-1}$, $i \in \mathrm{supp}(s)$ are then also independent. For every single $B_i s_i^{-1}$ we have $E(B_i s_i^{-1}) = E(B_i) s_i^{-1} = u b_j s_i^{-1}$, due to linearity of the expectation. To prove the proposition, we just have to show that the expectation of the minimum over at least two independent random variables is smaller than the minimum of the expectations of the individual random variables. This is shown in Lemma 5.2 below. □

If the B_i are normally or lognormally distributed, the support of $B_i s_i^{-1}$ is just the support of B_i, because $s_i > 0$ implies $s_i^{-1}\mathbb{R}_{>0} = \mathbb{R}_{>0}$ and $s_i^{-1}\mathbb{R} = \mathbb{R}$, so that Lemma 5.2 can be applied. The condition that $E(B_i) = u b_j$ is fulfilled for normal perturbations but in the lognormal case we have $E(B_i) > u b_j$ and Prop. 5.1 is not implied.

Lemma 5.2. *Let $X_i : \Omega \to \mathbb{R}$ be independent random variables on the sample space Ω with equal support $(a, b) \subset \mathbb{R}$ for $i = 1 \ldots, n$. Define $Z := \min(X_1, \ldots, X_n)$. The median of Z is strictly smaller than the median of X_i and the expectation of Z is strictly smaller than the expectation of X_i for all $i = 1 \ldots, n$.*

Proof. Let F_i be the cumulative distribution function (CDF) of X_i and f_i the probability density function (PDF) for $i = 1, \ldots, n$. Since $F_Z(x) = P(Z \le x) \Leftrightarrow (1 - F_Z(x)) = P(Z \ge x)$, we have

$$\begin{aligned} F_Z(x) &= P(Z \le x) = 1 - P(Z > x) \\ &= 1 - P(\min(X_1, \ldots, X_n) > x) = 1 - P(X_1 > x, \ldots, X_n > x) \\ &= 1 - \prod_{i=1}^{n} P(X_i > x) \\ &= 1 - \prod_{i=1}^{n} (1 - F_i(x)) \end{aligned} \tag{5.3}$$

Since (a, b) is the support of f_i, for a fixed $x \in (a, b)$ the corresponding CDF fulfils $0 < F_i(x) < 1$ for $i = 1, \ldots, n$. This gives $\prod_{j=1}^{n}(1 - F_j(x)) < (1 - F_i(x))$ which implies $F_Z(x) > F_i(x)$ for every $i = 1, \ldots, n$. If y_i is the median of F_i, defined by $F_i(y_i) = \frac{1}{2}$, then it follows that $F_Z(y_i) > \frac{1}{2}$. Since F_Z is strictly increasing on the support (a, b), we can conclude that the median y_Z of Z, defined by $F_Z(y_Z) = \frac{1}{2}$, is strictly smaller, i.e., $y_Z < y_i$ for all $i = 1, \ldots, n$. To see that also the expectation is smaller for Z we will use Lemma 5.3 to express the expectations:

$$\begin{aligned} E(Z) = \int_a^b x\, f_Z(x) dx \quad &= \int_0^b \prod_{j=1}^{n} (1 - F_j(x)) \mathrm{d}x - \int_a^0 \left(1 - \prod_{j=1}^{n} (1 - F_j(x))\right) \mathrm{d}x \\ E(X_i) = \int_a^b x\, f_i(x) dx \qquad &= \int_0^b (1 - F_i(x)) \mathrm{d}x - \int_a^0 (1 - (1 - F_i(x))) \, \mathrm{d}x \end{aligned} \tag{5.4}$$

Since $\prod_{j=1}^{n}(1 - F_j(x)) < (1 - F_i(x))$, it follows that $E(Z) < E(X_i)$, $i = 1, \ldots, n$. □

Lemma 5.3. *Let $X : \Omega \to \mathbb{R}$ be a random variable, with CDF F and PDF $f = F'$, the derivative of F. The expectation can then be given in terms of the CDF by $E(X) = \int_0^\infty (1 - F(x)) \mathrm{d}x - \int_{-\infty}^0 F(x) \mathrm{d}x$*

Proof. This can be shown with the fundamental theorem of calculus [Königsberger, 2001, p. 200]. Using the derivative of $x\,(F(x)-1)$, which is $x\,f(x)+(F(x)-1)$, we get

$$\int_0^\infty x\,f(x)\,dx = [x(F(x)-1)]_0^\infty - \int_0^\infty (F(x)-1)\,dx = \int_0^\infty (1-F(x))\,dx$$

Analogously, using $xF(x)$ with derivative $xf(x)+F(x)$ gives

$$\int_{-\infty}^0 x\,f(x)\,dx = [x(F(x)]_{-\infty}^0 - \int_{-\infty}^0 F(x)\,dx = -\int_0^\infty F(x)\,dx$$

and we can conclude that $E(X) = \int_{-\infty}^\infty x\,f(x)\,dx = \int_0^\infty (1-F(x))\mathrm{d}x - \int_{-\infty}^0 F(x)\mathrm{d}x$. □

The discussion here is focused on the expectation, but Lemma 5.2 shows that the situation is similar for the median, in particular Prop. 5.1 also holds if expectation is replaced by median.

5.4.2 Objective values in linear optimisation are deteriorated in expectation

After having analysed the effect of perturbations on the flux space we will now consider linear optimisation over this flux space. Since every fixed ray segment is getting smaller in expectation, this should intuitively also hold for the objective value in optimisation, but it is not straightforward to verify it formally. Here we will use projections of the flux space, given by the Fourier-Motzkin elimination method. Lemma 5.2 will be reformulated with another condition, which excludes in particular the case that a single non-redundant bound determines the whole network. The result is then obtained by a similar argument as in Prop. 5.1.

Problem specification The two cases of optimisation that we will consider here are the standard FBA

$$\begin{aligned} &\max c^\top v \text{ subject to} \\ &v \in \mathcal{F} = \{v \in \mathbb{R}^n \,:\, Sv = 0,\ 0 \le v \le ub\} \end{aligned} \tag{5.5}$$

and also the time dependent optimisation which was considered in the previous chapters and originally introduced (p. 36) as

$$\begin{aligned} &\min \tau \text{ s.t.} \\ &v \in \mathcal{F} \\ &S^*(\tau v) \ge \Gamma \\ &\tau \in \mathbb{R}_{>0}, \end{aligned} \tag{5.6}$$

where all components of Γ are strictly positive. If we assume that this problem is feasible, then $\tau > 0$ and τ^{-1} is well defined. As already mentioned in Sec. 2.3, the FBA problem (5.5) can be seen as a special case of the time minimisation (5.6). To express both LPs in the standard form, we write

$$A = \begin{pmatrix} 0 & Id \\ 0 & -Id \\ \Gamma & -S^* \\ 0 & S \\ 0 & -S \end{pmatrix}, d = \begin{pmatrix} ub \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{, and define } x := \begin{pmatrix} \tau^{-1} \\ v \end{pmatrix}$$

and the polytope $\mathcal{F}' := \{x \in \mathbb{R}^{n+1} : Ax \leq d\}$, so that (5.6) can be written as

$$\max\{x_1 : x \in \mathcal{F}'\} = \max\{x_1 : Ax \leq d\} \tag{OPT}$$

To express the LP (5.5) in this format, the constraints $S^*(\tau v) \geq \Gamma$ are translated to $x_1 \leq c^\top v \Leftrightarrow x_1 - c^\top v \leq 0$ by the identifications $S^* = c^\top$ and $\Gamma = 1$. Maximisation of x_1 is then equivalent to maximisation of $c^\top v$ and hence the resulting LP (OPT) is equivalent to (5.5).

Identifying the objective value by projection Now we will investigate the consequences of perturbations of the bounds ub in (OPT) on the objective value. Since the objective value of (OPT) depends only on the single variable x_1, it can be determined by using the Fourier-Motzkin Elimination method [Schrijver, 1998, Sec. 12.2] to project the whole flux space onto this variable. For the case of $\mathcal{F}$, this projection can be done by considering all the equalities between flux rates that are directly or indirectly given by $Sv = 0$ and substituting the bounding constraints $v \leq ub$ and $v \geq 0$ into these equalities, an example will be given below. In general, the Fourier-Motzkin Elimination to project the polytope $\mathcal{F}' := \{x \in \mathbb{R}^{n+1} : Ax \leq d\}$ onto the variable x_1 yields the following set of q linear inequalities:

$$x_1 \leq \sum_{i=1}^{2n+l+2m} \alpha_i^{(k)} d_i = \sum_{i=1}^{n} \alpha_i^{(k)} ub_i + 0, \quad k = 1, \ldots, q, \tag{5.7}$$

where l, m are the numbers of rows of S^*, S, respectively. The coefficients $\alpha^{(k)} \in \mathbb{R}$ emerge during the elimination procedure as linear combinations of entries of A. We get 0 for the sum $\sum_{j=n+1}^{2n+l+2m} \alpha_j^{(k)} d_j$, because $d_j = 0$ for $j = n+1, \ldots, 2n+l+2m$. The set of inequalities (5.7) is equivalent to $Ax \leq d$. We will abbreviate the right hand sides of the inequalities with $g_k := \sum_{j=1}^{n} \alpha_j^{(k)} ub_j$. The objective value of (OPT) is the maximal x_1 that still satisfies all these inequalities. It can directly be given as

$$x_1^* = \min_{k=1,\ldots,q} (g_k) \tag{5.8}$$

A perturbation of the bounds ub_i, given by the random variables $B_i : \Omega \to \mathbb{R}$, $i = 1, \ldots, n$, defines the random variables

$$G_k := \sum_{j=1}^{n} \alpha_j^{(k)} B_j \quad k = 1, \ldots, q.$$

Since the expectation is a linear function, we have $E(G_k) = \sum_{j=1}^{n} \alpha_j^{(k)} E(B_j) = \sum_{j=1}^{n} \alpha_j^{(k)} ub_j = g_k$. The objective value over the perturbed flux space is given as a random variable $\tilde{x}^* : \Omega \to \mathbb{R}$. It must satisfy for every sample $\omega \in \Omega$ the inequalities $\tilde{x}^*(\omega) \leq G_k(\omega)$, $k = 1, \ldots, q$, but no further constraints. Analogously to (5.8), $\tilde{x}^*$ is then given by

$$\tilde{x}^* = \min(G_1, \ldots, G_q) \tag{5.9}$$

To see, whether the objective value of (OPT) becomes smaller in expectation if the bounds are perturbed, the strict inequality $E(\tilde{x}^*) < x_1^*$ or, equivalently,

$$E\left(\min\left(G_1, \ldots, G_q\right)\right) < \min(g_k, \ldots, g_q) = \min\left(E(G_1), \ldots, E(G_q)\right) \tag{5.10}$$

has to be verified. If the random variables G_k were independent, Lemma 5.2 could be directly applied, since (5.10) is equivalent to :

$$E\left(\min\left(G_1, \ldots, G_q\right)\right) < g_k = E(G_k), \quad \text{for } k = 1, \ldots, q \tag{5.11}$$

However, the condition that the random variables G_k are independent is not given. They are linear combinations of the independent random variables B_i. If one B_i appears in the linear combinations G_k and $G_{k'}$, these two are not independent. Therefore we will formulate a similar Lemma, where the statement (5.11) is based on a different condition, namely

$$\begin{aligned} &\text{For each } G_k,\ k \in \{1, \ldots, q\} \text{ there exists a } k' \in \{1, \ldots, q\} \\ &\text{and a subset } \Omega' \subset \Omega \text{ with } P(\Omega') > 0 \\ &\text{such that } G_{k'}(\omega) < G_k(\omega) \text{ for each } \omega \in \Omega'. \end{aligned} \tag{COND}$$

Proposition 5.4. Consider the LP (OPT) and let the random variable B_j describe the perturbations of ub_j, such that $E(B_j) = ub_j$, $j = 1, \ldots, n$. Let $\{x_1 \in \mathbb{R} : x_1 \leq \sum_{j=1}^n \alpha_j^{(k)} ub_j,\ k = 1, \ldots, q\}$ be the projection on the first variable with coefficients $\alpha_j^{(k)} \in \mathbb{R}$. If the random variables $G_k := \sum_{j=1}^n \alpha_j^{(k)} B_j$ fulfil condition (COND), then the objective value of (OPT) becomes smaller in expectation.

Proof. If the condition (COND) is fulfilled, Lemma 5.5 can be applied and that establishes (5.11) and thus also the equivalent strict inequality (5.10), which states formally that the objective value of (OPT) becomes smaller in expectation. □

Lemma 5.5. *Assume $X_j : \Omega \to \mathbb{R}$, $j = 1, \ldots, n$, are random variables and for a given j there exists j' and a measurable subset $\Omega' \subset \Omega$ with $P(\Omega') > 0$ and $X_{j'}(\omega) < X_j(\omega)$ for $\omega \in \Omega'$. Then we have $E(\min_{i=1,\ldots,n}(X_i)) < E(X_j)$.*

Proof. Let $Z := \min_{i=1,\ldots,n}(X_i)$, so we have $Z(\omega) \leq X_j(\omega)$ for all $\omega \in \Omega$. Let j, j' and Ω' be as in the condition. Partitioning Ω into $\Omega' \dot\cup \Omega'' = \Omega$ (disjoint union, it follows that Ω'' is measurable as well), we can write the expectation value as

$$E(Z) = \int_\Omega Z(\omega) dP(\omega) = \int_{\Omega'} Z(\omega) dP(\omega) \ + \ \int_{\Omega''} Z(\omega) dP(\omega),$$

where the second term is $\leq \int_{\Omega''} X_j(\omega) dP(\omega)$ and the first term fulfils

$$\int_{\Omega'} Z(\omega) dP(\omega) \leq \int_{\Omega'} X_{j'}(\omega) dP(\omega) < \int_{\Omega'} X_j(\omega) dP(\omega).$$

The strict inequality holds because $X_{j'} < X_j$ on Ω' with $P(\Omega') > 0$. The expectation of X_j can be analogously expressed as

$$E(X_j) = \int_{\Omega'} X_j(\omega) dP(\omega) \ + \ \int_{\Omega''} X_j(\omega) dP(\omega)$$

and thus we have $E(Z) < E(X_j)$. □

Discussion of the condition The condition (COND) is tailored to establish the statement Prop. 5.4. Note that in some cases Lemma 5.2 and thus the proposition can also be proven by the same argument as in Lemma 5.2. This is the case if every G_k can be paired with another $G_{h(k)}$, such that this pair is independent (see the example below).

Now we will have a look at the implications of (COND) on the metabolic network model. In particular we want to know what kind of metabolic network models are excluded by (COND). In fact it is not too difficult to find metabolic networks models where (COND) is not given and where Prop. 5.4 does not hold. We can do this by considering the negation of (COND), which means there exists a $\bar{k}$, such that for all $k \in \{1, \ldots, q\}$ we have $G_{\bar{k}}(\omega) \leq G_k(\omega)$ almost surely (i.e. for all ω outside of a null set $\mathscr{O} \subset \Omega$, $P(\mathscr{O}) = 0$). This implies that $\min_{k=1,\ldots,q}(G_k) = G_{\bar{k}}$. In particular, the objective value $\widetilde{\tau}^*$ is just given by this single random variable, i.e. $\widetilde{\tau}^* = G_{\bar{k}}$. Independent of the perturbation, the network is thus always bounded by the same single linear constraint. The other constraints that were derived in the projection are not only redundant, but they also remain redundant after perturbation. Such a situation is of course given, if the network is only bounded by a single constraint by definition, as e.g. in FBA with molecular crowding (3.3), page 41. Another example is a metabolic network where only the uptake is bounded by a single constraint on one uptake flux or on the sum of several uptake fluxes, as discussed on page 31. (Formally, all other reactions j have upper bounds $ub_j = \infty$ and are not perturbed.) The expected objective value can then be given in terms of the single constraint alone and it is trivially not biased if the perturbation is not. Metabolic networks with a single bounding constraint therefore constitute a class of models where Prop. 5.4 cannot be applied. They can be seen as an extreme case. As will be shown in Figs. 5.7, 5.8, the interplay of several constraints determines the impact of perturbations on the objective value of optimisation. The effect described in Prop. 5.4 cannot occur if there is no interplay between several bounding constraints. This is in particular the case, if there is only one bounding constraint.

5.4.3 Example

The theory can be elucidated on a small example based on the network that was already used to introduce redundancy (p. 29). In fact, redundancy plays an important role for the effect of perturbations.

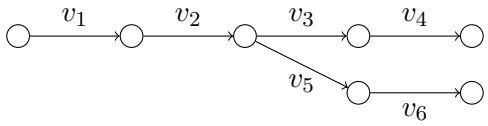


Figure 5.2: This network was analysed regarding redundancy in Sec 2.2.2. Now we consider optimisation of the objective function $c = (0, 0, 0, 1, 0, 1)$ over the flux space given by the depicted irreversible reactions with stoichiometric coefficients ± 1 and upper bounds ub.

The FBA optimisation problem $\max c^\top v = v_4 + v_6$ s.t. $Sv = 0,\ 0 \leq v \leq ub$, where

$$S = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

can be given in the standard form of (5.8), i.e., $\max\{x_1 : Ax \leq d\}$, with $A = \begin{pmatrix} 0 & Id \\ 0 & -Id \\ 1 & -c^\top \\ 0 & S \\ 0 & -S \end{pmatrix}$ and $d = \begin{pmatrix} ub \\ 0 \end{pmatrix}$, where 0 denotes the $6 + 1 + 4 + 4 = 15$-dimensional zero-vector. The steady-state assumption gives the equalities $v_1 = v_2,\ v_2 = v_3 + v_5,\ v_3 = v_4,\ v_5 = v_6$ and by substituting we get e.g. $v_1 = v_3 + v_5,\ v_1 = v_3 + v_6,\ v_1 = v_4 + v_6$, etc. Projecting can be done by substituting the bounds $v \leq ub$ into these equalities. For the target reactions $4, 6$ we get the get the six inequalities

$$\begin{array}{ll} v_4 + v_6 \leq ub_1, & v_4 + v_6 \leq ub_2, \\ v_6 \leq ub_6, & v_6 \leq ub_5, \\ v_4 \leq ub_4, & v_4 \leq ub_3. \end{array}$$

The above formulation of the optimisation problem introduced the new variable x_1, which is only constrained by $x_1 \leq c^\top v$. Maximising x_1 is therefore equivalent to maximising $c^\top v = v_4 + v_6$. Substituting the inequalities for $v_4,\ v_6$ we get the following bounds for the projection on x_1:

$$\begin{array}{ll} x_1 \leq ub_1 =: g_1, & x_1 \leq ub_2 =: g_2 \\ x_1 \leq ub_4 + ub_6 =: g_3, & x_1 \leq ub_3 + ub_6 =: g_4 \\ x_1 \leq ub_4 + ub_5 =: g_5, & x_1 \leq ub_3 + ub_5 =: g_6, \end{array}$$

The objective value is $x_1^* = \min(g_1, \ldots, g_6)$. In the perturbed case the bounds ub_j will be replaced by the random variables B_j and the g_k by G_k. Note that the pairs G_i, G_h with $(i, h) = (1, 2), (3, 6)$ or $(4, 5)$, are in fact independent, because they do not contain a common perturbation B_j. Therefore, we can apply Lemma 5.2 for these pairs and get

$$E(\tilde{x}_1^*) \leq E(\min(G_i, G_h)) < E(G_i), E(G_h), \quad (i, h) = (1, 2), (3, 6), (4, 5).$$

So we have in fact $E(\tilde{x}_1^*) < E(G_k) = g_k$ for $k = 1, \ldots, 6$ and hence, by (5.8), $E(\tilde{x}_1^*) < x_1^*$.

5.4.4 An asymptotic perspective

The previous results might be counter-intuitive at first sight. Why do perturbations which preserve the individual bounds in expectation, have a different effect globally? This phenomenon appears to be an inevitable consequence if we read the original bounds ub as a distinct point in $\mathbb{R}^n$ and the perturbations as steps of a random walk. From this perspective, the original choice of bounds turns out to be advantageous compared to a set of bounds which is randomly picked from $\mathbb{R}^n$. With very high probability such randomly picked bounds would not allow any flux through the network at all. Also random perturbations must therefore lead to a less favourable situation in expectation.

Iteration of perturbations We will examine the two examples of normally distributed perturbations $B_j := X_j + ub_j$ and of lognormal perturbations $B_j := Y_j\, ub_j = \exp(X_j)\, ub_j$, where $X_j \sim \mathcal{N}(0, \sigma_j^2)$. In case of the normal distribution, the iterated perturbations are given by $B_j^q := ub_j + \sum_{i=1}^q X_j^i$ and for the lognormal case by $B_j^q := ub_j \cdot \prod_{i=1}^q Y_j^i = ub_j \cdot \exp\left(\sum_{i=1}^q X_j^i\right)$, with independent $X_j^i \sim \mathcal{N}(0, \sigma_j^2)$.

Blocked reactions In this chapter we assumed that all reactions are irreversible and a fixed reaction j is hence blocked as soon as $ub_j = 0$. A normally distributed perturbation can block any reaction j immediately, because $P(X_j \leq -ub_j) > 0$. The probability that reaction j is blocked after q iterated normal perturbations is $P(\sum_{i=1}^q X_j^i \leq -ub_j)$. If the limit $\lim_{q\to\infty} P(\sum_{i=1}^q X_j^i \leq -ub_j)$ exists, we call it the probability that the reaction j is **asymptotically blocked**. Here we will show that this probability is $\frac{1}{2}$ for every fixed reaction j.

Iteration of normal perturbations The random variable $\sum_{i=1}^q X_j^i$ is again normally distributed with mean 0 and with variance $\sum_{i=1}^q \sigma_j^2 = q\,\sigma_j^2$ [Georgii, 2007, Korrolar 9.6, p. 235]. In particular, the variance is going to infinity. The statement that every reaction j is asymptotically blocked with probability $\frac{1}{2}$ is a direct consequence of the following Lemma.

Lemma 5.6. *Consider a sequence of normally distributed random variables with PDFs $f_q(x) = \frac{1}{\sigma_q\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2(\sigma_q)^2}\right)$ with mean μ and variance σ_q^2, such that $\sigma_q \to \infty$ as $q \to \infty$. For fixed μ and fixed $x \in \mathbb{R}$, the value of the corresponding cumulative distribution function $F_q(x)$ can be seen as a function of q and we have $\lim_{q\to\infty} F_q(x) = \frac{1}{2}$.*

Proof. Let $x \geq \mu$. Since the expectation μ is also the median, we have $\int_{-\infty}^x f_q(t)dt = P(X \leq x) = F_q(x) \geq \frac{1}{2}$ and $\int_{-\infty}^\mu f_q(t)dt = \frac{1}{2}$. This gives

$$0 \leq F_q(x) - \frac{1}{2} \;=\; \int_{-\infty}^{\mu} f_q(t)dt + \int_{\mu}^{x} f_q(t)dt - \frac{1}{2} \;=\; \int_{\mu}^{x} \frac{1}{\sigma_q\sqrt{2\pi}}\exp\left(-\frac{(t-\mu)^2}{2(\sigma_q)^2}\right)dt.$$

By substituting $z = \frac{t-\mu}{\sigma_q}$, $dz = \frac{1}{\sigma_q}dt$ we get

$$0 \leq F_q(x) - \frac{1}{2} \;=\; \int_0^{\frac{x-\mu}{\sigma_q}} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z^2}{2}\right)dz \;\leq\; \frac{x-\mu}{\sigma_q} \;\to 0 \quad \text{as} \quad q \to \infty.$$

For the case $x \leq \mu$ the proof works analogously, just note that $\int_\mu^x f(t)dt = -\int_x^\mu f(t)dt$. □

Iteration of log-normal perturbations The situation is a bit different for the iteration of the lognormal perturbations given by $B_j^q := ub_j \cdot \prod_{i=1}^q Y_j^i = ub_j \cdot \exp\left(\sum_{i=1}^q X_j^i\right)$, with independent $Y_j^i \sim log\mathcal{N}(0, \sigma_j^2)$. A reaction cannot be totally blocked after a finite number of perturbations, because $Y_j^i > 0$, $i \in \mathbb{N}$ and thus also $\prod_{i=1}^q Y_j^i > 0$ for every $q \in \mathbb{N}$. However, the perturbed bound can become arbitrarily small and from the practical point of view a reaction can be considered to be blocked as soon as the bound is smaller than some given threshold $\varepsilon > 0$. Therefore, we adjust the notion of asymptotically blocked. If the limit $\lim_{q\to\infty} P(ub_j \exp\left(\sum_{i=1}^q X_j^i\right) \leq \varepsilon)$ exists, we call it the probability that the reaction j is **asymptotically ε-blocked**. It follows directly from Lemma 5.6 that this limit probability as well is $\frac{1}{2}$, because we have

$$P\left(\exp\left(\sum_{i=1}^q X_j^i\right) \leq \frac{\varepsilon}{ub_j}\right) = P\left(\sum_{i=1}^q X_j^i \leq \log\left(\frac{\varepsilon}{ub_j}\right)\right) = F_q\left(\log\left(\frac{\varepsilon}{ub_j}\right)\right).$$

Consequences for the flux space For a single reaction j we saw that the probability to become asymptotically blocked is $1/2$. If we look at the whole metabolic network, we note that the fulfilment of any specific metabolic function requires a group of several reactions to be active. As we just saw, every single reaction becomes asymptotically blocked with probability $1/2$. If the number of the necessary reactions for a given metabolic function is k, then it follows that perturbations will disable this function asymptotically with probability $1 - (1/2)^k$. To get an idea of realistic values for k, elementary modes, the flux modes with minimal support, can be helpful. In the core metabolic network from Chap. 4 for example, the length of an EM is at least 11, which gives $1 - (1/2)^{11} \geq 1 - 10^{-3}$ as the probability that the network allows asymptotically no flux at all. This core network contains only 46 reactions, because functional units as e.g. the citric acid cycle are lumped together to one reaction. If these would be represented explicitly the probability for complete blockade would be even larger. In conclusion we can say that the iteration of perturbation of all bounds disables with high probability any function of a metabolic network asymptotically. The capabilities of the network are progressively inhibited and therefore, we also expect that already one of the indistinguishable iteration-steps is lowering the capabilities as is in fact observed in Prop. 5.1 or Prop. 5.4. These observations and conclusions are of course based on the assumption that all reaction bounds are progressively perturbed and can approach 0. As already mentioned and as will be analysed in detail in the following section, the situation is different if only a subset of reactions is bounded or if only a subset has bounds that have a chance to be non-redundant after perturbation. If this subset is sufficiently small, the effect of a decrease of the capability, that is, of the objective value in optimisation can even completely vanish, as was formally discussed for simple examples, see page 115, and also observed in the core network, see Figs. 5.7, 5.8.

5.5 Empirical quantitative analysis

The qualitative results from the analytical investigation above will now be substantiated by an empirical analysis. It will show that the strict inequality from Lemma 5.2 describes indeed a significant gap between the two expectation values. Also the decrease in the objective value in optimisation turns out to be significant, with normal as well as with lognormally distributed random variables. For the lognormal distribution this was not proven here. The dependence of the effects on the structure of the flux space representation is then investigated. We identify two extreme cases. If all bounds are weakly redundant, the objective value is strongly decreased, but if redundant bounds were discarded before, no decrease at all is observed.

5.5.1 Dependence on the distribution of the perturbations

The theoretical results were obtained for distributions with $E(B_j) = ub_j$. But if we consider the lognormal distributions $B_i := Y_i\, ub_i$ with $Y_i \sim log\mathcal{N}(\mu, \sigma^2)$, the expectation of an individual Y_i is $E(Y_i) = \exp(\frac{1}{2}\sigma_i^2) > 1$ and thus $E(B_i)$ is far above ub_i for large σ_i^2. With theoretical means it is difficult to quantify the effects of perturbations. For $Z = \min(X_1, \ldots, X_n)$, the CDF was

given by (5.3), this gives the PDF

$$f_Z(x) = F'_Z(x) = \sum_{j=1}^{n} f_j(x) \prod_{i \neq j} (1 - F_i(x)), \tag{5.12}$$

where $f_j = F'_j$ is the PDF of X_j. In particular, for normally distributed X_j we get

$$f_Z(x) = \sum_{j=1}^{n} f_j(x) \prod_{i \neq j} \left(\frac{1}{2} - \frac{1}{2} \mathrm{erf}\left(\frac{x}{\sqrt{2\sigma_i^2}} \right) \right). \tag{5.13}$$

It is not possible to read off something about the dependence of $E(Z)$ on n or on the variance here. The CDF of the (log-)normal distribution does not even have a closed form expression (the error function 'erf' is needed) and in the lognormal case we are confronted with the problem that the CDF of a sum of lognormal distributions cannot be given explicitly, but has to be approximated [Beaulieu and Xie, 2004]. On the other hand, values of the PDF and CDF of the normal and lognormal distribution can be computed very precisely. This is also possible for F_Z and f_Z by using (5.12) and (5.13), respectively. The following plots elucidate the dependence of $E(Z)$ on n and on σ_j^2. For $n = 2$, Fig. 5.3 shows the dependence of $E(Z) = E(\min(X_1, X_2))$ and of the median of Z on the variances of the two normally distributed random variables X_1, X_2. For a pair of lognormally distributed X_1, X_2 the analogous results are shown in Fig. 5.4.

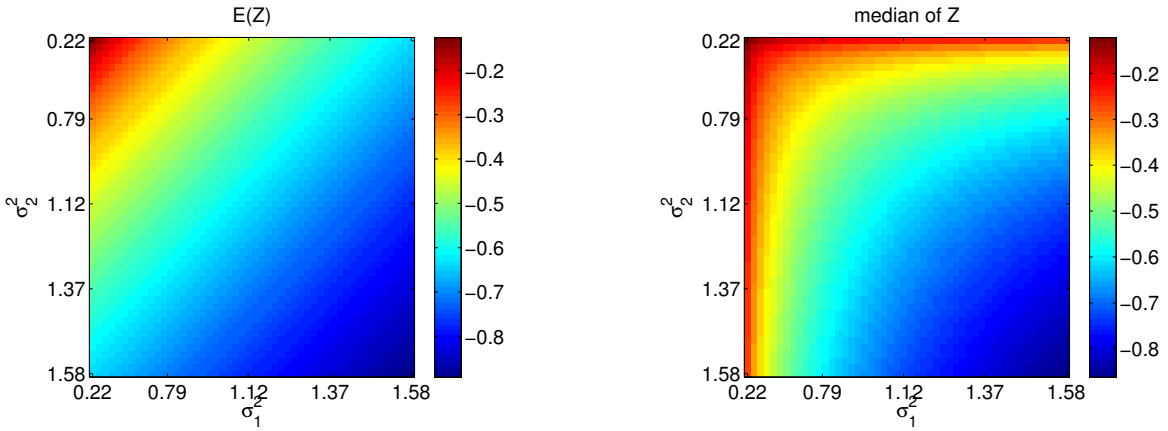


Figure 5.3: Mean and median of $Z := \min(X_1, X_2)$, with normally distributed X_1, X_2. Their variance is given on the axes and the expectation resp. median is indicated by colour. The values lie strictly below the expectation and the median of an individual X_i, which are 0 for $i = 1, 2$. As the variance increases, the expectation of Z decreases depending on $\sigma_1 + \sigma_2$. The median also decreases but with stronger dependence on the smaller variance.

The question if $E(Z) < 1$ in the lognormal case is important regarding the flux space perturbations. Since $E(X) = \exp(\frac{1}{2}\sigma^2) > 1$ for a lognormal random variable $X \sim log\mathcal{N}(0, \sigma^2)$, an individual bound is in expectation larger after perturbation. On the other hand the minimum over several constraints leads to a smaller expectation, which can affect a single ray segment of the flux space (Prop.5.1) as well as the objective value in optimisation (Prop.5.4). The empirical analysis shows that the decreasing effect of the minimum over several random variables largely

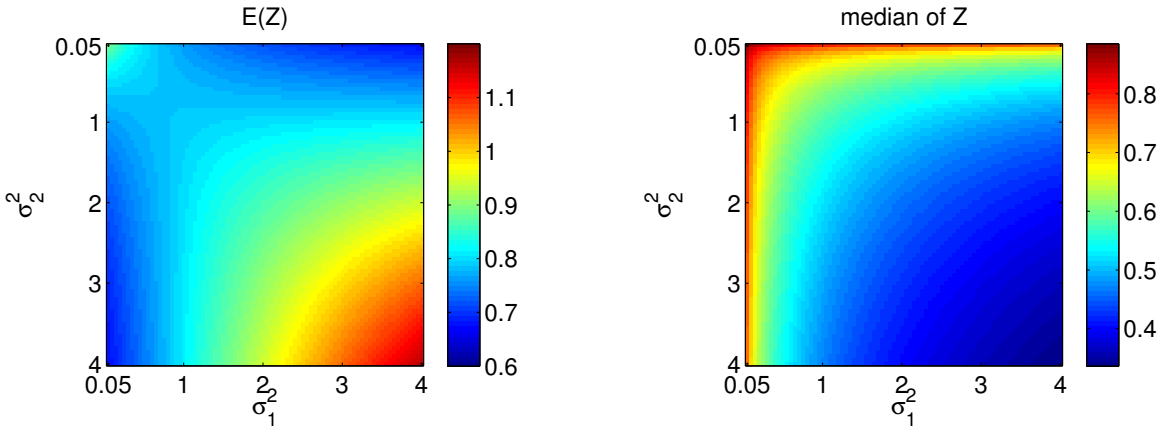


Figure 5.4: Mean and median of $Z := \min(X_1, X_2)$, with lognormally distributed X_i. Since the expectation of a lognormal $X_i \sim log\mathcal{N}(0, \sigma_i^2)$ is $\exp(\frac{1}{2}\sigma_i^2)$, it is far above 1 for large σ_i^2. Also if the minimum of two X_i is taken, this still leads to an expectation above 1 as soon as both variances are big enough. $E(Z)$ takes its smallest values if X_1 has a small variance, but X_2 a big one (or vice versa). In this case large deviations of X_2 above 1 are prevented by X_1, while large deviations below 1 can contribute to the expectation $E(\min(X_1, X_2))$.

dominates the larger expectation of the lognormal distribution, in particular the qualitative statement of Lemma 5.2 also holds in the lognormal case and is significant, at least for a variance in a reasonable range. Fig. 5.5 shows that for $X \sim log\mathcal{N}(0, \sigma^2)$, the expectation of the minimum will be significantly below 1, as soon as $n > 5$ and $4 \geq \sigma^2 \geq 0.5$. Variances above 3 can be considered as not relevant for our modelling purpose. With arbitrarily large variances we can in fact achieve for any fixed n that $E(Z) = E(\min(X_1, \ldots, X_n))$ becomes arbitrarily large.

Lemma 5.7. *For given $n \in \mathbb{N}$, $K \in \mathbb{R}$ and $X_i \sim log\mathcal{N}(0, \sigma^2)$, $i = 1, \ldots, n$, we have*

$$E(Z) = E(\min(X_1, \ldots, X_n)) \geq K$$

as soon as $\sigma \geq \frac{\log(K)+2n\log(2)}{\mathrm{erf}^{-1}(1/2)\sqrt{2}}$, where $\mathrm{erf}^{-1}\colon (0,1) \to \mathbb{R}$ is the inverse of the error function.

Proof. Let F_Z be the CDF of Z and F_X the CDF of the independent and identically distributed X_i, $i = 1, \ldots, n$. For any $x \in \mathbb{R}$ an ample estimate is given by $E(Z) \geq x \cdot P(Z \geq x) = x \cdot (1 - F_Z(x))$. We know that $F_Z(x) = 1 - (1 - F_X(x))^n$ and conclude that $E(Z) \geq (1 - F_X(x))^n x$. Define $a(x) := (1 - F_X(x))^n$, then $1 - a(x)^{\frac{1}{n}} = F_X(x)$. The goal is now to find x and σ such that $E(Z) \geq a(x)x \geq K$. Since we are interested in $K > 1$, also $x > 1$ must hold. The median of X is 1 for any σ, i.e., $F_X(1) = \frac{1}{2}$. Therefore, $x > 1$ implies $1 - a(x)^{\frac{1}{n}} = F_X(x) > \frac{1}{2} \Rightarrow a(x) < \left(\frac{1}{2}\right)^n$. Let us restrict x to the range where $a(x)$ fulfils

$$\left(\frac{1}{2}\right)^{2n} < a(x) < \left(\frac{1}{2}\right)^n \quad \Leftrightarrow \quad \frac{3}{4} > 1 - a(x)^{\frac{1}{n}} = F_X(x) > \frac{1}{2}.$$

The desired lower bound $E(Z) \geq K$ is then established if $K \leq \left(\frac{1}{2}\right)^{2n} \cdot x < a(x) \cdot x \Leftrightarrow x \geq K2^{2n}$. Apart from this bound, the choice of x is only limited by the upper bound $\frac{3}{4}$ on the CDF, so we

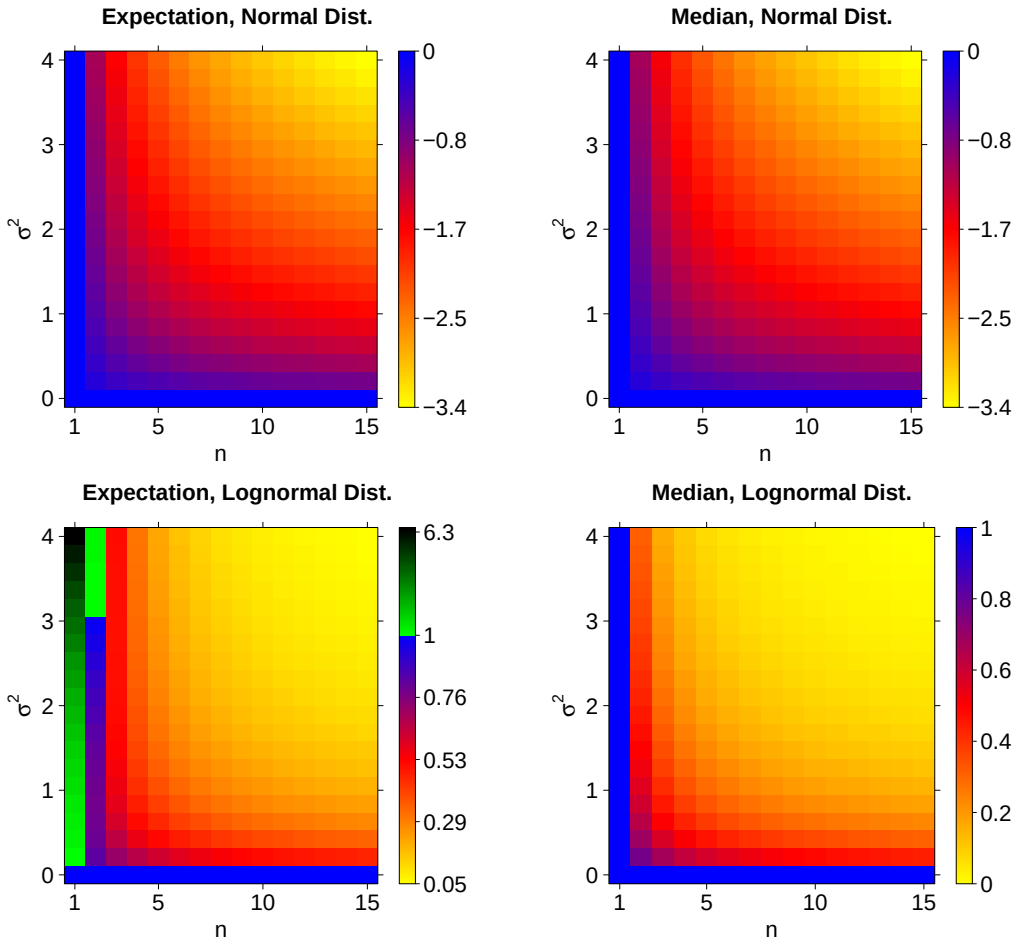


Figure 5.5: Dependence of $E(Z) = E(\min(X_1, \ldots, X_n))$ and the median of Z on $n = 1, \ldots, 15$ and on a varying variance, which is identical for all X_i. The random variables are independent and either all normally distributed, i.e., $X_i \sim \mathcal{N}(0, \sigma^2)$, or lognormally distributed, i.e., $X_i \sim log\mathcal{N}(0, \sigma^2)$, $i = 1, \ldots, n$. In case of normal distribution, the value of $E(Z)$ decreases with variance as well as with the number n of random variables in the minimum. The outcome is almost identical for the median. In the lognormal case, we have the antagonistic effects that the expectation increases with the variance, while the minimum decreases as n gets bigger. For $n = 2$ we still have $E(Z) > 1$ if $\sigma^2 > 3$, but for $n > 2$, the variance would have to be even larger to achieve $E(Z) > 1$. As soon as e.g. $n \geq 5$ and $4 \geq \sigma^2 \geq 0.5$, the values of $E(Z)$ are below 0.5. In general we can always achieve $E(Z) \geq K$ for any $K > 1$ by a sufficiently large variance as shown by the estimation of Lemma 5.7. However, a variance of lognormal perturbations with σ^2 above 3 is not reasonable for parameter perturbations. Already with $\sigma^2 = 3$ the probability of perturbing a given bound by a factor of 2 or $1/2$ is about 0.69 and for a factor $10^{\pm 1}$ we still get a 18% chance. Concluding, we can say that also in the case of lognormal perturbations, the effect given in Lemma. 5.2 occurs with significant magnitude in all relevant cases.

have

$$\frac{3}{4} > F_X(x) > F_X(K2^{2n}) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\log(K)2^{2n}}{\sigma\sqrt{2}}\right) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\log(K) + 2n\,log(2)}{\sigma\sqrt{2}}\right)$$

$$\Leftrightarrow \quad \frac{1}{2} \geq \mathrm{erf}\left(\frac{\log(K) + 2n\,log(2)}{\sigma\sqrt{2}}\right) \quad \Leftrightarrow \quad \sigma \geq \frac{\log(K) + 2n\,\log(2)}{\mathrm{erf}^{-1}(1/2)\sqrt{2}} \approx 3(\log(K) + 2n\log(2))$$

This inequality can always be satisfied by a large enough σ, which proves the Lemma. □

5.5.2 Dependence on the bounding constraints

Finally, the effect in optimisation problems over $\mathcal{F}$ and $\widetilde{\mathcal{F}}$ was analysed empirically for the core metabolic network from Chap. 4. In the discussion of the condition (COND) on page 115 it was pointed out that the relationship between all bounding constraints determines the strength of the effect. As an extreme case the example of a network that is bounded by a single constraint was given. The size of the ray segments as well as the objective value of optimisation after an unbiased perturbation of that single constraint is then trivially not biased in expectation. In particular the condition (COND) of Prop. 5.4 is not fulfilled. For the LP (5.8) only the projection onto the maximised variable is relevant. Even if the original description contains several bounding constraints it might lead to a single bound on this variable, an example is given in Fig. 5.6.

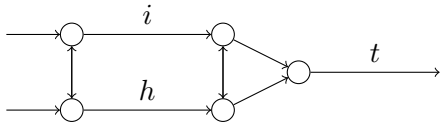


Figure 5.6: Assume that only the reactions i and h are bounded from above in this example network. These two bounds are surely non-redundant and the fluxes are independent of each other, i.e., the reactions are uncoupled (Def. 2.5). However, the flux through the target reaction t is only bounded by one constraint, namely $v_t \leq ub_i + ub_h$. The conditions of Prop. 5.4 are hence not fulfilled and in fact the expected objective value $v_t^* = ub_i + ub_h$ will not be biased, i.e., if $E(B_i) = ub_i$ and $E(B_h) = ub_h$, then we will also have $E(v_t^*) = ub_i + ub_h$.

If in the example of Fig. 5.6 the reactions apart from i and h were bounded as well, we would have several constraints on the objective value and Prop. 5.4 would apply (given suiting distributions of the perturbations, e.g. normal distributions).

These examples showed that having only few non-redundant bounding constraints in a network can lead to the extreme case where the objective value is not at all biased in expectation. The other extreme is a network with stoichiometric bounds obtained from FVA (p. 29). These bounds are all redundant, but many are weakly redundant, which means that an arbitrarily small perturbation can make them non-redundant. Geometrically speaking the n constraints are all touching the flux space, in particular each of them has a high chance to further restrict the flux space after a perturbation. Intuitively, we would therefore expect that the more bounding constraints and the closer they are to the flux space (if they are redundant), the stronger decrease of the objective value in expectation after perturbation. The results of the empirical analysis supports this hypothesis. If we have stoichiometric bounds, the objective value after perturbation is strongly decreased in expectation, see Fig. 5.7 B). In contrast, if we reduce the representation of the flux space to a subset of non-redundant bounds, the expected objective value is not at all decreased, see Fig. 5.7 C). The original bounds from literature give a scenario that lies between these two extreme cases. We have bounds for almost all reactions and therefore many more than in a non-redundant subset, but contrary to the stoichiometric bounds, they are not all located close to the flux space, which means that not all are likely to constrain the flux space

after a perturbation. The decrease in the expectation of the objective value, Fig. 5.7 A), is then significant but moderate compared to Fig. 5.7 B).

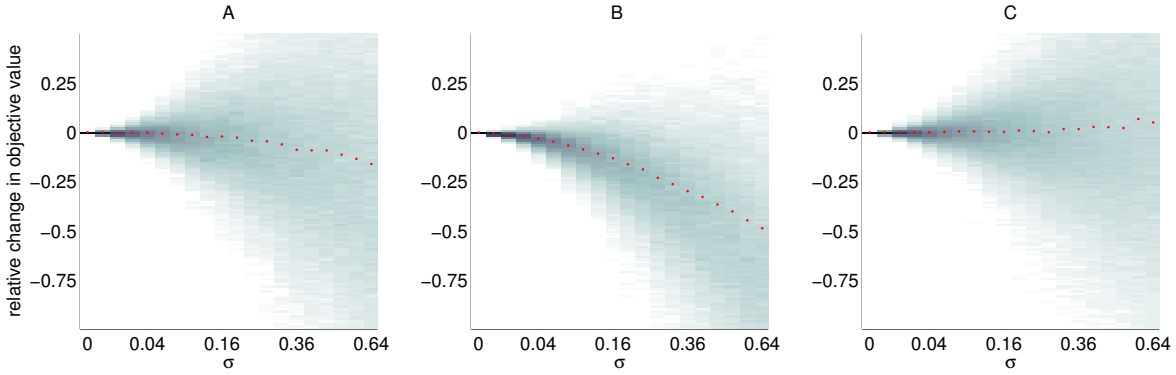


Figure 5.7: In the small core metabolic network, introduced in Chap. 4, Fig. 4.3, the bounds were perturbed by multiplication with $Y_i \sim log\mathcal{N}(0, \sigma^2)$, for varying σ^2. On the x-axis we have 20 different choices of the variance. For each of these σ fixed, 400 perturbed bounds $\widetilde{ub}_j$ were sampled and used to solve the linear FBA problem $\max\{c^\top v \ : \ v \in \widetilde{\mathcal{F}}\}$, with 15 different objective functions c. For each of this $400 \times 15 = 6,000$ LPs the objective value was compared to the corresponding objective value of the same optimisation over the unperturbed flux cone, i.e. $\max\{c^\top v \ : \ v \in \mathcal{F}\}$, by taking the ratio of the two objective values. All ratios $\{r_1, \ldots, r_{6000}\}$, corresponding to one fixed σ, were binned to obtain a density plot for each σ. On the y-axis the ratios minus one, i.e., $r_i - 1$, $i = 1, \ldots, 6000$ are plotted. Values below zero indicate that the objective value is smaller over the perturbed flux space. The red dotted line indicates the average $\frac{1}{6000}(r_1 + \cdots + r_{6000}) - 1$ for each variance σ. In **A**), the literature parameters are used to fix the bounds ub. In **B**) the corresponding stoichiometric bounds (Def. 2.4) are used and in **C**) the set of constraints $v \leq ub$ was reduced to a subset of only non-redundant bounds describing the same (unperturbed) flux space.

For normal perturbations the corresponding results are shown in Fig. 5.8. Qualitatively they are not different. The decrease in the objective values is even stronger than with the lognormal perturbations. A possible explanation is that the normal perturbations are frequently blocking reactions, since $P(B_j \leq 0) > 0$, whereas in the lognormal case $P(B_j \leq 0) = 0$. This explanation is supported by the fact that the number of perturbations giving an objective value of zero (meaning that the biological function which is expressed in the objective function c is completely disabled by the perturbation) was much higher in the computations with the normal distributions (data not shown in the plots).

5.6 Conclusion

The observations stated in Prop. 5.1 and Prop. 5.4 are surprising at first, because unbiased perturbations of the parameters lead to a biased effect on the whole model. The reason for this effect are the flux couplings and redundancies in the bounding constraints $v \leq ub$ resulting from the strict steady state condition $Sv = 0$. This is an example of an unexpected model property which was not intentionally incorporated in the construction. In general it is important to identify such properties, in order to avoid misinterpretations. The asymptotic viewpoint was given to offer an alternative explanation which can be interpreted in an evolutionary manner. In fact, the observation that random perturbations in the metabolic network impair the functionality

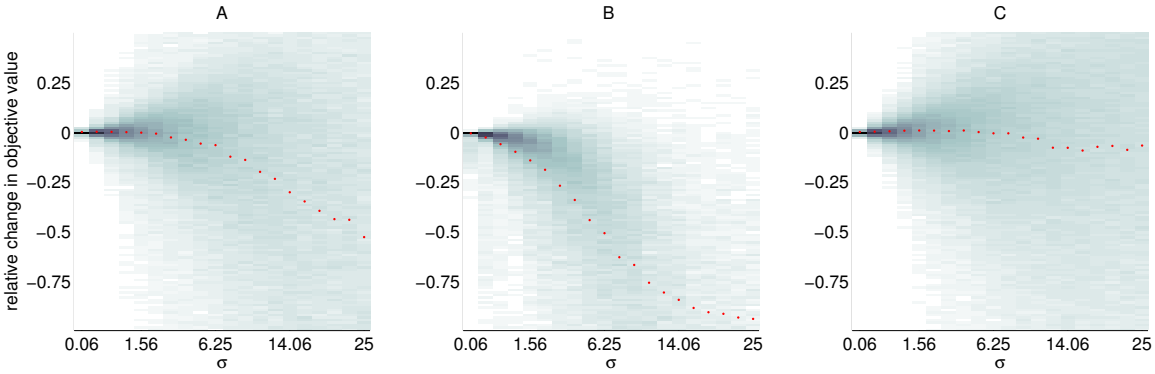


Figure 5.8: Results of the computations as for Fig. 5.7, but with normally distributed perturbations, i.e., $E(Z) = \min(X_1, \ldots, X_n)$ and $X_i \sim \mathcal{N}(0, \sigma^2)$. The variance was chosen much larger, because it defines the absolute magnitude of the perturbation and therefore has to be adjusted to the average of the turnover numbers ub.

in expectation is not surprising if we assume that the network has evolved to its given form in order to provide this functionality. The performance of the network with the original parameters should therefore be above the average of all biologically feasible sets of parameters.

An interesting aspect is that the effect does not always occur, but depends on the interplay of several bounds that are coupled and that are non-redundant or can become non-redundant by perturbation. Some examples of models were given where this condition is not fulfilled. These examples and the corresponding formal considerations showed that an interplay of redundancy before and after perturbation determines if the effect is occurring or not. Furthermore, flux coupling plays a role, because it might tell us if a redundant constraint is likely to become non-redundant. Most likely this relationship can be formulated to give equivalent versions of the condition (COND) in terms of redundancy and flux couplings, however this direction was not pursued here.

In general, a complete realistic model of metabolism, with maximal flux rates for almost all individual reactions, can be expected to satisfy the condition (COND) and to exhibit a significant decrease in expectation of the objective values in optimisation. The core metabolic network from Chap. 4 was employed to illustrate the effect and also its dependence on the representation of the flux space.

Different representations of the flux space, namely stoichiometric bounds or a subset of non-redundant bounds, can be obtained computationally and the outcome of the perturbations is very different in both cases. As a conclusion of this observation, we note that meaningful biological information is only represented by the original bounds of the metabolic network model, but not by the stoichiometric bounds or a subset of non-redundant bounds. The latter representations are only useful for the mathematical handling of the model.

6 Computation of optimal or approximated solutions

6.1 Overview

The method for modelling resource allocation and alterations in metabolism presented in Chap. 3 is based on optimisation problems which are hard to solve computationally. The question of complexity is not considered in detail here, but a short overview of different aspects of solving these MIQCP (mixed-integer quadratically constrained programming) problems is given in the next section. Reducing the complexity by a divide and conquer algorithm is in general not promising, as shown with a counterexample. Alternative approaches will then be presented and discussed. To find approximative optimal solutions, a pre-selection, as introduced in Sec. 3.8, can be constructed. Furthermore, we examine the possibility of computing optimal solutions for all sequences of gene activation patterns which have been confirmed to be relevant. This becomes interesting as more information about the gene regulatory network gives further restrictions on the gene expression space and on its dynamics, thus reducing the number of the relevant sequences. In general, these approaches replace one hard to solve MIQCP problem with a large number of smaller problems. This brings a gain in tractability, because the computation of these reduced problems requires an exponentially smaller amount of time and space.

6.2 Complexity of the optimisation problems

Problem formulation We will consider again the general optimisation problem (OP3), p. 58, of the resource allocation model in Chap. 3. Without maintenance constraints, it can be formulated as a mixed-integer quadratic programming (MIQP) problem, i.e., without quadratic constraints, but with a quadratic objective function. This reformulation is done by substituting

$\tau'_k := \tau_k \langle g^k, \gamma \rangle^{-1}$ and $w^k := \tau_k v^k$, which gives

$$
\begin{aligned}
&\min \sum_{k=1}^{l} \tau'_k \langle g^k, \gamma \rangle \text{ s. t.} \\
&S w^k = 0 \quad k = 1, \ldots, l \\
&\tau'_k (A_{tot} k c_j^-) \leq w_j^k \leq \tau'_k (A_{tot} k c_j^+), \quad k = 1, \ldots, l,\ j = 1, \ldots, n \\
&g_j^k = 0 \Rightarrow w_j^k = 0 \quad k = 1, \ldots, l,\ j = 1, \ldots, n \\
&g^k \in G, \quad k = 1, \ldots, l \\
&\sum_{k=1}^{l} w_t^k \geq \Gamma_t, \text{ for } t \in T \\
&\text{and variables } \tau'_k \in \mathbb{R}_{\geq 0},\ w^k \in \mathbb{R}^n,\ g^k \in \{0,1\}^n .
\end{aligned}
\tag{OP3b}
$$

The equivalence of (OP3), p. 58, and (OP3b) follows from the same arguments as given on page 38 for (OP1) and (OP1a). As soon as maintenance constraints are included in the model it is not possible anymore to avoid quadratic constraints. The flux is constrained on the one hand by the maintenance constraints, which are independent of resource allocation, and on the other hand by the lower and upper bounds, which are scaled by $\langle g^k, \gamma \rangle^{-1}$. One of these two groups of constraints must be expressed with quadratic constraints, depending on whether the original flux rates v_j^k are the variables as in (OP3) or instead $w_j^k := v_j^k \langle g^k, \gamma \rangle$ as in (OP3b), in which case the addition of maintenance constraints $mb_i \leq v_i^k$ is translated to the quadratic constraints $mb_i \langle g^k, \gamma \rangle \leq w_i^k$, $k = 1 \ldots, l$. In (OP3b) on the other hand, the bounds on v^k are given by quadratic constraints while the maintenance for reaction i can be included linearly as $mb_i \leq v_i^k$, $k = 1 \ldots, l$.

Avoiding a quadratic objective function In fact, it is even possible to formulate the MIQP problem (OP3b) without maintenance constraints as an equivalent mixed integer linear problem with linear objective function (MILP) [Bisschop, 2010, Sec. 7.7]. This is conceptually a simplification of the problem, but as it comes at the cost of introducing many new integer variables (in our case the number of binary variables is doubled) it might not bring a practical improvement of computational tractability. For the core metabolic network without maintenance constraints, using this linearisation is in fact slightly increasing the computation time (all computations were done with Gurobi Optimizer 5.6 (`http://www.gurobi.com`)).

Support variables In (OP3), p. 58, the binary variables g^k are constraining the flux mode v^k by $g_j^k = 0 \Rightarrow v_j^k = 0$. To implement this constraint, it can be formulated as

$$
-M g_j^k \leq v_j^k \leq M g_j^k \tag{6.1}
$$

for some $M \in \mathbb{R}_{\geq 0}$ sufficiently big to render these constraints redundant in case $g_j^k = 1$. For (OP3) this is fulfilled by $M := A_{tot} (\min_{g \in G} \langle g, \gamma \rangle)^{-1} \cdot \max_{j=1,\ldots,n} (-kc_j^-, kc_j^+)$. In fact, the minimum could be restricted to those $g \in G$ where $\mathcal{F}_g$ contains a non-zero flux mode. For (OP3b) we consider $w^k := \tau_k v^k$ instead of v^k, so M has to be multiplied by a factor $\bar{\tau}$ with $\bar{\tau} \geq \tau_k$ for all $k = 1, \ldots, l$. Such $\bar{\tau}$ can be computed by solving the LP that we get from (OP3b) with $l = 1$ and fixing all g_j^1 to 1 (this was called the reference case for the core metabolic network in Chap. 4 and A_{tot} was adjusted such that $\bar{\tau} = 8$ [h]).

Numerical precision Numerical errors are common in solving hard MILPs. They are often caused by values close to zero. Due to the finite precision of floating point arithmetic, the computational procedure cannot distinguish zero from values below a certain threshold ε (tolerance). During the computation, much smaller numbers than those in the input of the problem can occur. In (OP3b) this can happen e.g. if a small flux rate v_j^k is combined with a short duration τ_k, resulting in a very small value $w_j^k = \tau_k v_j^k$.

Complexity of linear problems The complexity of linear programming was thoroughly investigated but some questions are still open. Since the ellipsoid method was developed, we know that an optimal solution can be found in polynomial time in the dimension of the problem [Megiddo, 1987]. However, the practical performance of this algorithm cannot compete with the simplex method. All known variants of the simplex method have in the worst case an exponential running time in the size of the problem, but they exhibit polynomial and often even linear running times for a large class of problems. Interior point methods [Karmarkar, 1984] are polynomial time algorithms that can also compete with the simplex method. State of the art software is using both, the simplex method and interior point methods to find an optimal solution.

Complexity of mixed integer linear problems When integer variables, or as in our case binary variables, are included in the optimisation problem, the running time as well as the required space are tremendously increased. Solving MILPs is usually done by a Branch and Bound method (or Branch and Cut, which means that cutting planes are used in addition). These methods include implicitly a tree search on all 2^n possible assignments of the n binary variables. In every node of the tree an LP (a relaxation of the MILP) has to be solved. This tree search is called Branch and Bound method. Due to the size of the tree, even though some parts can be discarded, the number of LPs that have to be solved may be exponential in the number of binary variables. The running time may hence also increase exponentially compared to an LP of the same size. More problematic for the practical computation is usually the space consumption that comes with the tree search. As the tree is explored, a large number of nodes with partial solutions (which might be the optimal solution or an ancestor of an optimal solution) must be stored. Infeasible assignments of the binary variables do not have to be stored. In particular, only those gene expression patterns g that are defined as feasible in the model, i.e., $g \in G$, have to be tested during Branch and Bound. Therefore, the size of G is a critical parameter for the tractability of the optimisation problem.

Conclusion The optimisation problem (OP3), p. 58, contains $l \cdot n + l$ continuous variables for the l flux modes and the durations $\tau_1, \ldots, \tau_l$ and furthermore $n \cdot l$ integer variables for g_j^k, $k = 1, \ldots, l$, $j = 1, \ldots, n$. In the case where the gene expression space in generated by minimal gene sets (MGS, see Sec. 3.6.3) there will be an additional integer variable for each MGS. Nevertheless the complexity of the problem decreases if MGS are included, because the size of the gene expression space G and thus the tree for Branch and Bound is reduced. Since $l \leq |T|$, the number of target reactions and the total number of reactions are determining the problem size. In practice it turns out that complexity is reduced significantly by restrictions on G (e.g. MGS constraints) as well as on the allowed sequences of gene expression states (e.g. monotone sequences in Sec. 4.4.1). In the application to the core metabolic network of Chap. 4 it was still tractable to solve all involved optimisation problems with Gurobi Optimizer 5.6 (`http://www.gurobi.com`). However, as the size of the network increases and with more

target reactions, we are quickly confronted with an intractable optimisation problem. Therefore, alternative ways to obtain solutions will be explored below.

6.3 Estimation of the objective value

For a given target reaction $t \in T$, let $v^{(t)}$ be an optimal solution of

$$\begin{aligned} &\max\ v_t \text{ s.t.} \\ &v \in \mathcal{F}^* \\ &v \geq mb \\ &\text{with variables } v \in \mathbb{R}^n \end{aligned}$$

where $\mathcal{F}^*$ is the flux space and $v \geq mb$ are the maintenance constraints as defined in Sec. 3.7.2. The demanded total turnover of reaction t can be achieved in time $\Gamma_t(v^{(t)})^{-1}$ [h], but not in less time. This gives a straightforward estimate for τ^*, the objective value of (OP2a), p. 55:

$$\max_{t \in T}(\Gamma_t(v^{(t)})^{-1}) \leq \tau^* \leq \sum_{t \in T} \Gamma_t(v^{(t)})^{-1}$$

It is not difficult to give examples that each of the bounds can be attained. The minimal example from Fig. 4.1 attains the lower bound if $ub_0 \geq ub_1 + ub_2$ and the upper bound if $ub_0 \leq \min(ub_1, ub_2)$.

6.4 Partial solutions - a counterexample

> Divide and conquer algorithms divide a problem into subproblems that can be solved much easier and then reconstruct a solution from the solutions of the subproblems. Although such approaches are useful, as we will also see in this chapter, it seems in general impossible to obtain optimal solutions this way. For the straightforward subproblems where only a subset of the target reactions is considered, a counterexample is given.

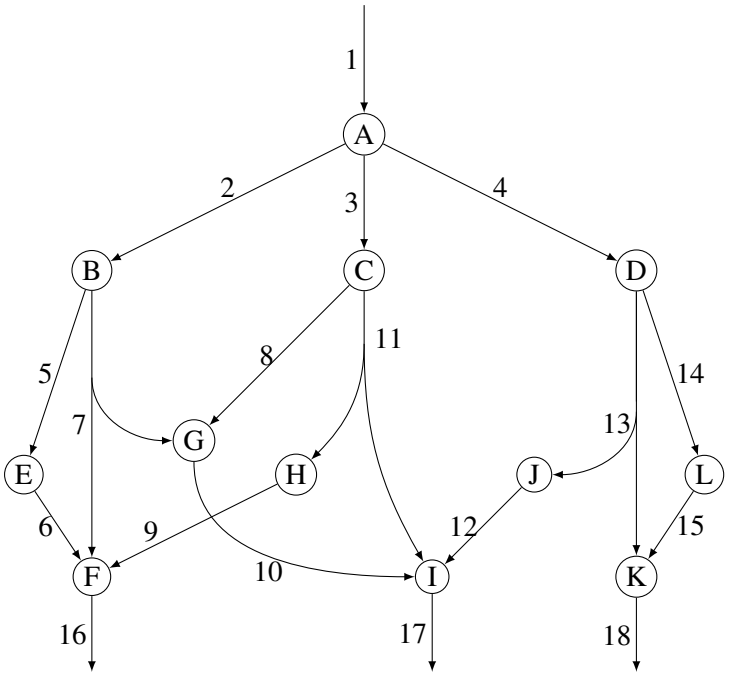
By considering only a subset of the set of the target reactions T in (OP3), p. 58, we get solutions that are optimal for producing this subset. In many cases, these optimal sub-solutions can even be combined to an optimal solution of (OP3).

Formally, for every subset $U \subset T$ we modify (OP3) to (OP3)^U by removing the output-constraints for $t \in T \setminus U$, so that only for $t \in U$ the constraints

$$\sum_{k=1}^{l} \tau_k v_t^k \geq \Gamma_t$$

remain. Since the number of integer constraints in the MIQCP problem (OP3) is then reduced from $|T| \cdot l$ to $|U| \cdot l$, these subproblems are easier to solve. Let $V_U = \{v^1, \ldots, v^i\}$ be a set of flux modes of an optimal solution of (OP3)^U. We can assume that $i \leq |U|$, see Prop. 3.10. The union $V_k = \bigcup_{U \subset T, |U| \leq k} V_U$ collects such flux modes of optimal sub-solutions for all $U \subset T$ with at most $|U| = k$ target reactions. Optimal solutions to all subproblems of this kind are collected in $V_{|T|-1}$

and intuitively we would expect this set to be a good pre-selection for (OP3), see Sec. 3.8. However, the following counterexample in Fig. 6.1 shows that an optimal solution of (OP3) cannot always be composed from solutions of subproblems $(\text{OP3})^U$, i.e., the pre-selection $V_{|T|-1}$ is not perfect in general.



reactions:
1: → A
2: A→ B
3: A→ C
4: A→ D
5: B→ E
6: E→ 5 F
7: B→ 4 F + 2 G
8: C→ 3 G
9: H→ F
10: G→ I
11: C→ 3 H + 3 I
12: J → I
13: D→ 2 J + 3 K
14: D→ L
15: L→ 3 K
16: F→
17: I →
18: K→

Figure 6.1: A network with 18 irreversible reactions. Reaction 1 takes up a substrate that serves for production of three possible outputs. The three output reactions are also the target reactions, i.e., $T = \{16, 17, 18\}$. The demanded output is $\Gamma_{16} = \Gamma_{17} = 10$ and $\Gamma_{18} = 7$. We assume that the resource allocation model is defined by (3.8) with the values $A_{tot} = 3$, $kc_1^+ = 1$, $kc_j^+ = 5$ for $j = 2, \ldots, 15$ and unbounded target reactions which are also excluded from resource allocation, i.e., $kc_t^+ = \infty$ and $\gamma_t = 0$ for $t = 16, 17, 18$. All reactions except the three target reactions are subject to resource allocation with $\gamma_j = 1$, $j = 1, \ldots, 15$. The flux space of this network is then defined as in (OP3) as $\mathcal{F}^* := \{v \in \mathbb{R}^{18} : Sv = 0,\ 0 \leq v \leq A_{tot} kc^+ \langle g, \gamma \rangle^{-1}$ with $g \in \{0,1\}^{18},\ g_j = 1 \Leftrightarrow v_j > 0, j = 1, \ldots, 18\}$, without any further constraints on the gene expression states g. The stoichiometry is given on the right. This network example is not realistic because mass is not conserved (e.g. D can be converted to 3 K or alternatively to 3 K plus 2 J). However, it can be part of a larger metabolic network which is realistic, in particular mass conserving, with the same target reactions and additional metabolic output.

Regarding the results of Sec. 3.8, we note that the network model of Fig. 6.1 is a conic sprout (Def. 2.7) and that only the single uptake reaction 1 is not redundant. It follows by Prop. 3.21 and the observations on page 32 that an optimal solution of (OP3) can always be composed of EMs and the same holds for the subproblems $(\text{OP3})^U$. There are six EMs of the network, their support and maximal output production is shown in Fig. 6.2. We will show that this model has the following two properties:

- Every optimal solution of (OP3) contains the EM e^1, i.e., $\text{supp}(e^1) \subset \text{supp}(v^i)$ for one of

the flux modes v^i, $i \in \{1,2,3\}$, of the optimal solution.

- The same EM e^1 is not contained in any optimal solution of (OP3)^U with $U \subsetneqq T$.

As a direct consequence, the complete set $V_{|T|-1}$ of optimal solutions of proper subproblems is not a perfect pre-selection, no matter which optimal solutions are chosen for $V_{|T|-1}$.

The inner representation of the flux cone of $\mathcal{F}^*$ is given by $\{v \in \mathbb{R}^{18} : v = E\lambda,\ \lambda \geq 0\}$, where the matrix E contains the EMs as columns, i.e., $E_{\cdot,i} = e^i$, $i = 1,\ldots,6$. A flux mode v that does not contain the EM e^1 must then be given as $v = E'\lambda,\ \lambda \geq 0$, where E' is obtained from E by deleting the first column e^1. We define $\mathcal{F}' := \{v \in \mathcal{F}^* : v = E'\lambda,\ \lambda \geq 0\} \subsetneqq \mathcal{F}^*$. Solving (OP3) over $\mathcal{F}^*$ gives a strictly smaller objective value than over $\mathcal{F}'$ and we can conclude that every optimal solution must include e^1.

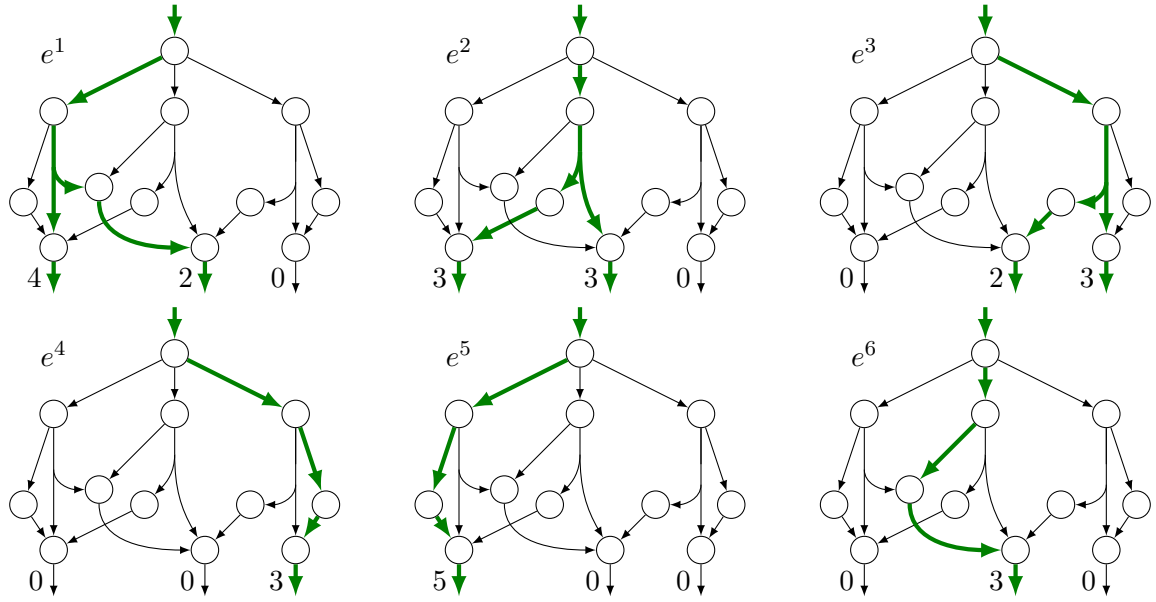


Figure 6.2: The six elementary flux modes of the network (computed with the efm-tool [Terzer and Stelling, 2008]). The whole flux space space can be given by convex combinations of these flux modes and zero. The flux rates of the target reactions are indicated for each maximal EM in the flux space $\mathcal{F}^*$ which is bounded according to our resource allocation model. All EMs have in fact the same enzyme cost, since they all use exactly four reactions subject to resource allocation (the target reactions in this example were excluded from resource allocation). The EM e^1 can be beneficially combined with e^3 to produce all three target metabolites. However, for the demand of any proper subset $U \subsetneqq T$, the use of e^1 is not efficient and for this reason this EM cannot be found by solving the subproblems (OP3)^U.

To see that the example network also fulfils the second property, we will look at all subsets of $T = \{16,17,18\}$. For the one-element subsets $U = \{16\},\{17\},\{18\}$ just note that e^5, e^6, e^3 would always replace e^1 because they have a higher flux rate through the sole target reaction 16,17 or 18, respectively. For the subset $U = \{16,18\}$, the EM e^1 only contributes to the output flux v_{16} and is hence replaced with e^5 again. Similarly e^6 substitutes e^1 in case $U = \{17,18\}$. It remains to check that e^1 cannot be contained in an optimal solution of (OP3)^U with $U =$

$\{16, 17\}$. Note that all EMs, generating the whole flux space, have output flux $e^i_{17} \leq 3$ and $e^1_{17} = 2$ only. Therefore, any flux mode v containing e^1 fulfils the strict inequality $v_{17} < 3$. Producing the demand $\Gamma_{17} = 10$ would therefore require strictly more time than $\frac{10}{3}$ whenever e^1 is part of the solution. But on the other hand, the time $\frac{10}{3}$ is sufficient to produce the whole demand of $U = \{16, 17\}$ with the flux mode e^2. We conclude that e^1 cannot be part of an optimal solution of (OP3)^U with $U = \{16, 17\}$ or with any $U \subsetneqq T$.

6.5 Pre-selections that approximate an optimal solution

Solving the optimisation problem (OP2a), p. 55, is not tractable for much larger networks than our core metabolic network of Chap. 4 and also computing the perfect pre-selection given in Sec. 3.8, based on EM enumeration, is only applicable in medium sized networks without maintenance constraints. This section presents the construction of a pre-selection that approximates an optimal solution. To guarantee a small error, the size of the pre-selection is required to be very large. In practice it is therefore not possible to formally assure a high quality of this approximation. Nevertheless, the pre-selection turns out to give good results in practice and this approach is hence promising.

In Sec. 3.7 we saw that all elements of $\mathcal{F}^*$ which are not decomposable in a specific sense, allow finding an optimal solution to (OP2a), p. 55, with $l = |T|$, just by solving the LP (LPps) which contains only $|T|$ constraints. In practice, the applicability of this approach is hindered on the one hand by the fact that EM enumeration is not tractable for large networks and on the other hand because it requires that the flux space is a conic sprout (Def. 2.7). As soon as we have maintenance constraints, as in the application in Chap. 4, this condition is not given anymore. The methods presented in this chapter are an alternative to deal with larger networks with or without maintenance reactions. In this section a pre-selection approach is given. Once a pre-selection is given, it suffices to solve the computationally trivial LP (LPps) in order to get a solution to (OP2a). The pre-selection is called perfect, if this solution is optimal. This is the case for the pre-selection that was defined by Prop. 3.19. However, its construction requires to enumerate all EMs or even the superset of non-decomposable elements. These enumeration problems have exponential complexity in the size of the network, since already the output of all EMs has exponential size in general [Fukuda and Prodon, 1996].

Here we will construct a pre-selection by solving (OP2a) with $l = 1$ for different demands Γ. The computational problem is hence reduced to $l \cdot n = n$ binary variables instead of $|T| \cdot n$. Since the individual steps of this method are independent, the space complexity is independent of the size of the resulting pre-selection, in contrast to EM enumeration with the double description method, where all EMs are computed simultaneously. Also if (OP2a) is solved with the Branch and Bound method an exponential number of intermediate candidate solutions is produced, see Sec. 6.2.

6.6 Formal construction of the pre-selection

The following proposition shows how to obtain a pre-selection that is approximating an optimal solution of (OP2a), p. 55. Approximating means that the optimal choice of flux modes from

the pre-selection has an objective value $\bar{\tau}^*$ that is close to the global optimum τ^* of (OP2a). By increasing the size of the pre-selection the difference $\bar{\tau}^* - \tau^*$ gets arbitrarily small. However, to guarantee a small difference in practice, the required size is so large that the construction is not tractable. Nevertheless the result is of practical use, because it tells us how the flux modes of the pre-selection should be chosen. Following these instructions, we can obtain good approximations, as will be illustrated with the core metabolic network of Chap. 4.

Proposition 6.1. Given an instance $(\mathscr{P})$ of the optimisation problem (OP2a), p. 55, with objective value τ^* and a tolerance $\theta > 1$, a pre-selection can be generated, such that an optimal choice, i.e., an optimal solution of (LPps) gives an objective value $\bar{\tau}^* \leq \theta\tau^*$.

Proof. A construction of the pre-selection $\mathscr{A}$ of flux modes $w \in \mathcal{F}$ will be given. In (OP2a) the demand for output through target reaction t is given by $\Gamma_t \geq 0$, $t \in T$. We will define a collection $\mathscr{D}$ of demands $D \in \mathbb{R}^T_{\geq 0}$ and for each $D \in \mathscr{D}$ the optimisation problem (OP2a) with $l = 1$ and demand D, i.e.,

$$\begin{aligned} &\min\ \delta \text{ s.t.} \\ &w \in \mathcal{F}^* \\ &\delta w_t \geq D_t \quad \text{for } t \in T \\ &\text{with variables } \delta \in \mathbb{R}_{\geq 0},\ w \in \mathbb{R}^n \end{aligned} \tag{6.2}$$

is solved. (We use δ here instead of τ for the duration to avoid confusion with the total duration τ of an optimal solution of the optimisation problem (OP2a) which we want to approximate.) If this problem is feasible, we get an optimal solution (w, δ) and w^* is added to the pre-selection $\mathscr{A}$. A given collection $\mathscr{D}$ thus generates a pre-selection $\mathscr{A}(\mathscr{D})$. This collection $\mathscr{D}$ will be constructed as a grid $\prod_{t\in T}\{x_t^0, x_t^1, \ldots, x_t^{k(t)}\} \subset \mathbb{R}^{|T|}_{\geq 0}$. To satisfy the tolerance θ with the pre-selection $\mathscr{A}(\mathscr{D})$ it suffices that $\mathscr{D}$ fulfils the following condition. For every flux mode v that is part of an optimal solution of (OP2a) there must be a $D \in \mathscr{D}$ such that an optimal solution w of (6.2) with D fulfils

$$\theta w_t \geq v_t, \quad t \in T. \tag{6.3}$$

A grid $\mathscr{D} = \prod_{t\in T}\{x_t^0, x_t^1, \ldots, x_t^{k(t)}\} \subset \mathbb{R}^{|T|}_{\geq 0}$ will now be constructed to fulfil this requirement. At first we set $x_t^0 = 0$ for all $t \in T$. We will now infer conditions on D which suffice to fulfil (6.3). By choosing the grid points suitably we can then assure that for any v from an optimal solution, these conditions are fulfilled by at least one $D \in \mathscr{D}$.

For a start we just require that $D_t = 0 \Leftrightarrow v_t = 0$ for $t \in T$. The components t with $v_t = 0$ are then fixed to $D_t = 0$. Only the components t where $v_t > 0$, or equivalently, $D_t > 0$, remain to be further specified. Their index set will be denoted $T_+ \subset T$. Let (w, δ) be an optimal solution of (6.2) with this D. For $t \in T_+$ we have $v_t, D_t > 0$, so we can write $\frac{v_t}{D_t}\delta w_t \geq v_t$ and hence

$$\max_{t\in T_+}\left(\frac{v_t}{D_t}\right)\delta w_t \geq v_t, \quad t \in T_+. \tag{6.4}$$

Since w is an optimal solution of (6.2), we can assume that there is a $t \in T_+$ with $w_t \geq v_t$, because otherwise v would give a better solution. As a consequence we have $\max_t\left(\frac{D_t}{v_t}\right) w_t \geq D_t$, $t \in T_+$, but δ is minimal for this inequality and therefore $\delta \leq \max_t\left(\frac{D_t}{v_t}\right)$. Substituting into (6.4) gives

$$\max_{t\in T_+}\left(\frac{v_t}{D_t}\right) \cdot \max_{t\in T_+}\left(\frac{D_t}{v_t}\right) w_t \geq v_t, \quad t \in T_+.$$

For $t \in T \setminus T_+$ we even have $w_t \geq v_t = 0$. Replacing in the optimal solution of (OP2a) v with w and the duration τ_v of v with $\tau_v \cdot \max_t \left(\frac{v_t}{D_t}\right) \cdot \max_t \left(\frac{D_t}{v_t}\right)$ gives thus a feasible solution of (OP2a). We conclude that the whole optimal solution can be approximated with tolerance θ if there is for every v from the optimal solution such a $D \in \mathscr{D}$ which fulfils

$$\theta \geq \max_{t \in T_+} \left(\frac{v_t}{D_t}\right) \cdot \max_{t \in T_+} \left(\frac{D_t}{v_t}\right) \tag{6.5}$$

Every v can then be substituted by an optimal solution w of (6.2) with some $D \in \mathscr{D}$ and the total duration increases then by at most θ, i.e., $\bar{\tau}^* \leq \theta\, \tau$.

Since $\left(\max_t \left(\frac{D_t}{v_t}, \frac{v_t}{D_t}\right)\right)^2 \geq \max_t \left(\frac{v_t}{D_t}\right) \cdot \max_t \left(\frac{D_t}{v_t}\right)$, a sufficient condition for (6.5) is given by $\frac{D_t}{v_t}, \frac{v_t}{D_t} \leq \sqrt{\theta}$ or, equivalently,

$$\frac{1}{\sqrt{\theta}} \leq \frac{D_t}{v_t} \leq \sqrt{\theta} \quad \text{for} \quad t \in T_+. \tag{6.6}$$

These conditions are stating that v and D are in close vicinity. The closeness is defined componentwise and by the ratio. In case $t \notin T_+ := \{t \in T : v_t > 0\}$, the components v_t, D_t are both zero and otherwise their ratio must be close to one. This means that the smaller the value $v_t > 0$, the closer must D_t be in absolute terms. As a consequence, in every component the required density of the grid is going to infinity as zero is approached. To avoid this explosion we have to define a threshold $\varepsilon > 0$ and assume $v_t \geq \varepsilon$, $t \in T_+$, for all flux modes v. This assumption is necessary to obtain a finite grid $\mathscr{D}$ satisfying the tolerance θ of the approximation. As explained below this proof, this is in fact not a relevant restriction.

The stoichiometric upper bounds sub_t (Def. 2.4) bound the flux rates v_t from above. Altogether we thus have $v_t \in \{0\} \cup [\varepsilon, sub_t]$. The interval $[\varepsilon, sub_t]$ will be partitioned into $I_m := [\varepsilon(\sqrt{\theta})^{m-1}, \varepsilon(\sqrt{\theta})^m]$, $m = 1, \ldots, M_t - 1$ and $I_{M_t} := [\varepsilon(\sqrt{\theta})^{M_t-1}, sub_t]$, where $M_t = \left\lceil \frac{\log(sub_t \varepsilon^{-1})}{\log(\sqrt{\theta})} \right\rceil$. If $v_t \in I_m$, then $\frac{1}{\sqrt{\theta}} v_t \leq \varepsilon(\sqrt{\theta})^{m-1}$ and $\sqrt{\theta} v_t \geq \varepsilon(\sqrt{\theta})^m$. Hence, (6.6) is implied by the inequality

$$\varepsilon(\sqrt{\theta})^{m-1} \leq D_t \leq \varepsilon(\sqrt{\theta})^m. \tag{6.7}$$

Therefore, it suffices to let D_t vary over the grid points

$$x_t^k = \varepsilon(\sqrt{\theta})^{2(k-1)}, \quad k = 1, \ldots, \lceil (M_t + 1)/2 \rceil \tag{6.8}$$

Then we have for every $m = 1, \ldots, M_t$ a grid point x_t^k, which coincides with one of the two bounds given in (6.7), namely

- $\varepsilon(\sqrt{\theta})^{m-1} = x_t^k$, in case m is even and we take $k = (m+2)/2$.
- $\varepsilon(\sqrt{\theta})^m = x_t^k$, in case m is odd and we take $k = (m+1)/2$.

For $v_t \in I_m$ we can then set $D_t = x_t^k$ to fulfil (6.7) for all $t \in T_+$. In case that v_t is not in any of the intervals I_m, we must have $v_t = 0$ and hence this component can be fitted exactly with $x_t^0 = 0$, which was already defined in the beginning. Altogether these grid points provide for an arbitrary v a $D \in \prod_{t \in T} \{x_t^0, x_t^1, \ldots, x_t^{k(t)}\} = \mathscr{D}$ which satisfies (6.7) and hence also (6.6). □

To obtain a finite collection $\mathscr{D}$, the proof of Prop. 6.1 required to exclude target fluxes $0 < v_t < \varepsilon$ of flux modes v from an optimal solution. This means that the optimisation is limited to $\mathcal{F}^\varepsilon := \mathcal{F}^* \cap \{v \in \mathbb{R}^n : v_t \notin (0, \varepsilon),\ t \in T\}$. This is in fact not a relevant restriction of the model, since flux rates below a certain threshold ε are biologically meaningless. The difficulty with flux rates close to zero is not just a characteristic of the construction of Prop. 6.1, but is also causing numerical troubles in solving the optimisation problems, as shortly noted at the end of Sec. 6.2, thus suggesting that this is an inherent property of the problem (OP2a).

The grid $\mathscr{D}$ is completely defined by (6.8). Constructing $\mathscr{D}$ and solving (6.2) for every $D \in \mathscr{D}$ yields the pre-selection $\mathscr{A}(\mathscr{D})$. In practice it is however not possible to realise this construction for a small tolerance $\theta > 1$. The size of the grid $\mathscr{D}$ that is sufficient to achieve an approximation with error $\leq \theta$ is given by $\prod_{t \in T}(1 + \lceil (M_t + 1)/2 \rceil)$ with $M_t = \left\lceil \frac{\log(sub_t \varepsilon^{-1})}{\log(\sqrt{\theta})} \right\rceil$, which gets very big for small $\theta > 1$. But the construction shows how the grid points should be distributed. The essential message is that the density of the grid points must be increased exponentially in each component $t \in T$ as we approach zero, instead of spanning for example an equidistant grid. Using a θ that is large enough to achieve a grid of tractable size, we obtain the desired distribution of grid points. This gives very good approximations when applied to the metabolic network from Chap. 4, as shown in Tab. 6.1 below. The construction of Prop. 6.1 gives hence also a practicable alternative to solving the optimisation problem exactly.

6.6.1 Improving a given solution

We will now present a modification of the optimisation problem (OP2a), p. 55, which allows to improve a given feasible solution containing several flux modes. This improvement can be applied to the approximated solution from the just presented pre-selection approach or to any feasible solution. Assume that the given solution contains d flux modes $x^1, \ldots, x^d$. To compute h new flux modes $v^1, \ldots, v^h$ which can be combined with the x^i, we solve the following optimisation problem extending (OP2a) with the continuous variables $\tau_1, \ldots, \tau_d$.

$$
\begin{aligned}
&\min \textstyle\sum_{k=1}^{l} \tau_k, \text{ subject to:} \\
&v^k \in \mathcal{F}^*, \quad k = 1, \ldots, l \\
&\textstyle\sum_{k=1}^{d} \tau_k x_t^k \ + \ \sum_{k=d+1}^{l} \tau_k v_t^k \geq \Gamma_t \quad \forall t \in T \\
&\text{with variables:} \\
&\tau_k \in \mathbb{R}_{\geq 0},\ v^k \in \mathbb{R}^n, \quad k = 1, \ldots, l = d + h
\end{aligned}
\tag{OP2x}
$$

The given flux modes $x^1, \ldots, x^d$ are fixed parameters. This extension adds only d continuous variables, but no integer variables and also no constraints. The tractability of the optimisation problem is therefore not affected (cf. p. 127). The computation of a pre-selection by Prop. 6.1 is based on solving (OP2a) with $l = 1$. In case this is tractable, also (OP2x) with $h = 1$ can thus be computed and used to improve the approximated solutions attained with the pre-selection.

6.6.2 Approximation in the core metabolic network

In the metabolic network introduced in Chap. 4 the exact optimal solutions could be computed and this model can hence be used to test the performance of the approximation approach. As mentioned above, it is not possible to choose a small tolerance θ, since this would immediately

lead to an intractable number of grid points $\mathscr{D}$. Some computational experiments with grids $\mathscr{D}$ of different sizes, generated according to Prop. 6.1 were carried out. The tolerance had to be larger than 5 to obtain tractable grid sizes. However, it turns out that the obtained results are very good, with errors below 1%, see Tab. 6.1. Also the improvement of the solution by employing (OP2x) turns out to bring a significant gain. Comparing the results for the three

					improvement		*2nd improvement*	
	$\lvert\mathscr{D}\rvert$	$\lvert\mathscr{A}\rvert$	time [h]	error [%]	time	error	time	error
glc, lac	80	76	4.887	2.191	4.824	0.869	4.795	0.267
	624	467	4.833	1.063	4.805	0.485	4.795	0.257
	2400	1304	4.818	0.757	4.791	0.182	4.785	0.060
glc	80	77	6.081	22.904	4.964	0.338	4.949	0.024
	624	546	5.882	18.892	5.524	11.648	4.982	0.692
	2400	1778	5.003	1.131	4.962	0.301	4.948	0.013
lac	80	67	5.852	0.003	5.852	0	5.852	0
	624	491	5.852	0.003	5.852	0	5.852	0
	2400	1115	5.852	0.003	5.852	0	5.852	0

Table 6.1: Results of computing an optimal solution by solving (LPps) with a preselection generated by the procedure given in Prop. 6.1 for our core metabolic model (compare with Tab. 4.3 where exact optimal solutions are given). Approximative pre-selections of different sizes were tested. The number of flux modes in $\mathscr{A}$ generated from the grid $\mathscr{D}$ is smaller than the number of grid points since several $D \in \mathscr{D}$ can give the same optimal flux mode. Furthermore, the improvement by (OP2x) with $h = 1$ was applied twice to each solution, computing one new flux mode each time. The errors for all solutions are given in percent of additional time needed for output production, w.r.t. the optimal solutions of Tab. 4.3.

different substrates shows that the performance of the approximation depends strongly on the optimisation problem. While the method even finds an exact optimal solution in the network with lactate as the only substrate, we have large errors with a small grid and glucose as the only substrate. Especially in this case, we can see that the additional use of (OP2x) has great benefits. Altogether, the results in Tab. 6.1 can be summarised by stating that increasing the size of the pre-selection as well as employing (OP2x) several times yields in all cases a solution with a duration that is only less than 1% increased, compared to an optimal solution. In the case where the large grid ($\lvert\mathscr{D}\rvert = 2400$) is used and two improving flux modes are computed one after the other, the error is even pushed below 0.1%. These results suggest that the two methods applied in concert can yield near optimal solutions also in larger networks, where the direct, exact solution of the optimisation problem (OP2x) is not tractable anymore. However, it has to be noted that this approach will be rather heuristic, because Prop. 6.1 is not applicable with a small tolerance in practice, due to the high number of required grid points $\mathscr{D}$. The computations here gave errors that were magnitudes smaller than formally guaranteed by Prop. 6.1. The application of the approach to other and larger networks relies on the hypothesis that a similar outcome can be expected in general.

6.7 Exhaustive enumeration of gene expression sequences

In the basic formulation, (OP3), p. 58, allows arbitrary gene expression patterns. However, to get a biologically meaningful model, the space G of possible gene expression patterns has to be limited. Also the sequence of different gene activations may be limited in order to capture certain aspects of the regulatory dynamics in the model. The more information about the gene regulation is included, the smaller becomes the space of feasible sequences of gene expressions. If it is sufficiently small, we can compute for every feasible sequence an associated optimal sequence of flux modes. This gives not only one optimal sequence, but a complete picture, in particular of the trade-off between the cost of switching and the time that is gained (see Fig. 6.4). The computation can be facilitated by sorting out those gene expression patterns, which are surely not used in an optimal solution. The approach is illustrated on the core metabolic network introduced in Chap. 4.

Assume a sequence $\vec{g} := (g^1, \dots g^l) \in Dyn_l$ of gene expression patterns is given. The allocation of enzymes is then fixed in each phase, i.e., for each flux rate $v^1, \dots, v^l$ in (OP3), p. 58, the bounds are fixed. The optimisation problem is thus just an LP. A brute force approach for solving (OP3) would be to compute an optimal sequence of flux modes on every feasible sequence $\vec{g} := (g^1, \dots g^l) \in Dyn_l$, that is, to solve the LPs

$$\begin{aligned} &\min \sum_{i=1}^{l} \tau_i, \text{ subject to:} \\ &v^i \in \mathcal{F}_{g^i}, \quad i = 1, \dots, l \\ &\textstyle\sum_{i=1}^{l} \tau_i \, v_t^i \geq \Gamma_t \quad \text{for } t \in T \\ &\text{with variables:} \\ &\tau_i \in \mathbb{R}_{\geq 0},\ v^i \in \mathbb{R}^n, \quad i = 1, \dots, l. \end{aligned} \tag{6.9}$$

Note that this remains an LP if maintenance constraints are added. For the definition of $\mathcal{F}_g$ see p. 46. For every $\vec{g} \in Dyn_l$ we get an objective value $\tau^*(\vec{g})$. Ordering $\vec{g}$ by their corresponding objective values gives a hierarchy in efficiency. A sequence $\vec{g}$ with minimal objective value $\tau^*(\vec{g})$ together with the corresponding optimal flux modes and durations computed in (6.9) is an optimal solution of (OP3). The availability of optimal solutions for all feasible sequences $\vec{g} \in Dyn_l$ gives furthermore the possibility to get more insight into the problem, for example by analysing how the objective value depends on parameters, as e.g. the usage of certain reactions or the number of switches of genes (Fig. 6.4). In particular, we can analyse the trade-off between the costs for switching and the gain in efficiency. This can be done for various switching cost functions. For all these additional examinations it is not necessary to repeat the computation of (6.9) for all $\vec{g} \in Dyn_l$.

Solving (6.9) for one given sequence is computationally trivial and it is therefore possible to compute the optimal solution for a large number of sequences $\vec{g}$. The size of Dyn_l depends on the size of G itself and on further conditions concerning the order in which the gene expression patterns $g \in G$ can occur. In Sec. 3.6.3 the concept of minimal flux modes (MFM) was

introduced. It reduces G to a biological meaningful set of gene expression patterns. The space of feasible sequences $Dyn_l \subset G^l$ was furthermore restricted by taking into account the slow degradation of enzymes, which lead to the monotonicity constraints $g^i \leq g^{i+1}$, see Sec. 3.6.3. All these restrictions have to be implemented in (OP3) by further constraints on the binary variables g_j^k. The representation can become very large and it might also be impossible to represent Dyn_l with linear or convex constraints. These restrictions on Dyn_l make the exhaustive enumeration of solutions for all $\vec{g} \in Dyn_l$ interesting, because on the one hand, a constraint-based representation of Dyn_l is not required anymore and on the other hand, Dyn_l might be small enough to compute an optimal solution for each $\vec{g} \in Dyn_l$.

6.7.1 Representing the space of sequences

The monotonicity constraint that was applied to the core metabolic network is an example of constraints on the sequence of flux modes. The constraints $g^i \leq g^{i+1}$ can be interpreted as defining a set of feasible transitions from one gene expression state to another. In general, defining such feasible transitions seems to be a natural structure that can capture many aspects of the gene regulatory network (GRN) dynamics. Defining a set Tr of all feasible transitions $(g^i, g^{i+1}) \in G \times G$ gives a directed graph (G, Tr) which can give a representation of Dyn_l.

Sequences defined by GRN dynamics A restricted space of sequences naturally emerges when a model is given where a specified GRN controls the expression of the metabolic genes g. Let $x \in G \times R$ be a state of the complete GRN, where G denotes as before the metabolic genes and R all other genes of the GRN. The state transition graph (STG, see Sec. 7.2 for a formal introduction) defines all transitions (x, x') that can take place in the model. A transition between two states $g, g' \in G$ would then be feasible if and only if it is realised by a corresponding transition in the whole GRN. Formally, define Tr to be the set of all tuples (g, g'), such that there exists a transition (x, x') in the STG of the whole GRN with $x|_G = g$ and $x'|_G = g'$.

Identifying all necessary sequences We will now describe how an enumeration of all feasible sequences can be obtained from the graph structure (G, Tr). Of course we want to avoid having redundant sequences Dyn_l. Repetitions $(g, g) \in Tr$ are for example redundant, since $\tau_k \in \mathbb{R}_{\geq 0}$ in (3.23) can represent any duration of the given gene expression state g. Furthermore, since also $\tau_k = 0$ is possible, feasible sequences of length $< l$ do not have to be listed if they are already a sub-sequence of another sequence in Dyn_l.

Evidently, repetitions occur if and only if the graph (G, Tr) has self-loops and we can hence avoid repetitions by discarding all self-loops. Regarding the sub-sequences we will assume now that after deleting the self-loops the graph contains no directed cycles of length $\leq l$. Every node can then occur at most once in a path. A node of (G, Tr) is called *terminal* if it has no outgoing edge, i.e., $\bar{g}$ is terminal if there is no $g \in G$ with $(\bar{g}, g) \in Tr$. If the graph (G, Tr) has no terminal nodes, every feasible sequence $g^1, \ldots, g^h$ with $h < l$ occurs also as a subsequence of some $\vec{g} \in Dyn_l$ with length l and at the same time every (sub-)sequence occurs only once. By adding self-loops to all terminal nodes we create this situation artificially. So let (G, Tr') be the graph obtained from (G, Tr) by deleting all self-loops of non-terminal nodes and adding self-loops to all terminal nodes. In the directed graph (G, Tr'), a path of length h is a sequence of h edges $(g^i, g^i + 1) \in Tr$, $i = 1, \ldots, h$, connecting $h + 1$ nodes $g^1, \ldots, g^{h+1}$. If all paths in

(G, Tr') of lengths l are enumerated, then it follows that every sequence of length $h = 1, \ldots, l$ is represented by one of these paths, or by a subsequence of one of these paths.

Graphs with directed cycles In case a directed cycle of length $\leq l$ is contained in the graph, nodes can be visited several times by a path of length l and it is necessary to allow these repetitions. For example, consider the "trefoil" graph, where one node a is connected to three other nodes x, y, z by edges in both directions. Apart from these six no further edges exist. To get from x via y to z the middle node a has to visited twice. In other words, the enumeration of the paths will include cycles of the graph, which leads to hard combinatorial problems, see e.g. [Johnson, 1975].

Enumerating all paths Under the assumption that no directed cycles of length $\leq l$ are contained in the graph (G, Tr), all required paths can be enumerated by performing from each node a depth-first search until depth l. Equivalently the search can just be started in an extra source node s that only has outgoing edges, one to each node $g \in G$ and search until depth $l + 1$. The pseudo-code is given below. It consists of one function, $EXPLORE()$, which is called recursively. We start in the source node s that was added to the graph. (All variables are global so that the function has no arguments.) The initialisation is $currentNode \leftarrow s$, $depth = 0$ and $path = (\text{none}, \ldots, \text{none}) \in (G \cup \{\text{none}\})^l$.

function EXPLORE( )
  if $depth = l$ **then**
    $pathList \leftarrow pathList \cup path$
  else
    $depth \leftarrow depth + 1$
    $parent \leftarrow currentNode$
    for all $child$ with $(parent, child) \in Tr'$ **do**
      $path(depth) \leftarrow child$
      $currentNode \leftarrow child$
      EXPLORE( )
    end for
    $currentNode \leftarrow parent$
    $depth \leftarrow depth - 1$
  end if
end function

The size of $pathList$, i.e., the number of paths that are enumerated can be computed independently of an enumeration as the sum of all entries of the l-th power $(Adj)^l \in \mathbb{N}^{|G|\times|G|}$, where $Adj \in \mathbb{N}^{|G|\times|G|}$ is the adjacency matrix of the graph (G, Tr'). The entry $Adj_{g,g'}$ is 1 if there is an edge from g to g' and the entry $((Adj)^l)_{g,g'}$ gives the number of path of length l starting in g and ending in g'.

6.7.2 Reducing the set of gene expression patterns for optimal solutions

Above we discussed how an enumeration of Dyn_l can be algorithmically obtained from a graph representation of the dynamics of the GRN. Of course many other ways to obtain enumerations of Dyn_l are possible depending on the representation of the gene expression state and its dynamics. In the very simple case where $Dyn_l = G^l$ for example, we just have to generate all

l-element subsets of G, since the order is irrelevant. In this case, it is furthermore possible to reduce the number of sequences $\vec{g}$ for which an optimal sequence of flux modes must be computed by (6.9) to find the most efficient solution. The idea is to find gene expression states $g \in G$ that can be omitted, because a different $g' \in G$ allows in all target reactions at least the same flux rate. Discarding such states we get a reduced gene expression space $\bar{G}$. If $Dyn_l = G^l$, the choice of $g \in G$ is free in every phase $k = 1, \ldots, l$ in (OP3) and therefore it suffices to optimise over $Dyn_l = \bar{G}^l$.

Definition 6.2. Let $g, g' \in G$ be gene expression states. We say that g is **dominated** by g', if for all $v \in \mathcal{F}_g$ there exists $v' \in \mathcal{F}_{g'}$, such that $v_t \leq v'_t$ for all target reactions $t \in T$.

To find an optimal solution by exhaustive enumeration, the gene expression space G can thus be reduced by discarding dominated elements. The problem to decide if g' dominates g is closely related to a polytope containment problem, i.e., the question if polytope P is contained in polytope P' [Eaves and Freund, 1982]. Domination of g by g' is assured as soon as

$$\pi_T\left(\mathcal{F}_g\right) \subset \pi_T\left(\mathcal{F}_{g'}\right),$$

where π_T is the projection to the target reactions T, i.e., $\pi_T(v) = v_T$, the restriction of the flux vector v to T. This containment of the two projected polytopes is a stronger property than the domination defined above, see Fig. 6.3. The following Lemma gives a different sufficient condition for domination. It has the advantage that it can easily be tested by solving one single LP and is hence well suited to be used for reduction of G and Dyn_l. Remember that $\mathcal{F}_g$ is the polytope $\{v \in \mathbb{R}^n : Sv = 0,\ lb(g) \leq v \leq ub(g)\}$, where $ub_j(g) = A_{tot} k c_j^+ \frac{g_j}{\langle g, \gamma \rangle}$, $j = 1, \ldots, n$.

Lemma 6.3. *If the LP* $\{\min\ \alpha\ :\ \alpha v_t \geq ub_t(g),\ t \in T$ *and* $v \in \mathcal{F}_{g'}\}$ *has an objective value* $\alpha^* \leq 1$*, then* g *is dominated by* g'*.*

Proof. Let $v^* \in \mathcal{F}_{g'}$ such that $\alpha^* v_t^* \geq ub(g)_t$ for all $t \in T$. Every flux mode $v \in \mathcal{F}_g$ fulfils $v_t \leq ub(g)_t \leq \alpha^* v_t^*$ for all $t \in T$. If $\alpha^* \leq 1$ then this gives exactly the definition of g being dominated by g' with $v' = v^*$. □

It is clear that Lemma 6.3 holds equally if $ub(g)$ is replaced by the corresponding stoichiometric bounds (Def. 2.4). In the practical application the stoichiometric bounds should be used because they are often much smaller for some reactions and this makes the sufficient condition stronger and more dominated states are detected.

Figure 6.3: **A)** Projections of the hypothetical flux spaces $\mathcal{F}_g$, $\mathcal{F}_{g'}$ and $\mathcal{F}_{g''}$ on two target reactions. The grey area is the projection of $\mathcal{F}_g$ and the projections of the flux spaces $\mathcal{F}_{g'}$ and $\mathcal{F}_{g''}$ are indicated by their boundaries (dashed lines). If there are only these two target reactions, we can conclude that the gene expression state g is dominated by the states g' and g'', since both flux space projections contain $\mathcal{F}_g$. However, the test given by Lemma 6.3 recognizes only the dominance of g' but not of g''. The test checks if the box around the projection of $\mathcal{F}_g$, indicated by the full line, is included in the other flux space. **B)** An illustration of the slight difference between $g^* \in G$ being dominated by $g^{**} \in G$ and the containment of the corresponding projected flux spaces. The blue and the grey region are projections of $\mathcal{F}_{g^*}$ resp. $\mathcal{F}_{g^{**}}$ on the only two target reactions. Although the containment $\pi_T(\mathcal{F}_{g^*}) \subset \pi_T(\mathcal{F}_{g^{**}})$ is not given, g^* is dominated by g^{**} by definition.

6.7.3 Application in the core metabolic network

In the core metabolic network, exhaustive enumeration is still tractable. For the computations in Sec. 4.4 no constraints on the sequences of gene expression are given, i.e., $Dyn_l = G^l$. The gene expression space G itself is generated by 20 MGS and contains 177 elements. In Tab. 4.3 the results of solving (OP3), p. 58, with $l = 1, 2, 3, 4$ are given. For the condition where both substrates are available (rows $(i) - (iv)$) also the exhaustive enumeration approach was applied. At first, the 177 gene expression states could be reduced to only 94 by discarding those that cannot fulfil one of the maintenance constraints or cannot produce any of the demanded output. Furthermore, we sort out elements $g \in G$ that were found to be dominated by another gene expression state by applying Lemma 6.3. Discarding another 33 elements this way, the gene expression space is reduced to 61 elements. To obtain a complete picture of the efficiency with $l = 1, 2, 3, 4$ consecutive flux modes, we solve (6.9) for all $\vec{g} \in Dyn_l$ for $l = 1, 2, 3, 4$. Altogether the number of LPs that must be solved is hence $\sum_{k=1}^{4} \binom{61}{k} \approx 5.6 \cdot 10^5$ (the number of $1, 2, 3$ or 4 element subsets of G). Since LPs can be solved very fast, this can be done in about 1 hour on a PC (2.5 GHz).

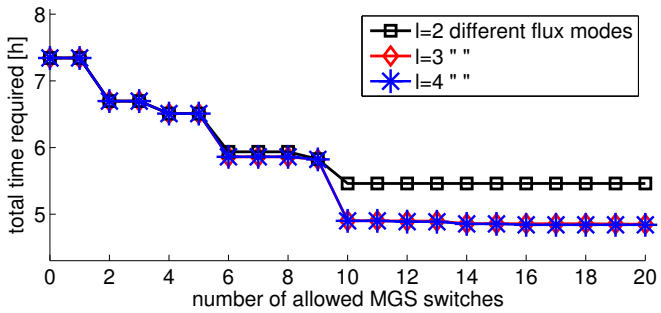


Figure 6.4: Durations of the optimal solutions for the core metabolic network. In Tab. 4.3 optimal solutions are given with $l = 1, 2, 3, 4$ and no constraints on the number of switches of MGS (rows (i) – (iv)). Here the number of MGS switched in the sequence of l flux modes is at most $n = 0, 1, \ldots, 20$. We can see that allowing only two switches between two consecutive flux modes already decreases the durations by about 30 [min]. Two similar large jumps occur when at most 6 and at most 10 switches of MGS are allowed, switching more than 10 MGS brings no significant improvement. While it makes almost no difference in this instance if three or four consecutive flux modes are allowed, the global optimal solution of less than 5 [h] cannot be achieved with only two consecutive flux modes. Instead of the number of flux modes, also a finer measure of the switching cost could be used, as e.g. the formulas derived in Sec. 4.4.2.

6.7.4 Conclusion

In the exhaustive enumeration approach, mostly the time complexity is critical for the tractability, because every LP is solved independently. Solving an individual LP is very fast and reliable. In case exhaustive enumeration can be applied, it has the advantage of giving very reliable results, compared to solving (OP3), p. 58, directly. The major advantage of the exhaustive enumeration is that the complete set of optimal solutions for all gene expression states allows analysing the model in various aspects. For example, an optimal solution on some arbitrary subspace of Dyn_l can be determined just by traversing the enumerated results. Also the trade-off between switching costs and gain in efficiency can be analysed w.r.t. to different measures of cost. Once the exhaustive enumeration of optimal solutions for all sequences $\vec{g} \in Dyn_l$ has been computed, the obtained data can be used for all these various purposes without further noteworthy computational efforts.

6.8 Discussion

The complexity of the general method, which consists of directly solving the optimisation problem (OP3), p. 58, is strongly limiting its applicability. Regarding further research based on our resource allocation model and on sequences of flux modes, it is therefore crucial to devise computational procedures with better tractability.

Reducing the search space - More computational capacity, more specific biological models

The alternative computational methods presented in this chapter are essentially restricting the search space of optimisation. The method of pre-selection, Sec. 6.5, is restricting the flux space to a finite set of flux modes. While the goal here was to span a grid over the flux space, such that

every optimal solution can be approximated with flux modes from this finite grid, a pre-selection could also be constructed by other strategies based on biological data or on computational methods. In the approach of exhaustive enumeration the flux space $\mathcal{F}^*$ is restricted to the polytopes $\mathcal{F}_{g^k}$ corresponding to the gene expression state of a fixed $\vec{g} = (g^1, \ldots, g^l) = Dyn_l$. The flux distributions are not preselected in this case. Although (OP3) reduces to an LP over the polytopes $\mathcal{F}_{g^k}$, the exhaustive enumeration is in general not a simplification of the problem, because the size of Dyn_l is limiting the tractability now and a small size of Dyn_l is also reducing the computational effort in solving the optimisation problem (OP3) directly, where the search tree of Branch and Bound is reduced. The exhaustive enumeration can be rather seen as an alternative way of computing an optimal solution which has the advantage that the LPs are solved separately and independently. The procedure is thus very reliable from a computational point of view. Furthermore, a large amount of relevant data is obtained, giving a complete picture of the efficiency in dependence on alterations in the metabolic network. It is a promising approach for models where the dynamics of the genes is restricted, so that Dyn_l is relatively small. A complete GRN model behind the metabolic genes for example is likely to reduce the feasible gene expression sequences dramatically.

Decomposition of optimal single flux modes

Not discussed in this chapter, but a near-by alternative to obtain solutions to (OP3) is the decomposition of a single flux mode which fulfils the requirements, i.e., an optimal solution of (OP3) with $l = 1$. In Chap. 3 we saw that optimal solutions with arbitrary l are characterised by non-decomposability of the flux modes. In the other direction, it is not true that decomposing an optimal solution of (OP3) with $l = 1$ gives an optimal solution for arbitrary l. The network of Sec. 6.4 serves as a counterexample again, because an optimal solution with $l = 1$ does not contain the EM e^1, while this EM is necessary for an optimal solution with $l \geq 3$. In practice, decomposition of single optimal flux modes yields nevertheless close to optimal solutions when applied to the core metabolic network of Chap. 4. However, this approach is not considered as an alternative here, because first a theoretical estimation of the approximation could not be established and second, decomposing a flux mode completely is a difficult task. The approach of [Chan and Ji, 2011] to decompose a flux mode into two flux modes with strictly smaller support as well as own variants based on optimisation were tested. The performance as well as the reliability were in all cases not satisfying.

7 Logical network models of gene regulation

7.1 Overview

Logical networks are discrete models of systems of distinct components that interact based on logical rules which can be represented by Boolean expressions, also in the case of multi-valued logic. In biology this formalism is used in particular for networks of gene regulation and signalling. It offers an intuitive way to encode knowledge about the biochemical interactions and can serve very well as an interface to a user who is not acquainted with the mathematical formalism and computational methods. Since Boolean expressions do not provide a unique representation of logical functions, it is desirable to find a minimal one. This can be achieved by the Quine-McCluskey algorithm and also by more efficient techniques. Here we will consider multi-valued logical networks which are encoded by Boolean expressions. The choice of an appropriate encoding leads to a modified version of the Quine-McCluskey algorithm to find a short and easily readable representation of the logical rules. After this procedure is presented, the formal derivation of the network dynamics from the logical rules is given and the application of the method is illustrated.

7.2 Transition systems and logical networks

7.2.1 Transition systems

> Logical networks can be seen as an instance of finite transition systems, which is a wider class of discrete models. The dynamics of such a system can be completely described by a directed graph, the so-called state transition graph. However, in practice the representation is implicit. In the case of logical networks, the interactions between the components are described. With additional specification of the dynamics, the state transition graph can then be derived.

Transition systems is a concept from computer science that is used in various forms to model real world systems, for example baggage transfer at airports, critical electronic circuits, as well as computer programs [Baier and Katoen, 2008]. For such man made systems the functionality is completely known (in contrast to biological systems), but due to the complexity we cannot explicitly identify all possible dynamic evolutions. Transition systems describe the dynamic behaviour with a discrete set of states and binary relations, called transitions, between these

states. All possible states and transitions constitute the so-called **state transition graph** (STG), which is basically a directed graph if the transition system is finite [Baier and Katoen, 2008]. The transition system is completely defined by the STG. However, in practice the size of the STG is often too large to be computed or stored. The description of the transition system is usually given in an implicit way.

Model checking The term model checking is used for a number of different methods of formal verification of concurrent systems [Baier and Katoen, 2008,Clarke et al., 1999]. Most prominent are those based on propositional temporal logic, such as Linear Temporal Logic (LTL) and Computation Tree Logic (CTL). These logics can describe the temporal ordering of events represented by the states of a transition system [Baier and Katoen, 2008]. The states of a transition system are labelled by so-called atomic propositions, in the case of modelling of gene regulatory networks (GRN) or other biological networks, these atomic propositions are the levels of activity of the different components (representing expression levels of genes or concentrations of regulatory or metabolic species). Based on these atomic propositions, the temporal logics can be used to specify properties of the possible evolutions from the given initial state [Baier and Katoen, 2008].

Petri nets Petri nets can be seen as a graphic representation of a restricted class of transition systems [Murata, 2002, Baier and Katoen, 2008]. Here we focus on finite transition systems. A corresponding finite Petri net can then be represented by an STG in the strict sense. Finite Petri nets have not only been used to model metabolic networks [Zevedei-Oancea and Schuster, 2003, Genrich et al., 2001, Schuster et al., 2002b], but also for regulatory networks [Steggles et al., 2007, Chaouiya et al., 2011]. The integration of a metabolic and a regulatory network in the Petri net formalism was exemplified in [Siñao et al., 2005] and an automated procedure was presented in [Palinkas and Bockmayr, 2011]. The studies [Chaouiya et al., 2011, Siñao et al., 2005, Palinkas and Bockmayr, 2011] are all based on a logical network model of gene regulation, as will be discussed in this chapter.

7.2.2 Logical networks

Biological knowledge about gene regulation or signalling can often be stated by logical conditions. For example, in *Escherichia coli* the trp operon (trpOp) is attenuated by tRNA and repressed by trpR [Yanofsky, 2000] and so we can state that ”trpOp is activated if not tRNA and not trpR”. Therefore it is common that models of regulation contain some logical components. For example [McAdams and Shapiro, 1995] proposes a hybrid approach where the biochemical network is partly modelled as a logic circuit. Logical networks as models for gene regulation were introduced in [Glass and Kauffman, 1973] and [Thomas, 1973], providing a theoretical foundation for modelling gene regulation as discrete events which are governed by logical rules. A specific mathematical framework of logical networks will be considered here. A general and concise introduction to this formalism can be found in [Richard, 2010]. Logical networks are an area of ongoing research and are often used to model GRN and also signalling networks. Apart from this framework, many other methods use a logical description of gene regulation, see e.g. [Karlebach and Shamir, 2008, McAdams and Arkin, 1998, McAdams and Shapiro, 1995]. Also integrated models of metabolism and gene regulation rely on logical descriptions [Covert et al., 2001, Siñao et al., 2005]. A toolbox that integrates logical rules for

transcriptional regulation into a constraint-based model of metabolism is presented in [Jensen et al., 2011].

Formal definition To describe a logical network with n components, we denote each component by an integer $i \in \{1, \dots, n\}$. The component i is assigned a maximal value $k(i)$ and it can take integer values from $\mathbb{M}_{k(i)} := \{0, 1, \dots, k(i)\}$. In the case that $k(i) = 1$ for all $i = 1, \dots, n$ we have a Boolean (binary) network, where every component can only take the values 0 or 1. In case $k(i) > 1$, the component i is called multi-valued (MV) and in a GRN we can then distinguish different expression levels, where $k(i)$ is the maximal activity. A logical network with at least one MV component is called MV network. The state space is then given by $\prod_{i=1}^{n} \mathbb{M}_{k(i)}$. The regulation of each component $l \in \{1, \dots, n\}$ is given by a function $\phi^l \colon \prod_{i=1}^{n} \mathbb{M}_{k(i)} \to \mathbb{M}_{k(l)}$ which will be called **target function** here. The target value $\phi^l(x)$ tells us where the component l is tending to in the state x.

To define possible dynamics, we must specify if one or several components can change their value simultaneously. In the case of MV components we must also specify if the value can change by steps > 1 or not. Here we will consider only updates that are asynchronous and unitary. **Asynchronous** means that only one component changes its value in one transition and **unitary** means that this value changes only by ± 1. For an overview of different updates of logical networks and the resulting dynamics see [Richard, 2010, Gershenson, 2003].

From target functions to update functions Once the update is fixed, to asynchronous and unitary in our case, two so-called **update functions** will be derived for each component. They tell us whether a component tends to be up or down regulated in a given state $x \in \prod_{i=1}^{n} \mathbb{M}_{k(i)}$ or whether it is stable. For a component $l \in \{1, \dots, n\}$ we have the update functions $\Psi^l_+, \Psi^l_- \colon \prod_{i=1}^{n} \mathbb{M}_{k_i} \to \mathbb{B}$, where Ψ^l_+ evaluates to 1 in exactly those states, where the actual value of l is smaller than the target value. In this case l is up regulated. Ψ^l_- is defined similarly for down-regulation, if the actual value is smaller than the target value. If both update functions give 0, this means that l is stable in this state. The case that both update functions evaluate to 1 is excluded by the definition. A detailed description of the update functions will be given in Sec. 7.5.

7.3 Representation of binary functions with minimal DNF

The research area of *logical synthesis* deals with representation, manipulation and optimisation of logical functions. It was initiated 1938 by Claude Shannon who noted the connection between Boolean functions and electrical circuits [Shannon, 1938]. Minimisation is one of the subjects of logical synthesis. A technique to find a certain minimal Boolean expression for a given function will be formally introduced, before the approach is adapted to the multi-valued case in the next section.

The representation of logical networks and their dynamics by logical functions was briefly introduced above. Binary logical functions $f \colon \mathbb{B}^n \to \mathbb{B}$ can be given by Boolean expressions, introduced on p. 16. In the following we will see how this formalism can be used to represent also MV logical functions. This representation is very well suited for the application to logical

network models of biological systems. On the one hand, the biological data and knowledge can be easily encoded without any acquaintance of the formalism. On the other hand, the representation is suitable for further computational processing such as simulation or model checking. In fact, Boolean expressions are frequently used to represent biological models of gene regulation and signalling by logical target functions, see e.g. [Chaouiya et al., 2011, Saez-Rodriguez et al., 2007, Zhang et al., 2008, MacLean and Studholme, 2010, Fauré et al., 2006].

Other representations of logical functions Many examples for Boolean or MV logical networks modelling gene regulation or signal transduction can be found on the GINsim webpage (`http://www.ginsim.org/`). The logical networks on the GINsim web page are often given by so-called logical parameters. This is another way of representing the truth-table of the logical functions that govern the behaviour of the components of the network [Thieffry and Thomas, 1995, Chaouiya et al., 2004]. This formulation is adjusted to the specific mathematical treatment of the networks, but as a representation it is not well suited, since even simple logical rules are not easily understandable from the corresponding logical parameters. A canonical representation of binary logical functions are the so-called Binary Decision Diagrams (BDD) [Crama and Hammer, 2011, Brayton and Khatri, 1999]. For MV logical functions a corresponding extension, called Multi-valued Decision Diagram (MDD), can be used [Brayton et al., 1990]. Both alternative representations of logical functions do not provide a self-explaining description as it is provided by Boolean expressions and which can serve as an interface to a user who is not acquainted with the formal background.

7.3.1 Boolean functions in sum-of-product expressions

Boolean expressions (cf. p. 16) give a representation of logical functions which is not at all unique. In fact an arbitrary number of equivalent expressions can be given for any logical function. A formal description of Boolean expressions and the relationship to logical functions is given next. In particular, this is necessary for the algorithm presented in Sec. 7.4.

Definition 7.1. A **literal** is a statement about the value of a Boolean variable x. We write the literal x, to state that $x \equiv 1$ and the literal $\overline{x}$, the negation, to state that $x \equiv 0$.

The binary function $f\colon \mathbb{B}^n \to \mathbb{B}$ that is defined by an expression over n variables can only take two values. Hence it is already defined by the **true points** $T := \{t_1, \ldots, t_r\} \subseteq \mathbb{B}^n$ of f, containing all states that are mapped to 1 [Crama and Hammer, 2011]. All other states $\mathbb{B}^n \setminus T$ are mapped to 0. When a biological system is modelled, the Boolean variables usually refer to the state of individual components, for example they can specify the expression level of a gene or the abundance of a signalling molecule or a metabolite. Since any Boolean expression can be written just by using the operators `and`, `or`, `not`, this representation of the logical function is easily understandable, even without any knowledge of the formal background.

Minimisation As already mentioned and illustrated in Fig. 7.1, the representation of a logical function by a Boolean expression is not unique. Operations on Boolean expressions lead usually to very large expressions which often have a much shorter equivalent form. An example is the derivation of the update functions for up- and down-regulation of a component from the target functions (see Sec. 7.5). For further processing of the model as well for the user who wants to read and understand the resulting Boolean expressions, it is hence desirable to find a minimal

$f(a,b,c)$	a	b	c	$\in T$			
0	0	0	0				
1	0	0	1	✓	→	$t_1 =$	$\overline{a}\cdot\overline{b}\cdot c$
1	0	1	0	✓	→	$t_2 =$	$\overline{a}\cdot b\cdot\overline{c}$
0	0	1	1				
1	1	0	0	✓	→	$t_3 =$	$a\cdot\overline{b}\cdot\overline{c}$
1	1	0	1	✓	→	$t_4 =$	$a\cdot\overline{b}\cdot c$
1	1	1	0	✓	→	$t_5 =$	$a\cdot b\cdot\overline{c}$
1	1	1	1	✓	→	$t_6 =$	$a\cdot b\cdot c$

$\rightsquigarrow$ $f(a,b,c) \equiv \overline{a}\cdot\overline{b}\cdot c + \overline{a}\cdot b\cdot\overline{c} + a\cdot\overline{b}\cdot\overline{c} + a\cdot\overline{b}\cdot c + a\cdot b\cdot\overline{c} + a\cdot b\cdot c$

Figure 7.1: Truth table for a Boolean function $f\colon\mathbb{B}^3 \to \mathbb{B}$ and the directly derived DNF. Every true point $t \in T$ can be written as a conjunction of literals such that every variable is fixed. A straightforward way to obtain a DNF for f is to take the disjunction of all these conjunctions, yielding the DNF $t_1 + \cdots + t_r \equiv f$. This Boolean expressions can often be drastically reduced. In this example, an equivalent expression is $f(a,b,c) \equiv a + \overline{b}\cdot c + b\cdot\overline{c}$. In verbal form it can be stated as "f gives 1 if a `or` (`not` b `and` c) `or` (b `and not` c). Otherwise f gives 0".

Boolean expression for a given logical function. Minimisation is one of various procedures on logical functions that are considered in the field of logical synthesis [Brayton et al., 1990]. To clarify what we mean here by minimality, the composition of a Boolean expression will be defined more precisely.

Definition 7.2. A conjunction (product) of literals is called **minterm**. A **disjunctive normal form (DNF)** is a disjunction (sum) of minterms.

As we saw in Fig. 7.1, a DNF can be directly derived from the true points of a function. However, the example also illustrated that often an equivalent DNF of smaller size exists.

Definition 7.3. A Boolean expression in DNF will be called **minimal** if any equivalent expression in DNF has at least the same number of minterms and if it is the same number, then it has at least the same number of literals.

It is convenient to represent minterms by vectors of a fixed size. To do so, we have to assume that the minterm contains at most one literal of each variable, which is a reasonable assumption, since otherwise there would be a tautology or a contradiction.

Definition 7.4. A minterm m can be represented by a **vector** $\boldsymbol{\alpha} \in \{0, 1, \star\}^n$, where $\alpha_i = 1$ if m contains the positive literal x_i of the i-th variable, $\alpha_i = 0$ if m contains the negative literal $\overline{x_i}$ or $\alpha_i = \star$ if m contains no literal of x_i.

The expression $a + \overline{b}\cdot c + b\cdot\overline{c}$ from Fig. 7.1 for example can be given in vector notation as $(1\star\star) + (\star 01) + (\star 10)$.

Definition 7.5. On binary functions with domain $\mathbb{B}^n$ we introduce the **partial order** $\leq$ by defining $g \leq f$ if $\forall x \in \mathbb{B}^n : g(x) = 1 \Rightarrow f(x) = 1$.

Since every Boolean expression in n variables defines a binary function on $\mathbb{B}^n$, this gives also a partial order on the set of all expressions in n variables. Equivalent expression are equal in this order.

Definition 7.6. Given a function f, an **implicant** is a minterm m, such that $m \leq f$. The implicant p is a **prime implicant** of f if there is no implicant $m \leq f$ with $p < m$ (which means $p \leq m$ and not $p = m$).

In other words prime implicants are maximal elements (w.r.t. $\leq$) in the set of all implicants of f. In a DNF of the function f all minterms are of course implicants of f. Furthermore, their sum gives exactly f.

Definition 7.7. A **cover** of f is a set of implicants $m_1, \ldots, m_k$, such that the disjunction $m_1 + \ldots + m_k \equiv f$. If all m_i are prime implicants it is called a **prime cover**. It is called **minimal** prime cover, if there is no prime cover of f with less than k prime implicants.

A DNF is hence a cover of f and we will see that a minimal DNF is a prime cover.

7.3.2 Finding minimal representations

State of the art in practical minimisation A logical function in n variables can have a minimal DNF with an exponential number of minterms in the worst case (an example is the function that evaluates to 1 if and only if the sum of all variables that are `true` equals $\lceil \frac{n}{2} \rceil$). The standard procedure for Boolean minimisation is the Quine-McCluskey algorithm [Quine, 1952, Quine, 1955, McCluskey, 1956]. But efficient minimisation of large expressions requires more sophisticated algorithms and often heuristic approaches are used. A collection of such algorithms for minimisation was developed under the name ESPRESSO [Rudell and Sangiovanni-Vincentelli, 1987], providing exact as well as approximative minimisation. These methods were also extended to handle MV logic [Rudell and Sangiovanni-Vincentelli, 1987]. An implementation of the ESPRESSO algorithms is also available [Lavagno et al., 1990]. The ESPRESSO algorithms essentially represent the state of the art in efficient two-level minimisation. Some methods were developed that are similar to ESPRESSO, but also contain different approaches, [Hlavička and Fišer, 2001, Sapra et al., 2003]. For functions with certain properties they were found to further improve the performance.

To work with binary logical networks that are used for biological modelling the software tool Bool-net [Müssel et al., 2010] was developed. It also features the computation of minimal expressions to represent a given Boolean network. In case the variables are Boolean and their number is small (≤ 5), the standard Quine-McCluskey algorithm can be used for minimisation. This is usually the case for models of gene regulatory or signalling networks. In the following section we show how the Quine-McCluskey algorithm can be modified to deal also with MV logical functions in order to find a minimal and well readable representation. The presented algorithm is also part of a tool [Palinkas and Bockmayr, 2011] for translating integrated models, consisting of a logical network for the gene regulation and a metabolic network, into Petri nets and into continuous time Markov chains (CTMC), see p. 17, in a format that serves as input to the probabilistic model checker PRISM [Hinton et al., 2006], see p. 165.

Comment on multi-level minimisation Another name for DNF is sum-of-products. Equivalently there are product-of-sum expressions also known as conjunctive normal form (CNF). Both are two-level expressions. An example for a three-level expression is $A \cdot (B + C) + X \cdot Y$. This sum of products of sums is not a two-level expression, but has to be expanded to $A \cdot B + A \cdot C + X \cdot Y$ to become a DNF. The three-level expression is in this example shorter than a minimal DNF, which shows that minimal DNFs can be further reduced, if expressions with more than two

levels are admitted. An algorithm to achieve Boolean minimisation for an arbitrary number of levels is given in [Lawler, 1964]. The goal here is to find short and easily understandable expressions. It is clear that expressions with too many levels will not be easily understood, because the structure is too complex. On the other hand, allowing more than two levels can reduce the size. The example $A \cdot (B + C) + X \cdot Y \equiv A \cdot B + A \cdot C + X \cdot Y$ was given above. The three level expression might be preferably here. However, a large impact compared to a minimal DNF cannot be expected in general, so that this direction is not further pursued here. Two levels are clearly necessary at minimum to express certain Boolean functions. In the other direction, Fig. 7.1 shows how a DNF can be directly derived from the true points of a function and therefore two levels also suffice. The methods mentioned in the previous paragraph refer to two-level minimisation (however, EXPRESSO provides also methods for multi-level minimisation [Bartlett et al., 1988]).

7.3.3 The Quine-McCluskey algorithm

The method proposed here for minimising expressions of MV logical functions uses a modification of the standard algorithm for Boolean minimisation from Quine and McCluskey [Quine, 1952, Quine, 1955, McCluskey, 1956] which will be abbreviated QMC. Let $f: \mathbb{B}^n \to \mathbb{B}$ be a Boolean function with true points $T = \{t_1, \ldots, t_r\}$. The covering $t_1 + \cdots + t_r$ gives already a DNF that is equivalent to f. The purpose of QMC is to find an equivalent DNF which is minimal according to Def. 7.3. To fulfil the first condition of this notion of minimality, the number of implicants in the covering must be minimal. Note that every covering can be replaced by a prime covering of at most the same size, because each implicant m in the cover is either a prime implicant or there is a prime implicant p such that $m \leq p \leq f$ and hence p can replace m and the cover will still be equivalent to f. The second condition of minimality, that the number of literals in the implicants of the covering is minimal, is surely satisfied by the prime implicants, because a minterm can only be increased in the partial order $\leq$ if a literal is discarded. Therefore, to find a minimal DNF for f we have to find a minimal prime cover of f according to Def. 7.7. This is done in two disjoint steps:

- Finding all prime implicants
- Choosing a minimal number of prime implicants that cover f

Here we will only be concerned with the first step. Once the prime implicants are determined, the second step is a general combinatorial problem (set covering problem) and it makes no difference whether binary or MV logic or a binary encoding of MV logic is considered.

Computation of all prime implicants

All prime implicants of a Boolean function f can be generated from the set of all true points $T = \{t_1, \ldots, t_r\}$ [Crama and Hammer, 2011, Sec. 3.2.1]. Once the set of all implicants is available, it is ordered by $\leq$ of Def. 7.5 and the prime implicants are then identified as the maximal elements. Another method, which can be applied to any DNF of f is called *consensus procedure* as described in [Crama and Hammer, 2011, Sec. 3.2.2]. A variant of this approach, which is adapted for the practical implementation, will be described now. In this variant, we assume that the initial DNF is the disjunction of the true points. Pairs t_i, t_j of true points are considered where the number of positive literals in t_i is by one larger than in t_j. The vectors

of the true points hence differ in at least one entry and if they differ in exactly one entry they are merged to one new implicant of f. This method, called *binary-merge* here, consists of iteratively testing pairs and merging them eventually to a new implicant.

Generating all implicants with Binary-Merge The merging procedure will be carried out by applying the following rule to pairs of implicants, starting with pairs of true points. All implicants are given in vector notation (Def. 7.4).

`Binary-Merge` rule:

> `If:`
> • α and α' are identical except in the j-th variable, where $\alpha_j = 1$ and $\alpha'_j = 0$.
> `Then:`
> • Merge to α'' with $\alpha''_j := *$ and $\alpha''_i := \alpha_i$ $(= \alpha'_i)$ for $i \neq j$.

This rule is applied to all pairs of true points. After this, all implicants that were merged at least once, are discarded. Since $\alpha'' \equiv \alpha + \alpha'$ (where $\equiv$ is used for equality of Boolean values), we can replace the two implicants α, α' with α'' and the cover will still be equivalent to f. In a next iteration, the rule is applied to all pairs from the set of the new implicants that were obtained by merging in the previous iteration, see the example below. (In every iteration, only such pairs have to be tested where the number of positive literals differs by one. Note that this order is inherited from the previous set of implicants and hence only the initial set of true points has to be ordered.) In the end, when no merges are possible anymore, the prime implicants are given by the set of all implicants that were never merged in the whole procedure.

This algorithm is a variant of the *consensus procedure* as mentioned above. A proof that the *consensus procedure* yields all prime implicants is given in [Crama and Hammer, 2011, Sec. 3.2.2]. Another proof, tailored for the variant used here, will be given below, it will also serve to prove the generalised procedure in the next section.

Example Consider $T = \{a \cdot \overline{b} \cdot \overline{c},\ a \cdot \overline{b} \cdot c,\ a \cdot b \cdot c,\ a \cdot b \cdot \overline{c}\}$. The first iteration merges four pairs of true points:

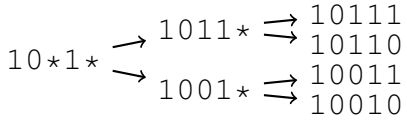
$$\left.\begin{matrix} a \cdot \overline{b} \cdot \overline{c}\ (100) \\ a \cdot \overline{b} \cdot c\ (101) \end{matrix}\right\} a \cdot \overline{b}\ (10*), \quad \left.\begin{matrix} (100) \\ (110) \end{matrix}\right\} (1*0), \quad \left.\begin{matrix} (111) \\ (110) \end{matrix}\right\} (11*), \quad \left.\begin{matrix} (101) \\ (111) \end{matrix}\right\} (1*1)$$

Since all initial implicants (the true points) were merged at least once, they are all discarded and only the new implicants are potential prime implicants. (If one implicant would not be merged with any partner, it would be a prime implicant). The second iteration merges two pairs that fulfil the merging condition. Both result in the same single prime implicant a, which is hence equivalent to the function given by T.

$$\left.\begin{matrix} a \cdot \overline{b}\ (10*) \\ a \cdot b\ (11*) \end{matrix}\right\} a\ (1**) \begin{matrix} a \cdot \overline{c}\ (1*0) \\ a \cdot c\ (1*1) \end{matrix}$$

Proposition 7.8. All implicants of f are generated in the merging procedure `Binary-Merge`. The implicants that were never merged are exactly the prime implicants of f.

Proof. First note that a prime implicant cannot be merged with any other implicant, since merging results in a strictly greater implicant in the partial ordering $\leq$. We will show now that every implicant of f is generated during `Binary-Merge`. This implies that also all prime implicants are generated. Given α'' (the vector of an implicant of f) with at least one $*$-entry $\alpha''_j = *$ we can split α'' up. This means that we invert the merging that produced this $*$-entry. Splitting α'' at the entry $\alpha''_i = *$ produces the implicants α, α' with $\alpha_j = 1$ and $\alpha'_j = 0$. The resulting implicants are strictly smaller than α'', i.e., $\alpha < \alpha''$ and $\alpha' < \alpha''$. We proceed by splitting α and α' at one of the remaining $*$-entries and iterate this. Finally this gives implicant-vectors that contain no $*$ at all, which means that they describe one single state where f evaluates to 1, i.e., a true point $t_i \in T$.



The splitting is the inverse operation of the merging procedure described above, so we can conclude that the implicant α'' is generated in this procedure. Note that in the iterative application of the splitting of an implicant α, all implicants $\omega < \alpha$ occur and all these implicants are merged at one point in `Binary-Merge`. This implies that an implicant ω that was never merged in `Binary-Merge` must be a prime implicant. □

In the example of Fig. 7.1 the prime implicants are a, $\overline{b} \cdot c$ and $b \cdot \overline{c}$. This is already a minimal covering set and $f(a,b,c) \equiv a + \overline{b} \cdot c + b \cdot \overline{c}$ is hence a minimal DNF.

7.4 Generalisation to multi-valued functions

To handle multi-valued (MV) logical functions, we will encode the MV variables with binary variables. This leads to binary logical functions representing the original MV function. The binary variables of one MV variable are interdependent, which leads to a restricted state space of the binary variables. In this setting the notion of implicant will be extended and the merging procedure described above will be adapted to find all prime implicants according to the extended notion. In the beginning, the encoding of the MV variables is chosen such that this adapted minimisation procedure will lead to a short and also well readable expression to represent the MV logical function.

The concepts from logical synthesis can be generalised from binary (Boolean) to multi-valued (MV) logic [Brayton et al., 1990]. This holds in particular for all definitions of Sec. 7.3.1. The generalisation used here is different from [Brayton et al., 1990]. It is tailored for the application to biological networks, with the goal to give a well readable representation. Once a generalisation is defined, the next question is how to deal with the new formalism computationally. On the one hand, computational methods from the binary setting can be adapted for the MV functions. On the other hand, the MV functions can be encoded in binary form, so that the methods that were developed for Boolean functions can be applied. A variation of such encodings is possible. As pointed out in [Brayton et al., 1990], the efficient processing of MV functions depends on a good choice of the encoding. In [Villa and Sangiovanni-Vincentelli, 1990] a general method

is presented that finds an optimal encoding for a given MV function, in the sense that a MV prime-cover corresponds to a prime cover in the binary encoding of the same size.

Apart from the issue of efficient processing of the function, also the expressiveness of the binary encoding is of interest. As pointed out in [Van Ham, 1979] a 2^k-valued variable can be encoded by k binary variables. However, these binary variables will have no individual meaning w.r.t. the biological network under consideration. For our purpose here we chose an encoding which allows good computational handling and, at the same time, consists of meaningful variables. This ensures that the resulting minimal expressions represent the biological information in a compact and immediately understandable form. This is an important feature, because otherwise an additional method would be needed to translate the obtained representation, so that it is comprehensible. The goal is that the modeller can read of the interactions directly from the Boolean expressions.

Notation A multi-valued domain consists of consecutive integers from 0 to a maximal value k. This domain is denoted by $\mathbb{M}_k := \{0, 1, \ldots, k\} \subset \mathbb{Z}$, so that in particular $\mathbb{M}_1 = \mathbb{B}$. A statement about variables in square brackets is meant to be interpreted as a $\mathbb{B}$-value, according to the validity of the statement. For example consider the variables $x \in \mathbb{M}_5,\ a, b \in \mathbb{B}$ and let $x = 5,\ a = 1,\ b = 0$, then we have the following equalities of $\mathbb{B}$-values (indicated by $\equiv$ to distinguish it from arithmetic equality): $[3 \geq x] \equiv 0,\ [3 \leq x] \equiv a \cdot \bar{b}$ and $[x = 3] \equiv b$.

A MV logical function is a mapping $\phi\colon \prod_{i=1}^{n} \mathbb{M}_{k(i)} \to \mathbb{M}_h$ for some $h \geq 1$. If $h > 1$, ϕ can be represented by several $\mathbb{B}$-valued functions ϕ_j, $j = 1, \ldots, h$, where ϕ_j evaluates to $\phi_j(z) \equiv 1$ at exactly those states $z \in \prod_{i=1}^{n} \mathbb{M}_{k(i)}$ which are mapped to $\phi(z) = j$. These states are the true points T_j of ϕ_j. Note that these are disjoint sets, i.e., $T_j \cap T_i = \varnothing$ if $j \neq i$, and $\phi(z) = 0$ if and only if $\phi_j(z) = 0$ for all $j = 1, \ldots, h$. Therefore, the mappings ϕ_j, $j = 1, \ldots, h$, can be seen as a decomposition of ϕ into MV logical functions with Boolean range. If such a decomposition is given, ϕ can be retrieved as $\phi(z) = j$, if $\phi_j(z) = 1$ and $\phi(z) = 0$, if $\phi_j(z) = 0$ for all $j = 1, \ldots, h$, since only one of these $h+1$ cases can be satisfied by z. In the following we will therefore focus the discussion on binary-valued functions $\phi\colon \prod_{i=1}^{n} \mathbb{M}_{k(i)} \to \mathbb{B}$.

7.4.1 Binary encoding and minimisation of MV functions

Encoding for MV variables The logical function ϕ takes binary values, but the components of the state $z \in \prod_{i=1}^{n} \mathbb{M}_{k(i)}$, in other words the variables, are MV. A single MV variable $z^i \in \mathbb{M}_{k(i)}$ will be encoded by the $k(i)$ binary variables

$$x_j^i :\equiv [z^i \geq j] \text{ for } j = 1, \ldots, k(i). \tag{7.1}$$

This encoding was introduced in [Van Ham, 1979]. It gives a natural representation of the ordering relation on the integers which correspond to the expression levels of the genes in the biological model. From the modelling point of view it is hence particularly suitable for biological regulatory networks, because in many cases the regulatory effect of a gene depends monotonically on the expression level (although very high concentrations of activating molecules sometimes have the contrary effect in gene regulation). Hence it often suffices to know that $z \leq k$ or $z \geq k$ in order to determine the regulatory effect on another component of the network. This encoding was also used for Constraint Satisfaction Problems in [Tamura et al., 2009], where the benefit is that ordering relations between variables can be expressed easily.

Defining MV-literals In the Boolean case the choice of literals (Def. 7.1) was rather canonical. For each variable a a positive and a negative literal is defined, stating that $a \equiv 1$ or $a \equiv 0$, respectively. A straightforward extension to the MV case of a variable $z^i \in \mathbb{M}_{k(i)}$ would be to consider every statement $[z^i = j]$ as a literal, $j = 0, \ldots, k(i)$. In [Brayton et al., 1990] the generalisation is based on a larger set of literals. Every subset $L \subset \mathbb{M}_{k(i)}$ defines a literal which states that z has a value from L, i.e., the literal expresses the disjunction $\sum_{l \in L}[z^i = l]$. Obviously, this larger set of literals makes it possible to represent logical functions by shorter expressions. Here we will choose a set of literals that lies between these two cases. Instead of defining a literal for every subset of $\mathbb{M}_{k(i)}$ we do it only for the intervals $I(j, j') := \{j, j+1, \ldots, j'\} \subset \mathbb{M}_{k(i)}$, with $j, j' \in \mathbb{M}_{k(i)}$, $j < j'$. The encoding (7.1) gives the binary variables $x^i_j \equiv [z^i \geq j]$ and $\overline{x}^i_j \equiv [z^i \leq j-1]$, for $j = 1, \ldots, k$. Every interval is thus given by the product of two binary variables or by only one binary variable, i.e.,

$$z^i \in I(j, j') \quad \Leftrightarrow \quad x^i_j \cdot \overline{x}^i_{j'+1}$$

if we define the symbols $\overline{x}^i_{k(i)+1} :\equiv 1$ and $x^i_0 :\equiv 1$.

Arithmetic literals and polynomial representation The sets of literals that were presented above have in common that each individual literal refers to one single MV variable. In principle it would also be possible to encode relations between several variables in one literal by using arithmetic expressions. For example, we could add the literals $[a \leq b]$ or $[a + b \leq 4]$ for a pair of MV variables $a, b \in \mathbb{M}_k$ (where $\leq$ denotes the usual order on $\mathbb{R}$). This would allow even shorter expressions, e.g., if $[a \leq b]$ is not available as literal it has to be expressed as the disjunction $\sum_{j=1}^{k} \overline{[a \geq j]} \cdot [b \geq j+1]$. How an algorithm for finding a minimal expression with such 'mixed' literals could be designed is unclear, since existing approaches (or at least those mentioned in this text) depend crucially on the separation of variables in the literals. In this context it should be noted that a totally different and purely arithmetic description of functions $f: \prod_{i=1}^{n} \mathbb{M}_{k(i)} \to \mathbb{B}$ can be given by polynomials over finite fields [Dingel and Milenkovic, 2008].

Encoding of MV functions The binary encoding of MV variables $z \in \prod_{i=1}^{n} \mathbb{M}_{k(i)}$ given by (7.1) leads to a binary function $f: \mathbb{B}^q \to \mathbb{B}$, where q is the number of binary variables needed for the encoding. According to (7.1) we have $q = \sum_{i=1}^{n} k(i)$. To map a state $z \in \prod_{i=1}^{n} \mathbb{M}_{k(i)}$ to its binary encoding, we will define a mapping $\beta: \prod_{i=1}^{n} \mathbb{M}_{k(i)} \to \mathbb{B}^q$.

Let $J := \{(i, j) : i \in \{1, \ldots, n\},\ j \in \{1, \ldots, k(i)\}\}$ and note that for each $m \in \{1, \ldots, q\}$ there is exactly one pair $(i, j) \in J$, such that $m = j + \sum_{h=1}^{i-1} k(h)$. Therefore the m-th component of the image of z will be defined by

$$\beta(z)_m \equiv x^i_j \equiv [z_i \geq j] \quad \text{for } (i, j) \in J \text{ with } m = j + \sum_{l=1}^{i-1} k(l).$$

To refer easily to all those binary variables that encode one distinct MV variable z_l, we introduce the notion of an l**-section** for the binary variables $x^l_1, \ldots, x^l_{k(l)}$ and the corresponding entries in the vector notation of an implicant.

Example A MV state $z \in \prod_{i=1}^{n} \mathbb{M}_{k(i)} = \mathbb{M}_3 \times \mathbb{M}_4 \times \mathbb{M}_4 \times \mathbb{M}_3 \times \mathbb{M}_1$ is encoded as $x \in \mathbb{B}^{15}$ and the l-sections partition the state as follows:

$$x = (\overbrace{x_1^1 \quad x_2^1 \quad x_3^1}^{1-section} \quad \overbrace{x_1^2 \quad x_2^2 \quad x_3^2 \quad x_4^2}^{2-section} \quad \overbrace{x_1^3 \quad x_2^3 \quad x_3^3 \quad x_4^3}^{3-section} \quad \overbrace{x_1^4 \quad x_2^4 \quad x_3^4}^{4-section} \quad \overbrace{x_1^5}^{5-section}).$$

Lemma 7.9. *A state $x \in \mathbb{B}^q$ is in the image of β if and only if it obeys the dependencies $x_{i+1}^l \Rightarrow x_i^l$ for all $l = 1, \ldots, n$ and $i = 1, \ldots, k(l) - 1$.*

It is clear that these implications are necessary and in the other direction a state that satisfies all implications uniquely defines a state of the MV variable according to (7.1). In particular, the image $\mathrm{im}(\beta) \subset \mathbb{B}^q$ is completely characterised by the implications between the Boolean variables. The size of the image of β is just the number of elements of $\prod_{i=1}^{n} \mathbb{M}_{k(i)}$, which is $\prod_i (k(i) + 1)$. The Boolean state space $\mathbb{B}^q$ has $2^{\sum_i k(i)} = \prod_i 2^{k(i)}$ elements. If at least one $k(i) > 1$, we have $\prod_i (k(i) + 1) < \prod_i 2^{k(i)}$. This gap is getting very big when we have several variables. For example, five variables from $\mathbb{M}_3$ have a state space with 243 elements. But β maps into $\mathbb{B}^{15}$ which has 134 times as many elements.

Figure 7.2: The four states of a single variable $x \in \mathbb{M}_4$ and its image under the map β onto the cube of the three Boolean variables $[x \geq 1]$, $[x \geq 2]$, $[x \geq 3]$ on the x, y, z-axis. Elements outside the image of β are contradictory assignments as e.g. $[x \geq 1] \cdot [x \geq 2] \cdot \overline{[x \geq 3]}$, corresponding to the front-left-top vertex.

Encoding the function as binary The MV function $\phi \colon \prod_{i=1}^{n} \mathbb{M}_{k(i)} \to \mathbb{B}$ can be encoded as a binary function $f \colon \mathbb{B}^q \to \mathbb{B}$. For elements $x \notin \mathrm{im}(\beta)$ we define $f(x) = 0$ and on the image of β as the composition of β and ϕ:

$$f(\beta(z)) := \phi(z).$$

In order to be able to compare Boolean expression in q variables with such a binary encoding f (in particular to consider minterms in q binary variables as implicants of f), it is necessary to define f also outside of the image of β.

Ordering of binary encoded functions Let f be the binary encoding of the MV function ϕ. The implicants of f are minterms over the q Boolean variables that encode the MV states. Due to the implications given in Lemma 7.9, additional minterms can be considered as implicants, see for example Fig. 7.3. For binary functions $g, f \colon \mathbb{B}^q \to \mathbb{B}$ the order relation $g \leq f$ was defined by $g(x) \equiv 1 \Rightarrow f(x) \equiv 1$ for all $x \in \mathbb{B}^q$. In case that g, f are binary encodings obtained with β, we will consider instead the order $g \preccurlyeq f$ defined by $g(x) \equiv 1 \Rightarrow f(x) \equiv 1$ for all $x \in im(\beta) \subset \mathbb{B}^q$. From the definitions it follows directly that all implicants according to $\leq$ are also implicants according to $\preccurlyeq$.

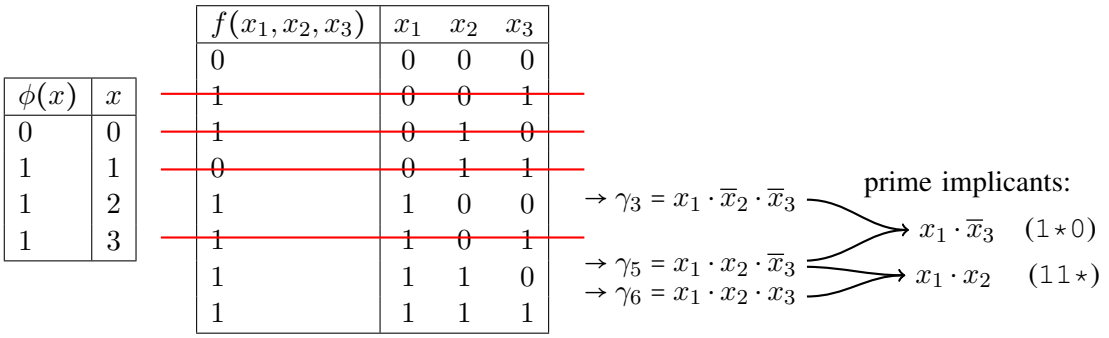


Figure 7.3: An example of a binary encoding. Assume that the function f from Fig. 7.1 encodes a multi-valued function $\phi\colon \mathbb{M}_3 \to \mathbb{B}$. Instead of the Boolean variables a, b, c from Fig. 7.1 we then have $x_1 := [z \geq 1], x_2 := [z \geq 2]$ and $x_3 := [z \geq 3]$ to encode the variable $z \in \mathbb{M}_3$. The encoding with $\beta\colon \mathbb{M}_3 \to \mathbb{B}_1^3$ gives the value of z in these Boolean variables. The truth-table of f is reduced, since not all elements in $\mathbb{B}^3$ are in the image of β. According to Lemma 7.9, this is the case for four of the states and these are crossed out in the truth-table. Three true points of f remain. The merging-rule can be applied two times to obtain the two prime implicants $1\star0$ and $11\star$ (in vector notation). According to the standard order $\leq$, these are prime implicants. Hence they cannot be merged in the Quine-McCluskey algorithm and $f(x_1, x_2, x_3) \equiv x_1 \cdot \overline{x}_3 + x_1 \cdot x_2$ is the minimal DNF. However, if we take the MV logic into account then $x_1 \cdot \overline{x}_3 :\Leftrightarrow [x \geq 1] \cdot [x < 3] \equiv [1 \leq x \leq 2]$ and $x_1 \cdot x_2 :\Leftrightarrow [x \geq 1] \cdot [x \geq 2] \equiv [x \geq 2]$. So actually $f(x) \equiv [1 \leq x \leq 2] + [x \geq 2]$ and this is obviously equivalent to $[x \geq 1]$, so that we can write $f \equiv x_1$. In the ordering $\preccurlyeq$ the minterm x_1 is in fact an implicant of f and hence $x_1 \cdot \overline{x}_3$ and $x_1 \cdot x_2$ are not prime implicants.

7.4.2 Minimising the Boolean expression of the binary encoding

The example in Fig. 7.3 shows that the binary encoding of MV functions can be reduced further, if the implications between the Boolean variables are taken into account. We will now show how the merging procedure `Binary-Merge` described above can be extended to overcome this limitation. Other techniques to generate the prime implicants might be modifiable in a similar way. This is the case e.g. for the method described in [Crama and Hammer, 2011, Sec. 3.2.1], where every implicant is generated from a pair of true points. To every pair of true points an operation is applied that yields a new implicant (the so-called hull) and it has to be checked whether this is an implicant or not. It is hence straightforward to modify this technique by checking if the hull is an implicant according to the relation $\preccurlyeq$. For `Binary-Merge` we will define the modification formally and prove that all prime implicants according to $\preccurlyeq$ are generated.

Block structure of l-sections The difference between the relations $\leq$ and $\preccurlyeq$ lies in the implications between the binary variables of one l-section which define the image of β. Let f be the binary encoding of a MV function $\phi\colon \prod_{i=1}^{n} \mathbb{M}_{k(i)} \to \mathbb{B}$, i.e., $f\colon \mathrm{im}(\beta) \to \mathbb{B}$. Let α be the vector of an implicant of f. From Lemma 7.9 we can infer that in each l-section a 1-entry is only preceded by 1-entries. Furthermore, a 0-entry can only be succeeded by 0-entries. This means that the l-section consist of a block of 1s followed by a block of $\star$, followed by a block of 0s. One or two of these blocks can be empty.

Lemma 7.10. *Let f be as above and assume the truth table of f is given. The merging procedure* `Binary-Merge` *given on page 150 results in at most one $*$ in each l-section.*

Proof. Every merge in `Binary-Merge` gives one new $*$ and the initial input consists of implicants that do not contain any $*$. Hence, to get by merging a minterm μ with two $*$ in one l-section, it would be necessary that two implicants are merged which are identical outside this l-section and have each exactly one $*$ in it. Due to the block-structure just mentioned, these l-sections must be of the form `1...1*0...0`. Hence, if the implicants are not identical on this l-section then their $*$ is located at different entries and they cannot be merged. □

Merging procedure for encodings of MV functions The merging procedure `Binary-Merge` is extended to `Multi-Merge` by allowing to additionally merge pairs α, α' of implicant vectors according to the following rule:

`Multi-Merge` rule:

`If:`
- α and α' are identical except on one l-section and for all entries j in this l-section $\alpha_j = \alpha'_j$ or $\alpha_j = *$ or $\alpha'_j = *$.

`Then:`
- Merge to α'' as follows. In all entries j where $\alpha_j = \alpha_j$, set $\alpha''_j := \alpha_j$. In all other entries we have $\alpha_j = *$ or $\alpha_j = *$ and define $\alpha''_j := *$.

This rule is applied additionally, which means that a pair is merged according to the rule `Binary-Merge` if possible, but in case this rule does not apply the pair can also be merged according to the `Multi-Merge` rule.

α: ...***000...	α: ...11***0...	α: ...111**0...
α': ...111***...	α': ...1**000...	α': ...1*0000...
α'': ...******...	α'': ...1****0...	merge not allowed

Figure 7.4: **Example** of the application of the `Multi-Merge` rule to three different pairs α, α' of implicant vectors. Only one l-section is shown, the implicants are assumed to be identical in all omitted entries. For the last pair the condition for merging is not fulfilled.

Proposition 7.11. The merging procedure `Multi-Merge`, that is, `Binary-Merge` extended by additionally merging pairs according to the `Multi-Merge` rule above, generates all implicants according to the ordering relation $\preccurlyeq$ and the prime implicants are exactly those that were never merged.

Proof. At first we will show that the new minterms α'' are in fact implicants of f. Similar to `Binary-Merge` the new minterms turn out to be equal to the disjunction of the pair of implicants, i.e., $\alpha'' = \alpha + \alpha'$, which shows that they are implicants as well. To see this identity, it

suffices to consider only the single l-section where α, α' are not identical. Let j' be the largest index in this l-section, such that $\alpha_{j'} = 1$, and j'' the smallest index, such that $\alpha_{j''} = 0$. Due to the block structure, this l-section of α encodes $[j' \leq z_l < j'']$. Similarly we can define h', h'' for α'. The condition for merging α and α' is satisfied if and only if $h' < j''$ and $j' < h''$, because if $h' \geq j''$ we have $\alpha_{h'} = 0$ and $\alpha'_{h'} = 1$, and if $j' \geq h''$ we have $\alpha_{j'} = 1$ and $\alpha'_{j'} = 0$. For the MV variable z_l the disjunction $\alpha + \alpha'$ encodes $[j' \leq z_l < j''] + [h' \leq z_l < h''] \equiv [\min(j', h') \leq z_l < \max(j'', h'')]$. Note that j', j'' can equivalently be defined by $\alpha_j = * \Leftrightarrow j' < j < j''$, as a direct consequence of the block structure. Therefore for the l-section of the disjunction $\omega := \alpha + \alpha'$ we have $\omega_j = * \Leftrightarrow \min(j', h') < j < \max(j'', h'')$, which shows that $\omega = \alpha''$.

We proceed similarly as in the proof of Prop. 7.8 and define a splitting operation that is inverting the merging extended with the `Multi-Merge` rule. Those implicants where at most one $*$ occurs in each l-section of the vector are split in the same way as described in the proof of Prop. 7.8. For all other implicants we define a splitting operation that is the inverse to the extended merging described above. Let α be the vector representation of such an implicant. An l-section where the number of $*$-entries is $d > 1$ is fixed. These entries constitute a block starting at some index s, i.e., $\alpha_s = \alpha_{s+1} = \cdots = \alpha_{s+d-1} = *$. We split α into h implicants $\omega^1, \ldots, \omega^h$. Each ω^i has exactly one $*$ in this l-section and this $*$ is located at ω^i_{s+i-1} for $i = 1, \ldots, h$. See also the example below. Note that, due to the block structure, the l-section is completely determined by fixing the index of the single $*$. We conclude that the described splitting operation is well defined.

By using this splitting operation, all merges in `Multi-Merge` can be inverted and the argument from the proof of Prop. 7.8 can be adapted to see that all implicants must be generated in `Multi-Merge` and the prime implicants are exactly those elements that are never merged. □

Example Splitting of an l-section of α with $d = 3$ into $\omega_1, \omega_2, \omega_3$. The dots '..' indicate the variables outside the l-section which are identical in $\alpha, \omega_1, \omega_2$ and ω_3.

```
..11***0..  ──┬──►  ..11*000..
              ├──►  ..111*00..
              └──►  ..1111*0..
```

Further simplification of the minimal DNF The procedure `Multi-Merge` thus gives us the set of all prime implicants w.r.t. the ordering by $\preccurlyeq$. Selecting a minimal cover from this set gives a minimal DNF composed of the MV-literals defined at the beginning of Sec. 7.4.1. Due to the choice of these MV-literals, this DNF gives a well readable representation of the logical function. To further improve the readability, it is straightforward to apply additionally the following simplifications:

- $[z^i \geq j] \cdot \overline{[z^i \geq j+1]}$ is substituted with $[z^i = j]$
- $[z^i \geq j] \cdot \overline{[z^i \geq h]}$ with $h \geq j+2$ is substituted with $[h-1 \geq z^i \geq j]$.

This way we make sure that every MV literal, i.e., every interval of $\mathbb{M}_{k(i)}$ for some $i \in \{1, \ldots, n\}$, is represented by one single binary literal. Furthermore, negations can be eliminated by applying the rule that

- $\overline{[z^i \geq j]}$ is substituted with $[z^i \leq j-1]$ if $j \geq 2$ and in the case $j = 1$ with $[z^i = 0]$.

7.5 Representation of logical networks and their dynamics

The formal derivation of the update functions of a logical network, which describe the dynamics of the logical network starting from the target functions, will be given. This is also an example where minimisation of Boolean expressions is necessary, because the formal construction of the update function leads to large Boolean expressions, which can be drastically simplified. An example of application is given, based on an implementation of the algorithm.

The representation of logical networks was introduced in Sec. 7.2. As mentioned before, the biological knowledge is usually given by the target functions. Each component $l \in \{1,\dots,n\}$ has a target function $\phi^l\colon \prod_{i=1}^n \mathbb{M}_{k(i)} \to \mathbb{M}_{k(l)}$, which can be equivalently encoded by the mapping β in $k(l)+1$ binary functions as described in the previous section. For ϕ^l the binary functions $f_j^l\colon \mathrm{im}(\beta) \to \mathbb{B}$ are defined by

$$f_j^l(\beta(x)) = \begin{cases} 1, & \text{if } \phi^l(x) = j \\ 0, & \text{otherwise} \end{cases} \quad , \quad j = 0,1,\dots,k(l). \tag{7.2}$$

In fact, one of the $k(l)+1$ functions f_j^l is implicitly defined by $f_j^l(x) = 1 \Leftrightarrow \sum_{i\neq j} f_i^j(\beta(x)) = 0$. Alternatively, we can also define the binary functions by inequalities instead of equalities as follows.

$$g_j^l(\beta(x)) = \begin{cases} 1, & \text{if } \phi^l(x) \geq j \\ 0, & \text{otherwise} \end{cases} \quad , \quad j = 1,\dots,k(l) \tag{7.3}$$

and $g_0^l = \overline{g}_1^l$.

$\phi(x)$	x^1	x^2	$f_0(\beta(x))$	$f_1(\beta(x))$	$f_2(\beta(x))$	$f_3(\beta(x))$	$[x^1 \geq 1]$	$[x^1 \geq 2]$	$[x^2 \geq 1]$	$[x^2 \geq 2]$
0	0	0	1	0	0	0	0	0	0	0
2	1	0	0	0 (1)	1	0	1	0	0	0
3	2	0	0	0 (1)	0(1)	1	1	1	0	0
1	0	1	0	1	0	0	0	0	1	0
1	1	1	0	1	0	0	1	0	1	0
1	2	1	0	1	0	0	1	1	1	0
0	0	2	1	0	0	0	0	0	1	1
0	1	2	1	0	0	0	1	0	1	1
0	2	2	1	0	0	0	1	1	1	1

Figure 7.5: Example with a function $\phi\colon \mathbb{M}_2 \times \mathbb{M}_2 \to \mathbb{M}_3$. The evaluation on all states $x = (x^1, x^2) \in \mathbb{M}_2 \times \mathbb{M}_2$ is given in the left table. The right table contains the truth tables for the binary encodings f_0, f_1, f_2 and f_3 of ϕ. The MV variables x^1, x^2 are encoded by four binary variables. Only the states of the domain $\mathrm{im}(\beta) \subset \mathbb{B}^4$ are listed in the table. The values of the encoding functions refer to the first variant (7.2) with f_j^l, while the values of the alternative variant (7.3) with g_j^l are given in brackets, if they are different.

Composition of the update functions Once such a formal description of the network is given, the update functions $\Psi^l_\pm \colon \prod_{i=1}^n \mathbb{M}_{k(i)} \to \mathbb{B}$ can be derived. Using the second variant (7.3), we can write the update functions (without using g^l_0) as

$$\Psi^l_+(x) = \sum_{i=1}^{k(l)} [x^l \leq i-1] \cdot g^l_i(\beta(x)) \qquad \text{and} \qquad \Psi^l_-(x) = \sum_{i=1}^{k(l)} [x^l \geq i] \cdot \overline{g^l_i(\beta(x))}.$$

These are very large expressions and it is necessary to replace them by a short equivalent form to understand their meaning for the GRN model and to allow for further automatic processing of the model, as e.g. translation into other formats.

Implementation The `Multi-Merge` procedure was implemented as part of an tool to translate integrated logical models of GRN and metabolism into the Petri net format [Palinkas and Bockmayr, 2011] and into continuous time Markov chains (CTMCs, see Chap. 8) in the modelling language of PRISM. The derivation of update functions from target functions as outlined above is part of the translation procedure and the minimisation of the representation is hence a crucial part. The implementation of the whole tool was done in the python language and is available upon request.

Uniquely satisfied implicants The minimisation algorithm gives a DNF where all minterms are prime implicants. In the translation to Petri nets these minterms are translated to transitions. If several minterms of the DNF can be true in one state, this leads to several transitions realising the same shift of tokens which can be enabled simultaneously. In the state transition graph the consequence is that there are some states s and s' with multiple edges from s to s'. In nondeterministic network dynamics, it is usually not relevant if there is one or multiple edges between s and s'. However, it is important when we consider the extension to a stochastic Petri net. In that case every edge in the state transition graph is assigned a probability (or a rate in a CTMC). The multiplicity of edges from s to s' affects then the probability (or velocity) to get from state s to state s'. In such a situation it might be appropriate to require that at most one minterm in the DNF can be satisfied. This can be implemented by further manipulating the update functions, see [Palinkas, 2011]. The number of required products in the DNF might be increased when this property is enforced.

Usage of the python implementation In the python implementation of the `Multi-Merge` procedure, the function `minexpr` gives a minimal DNF of a given statement `bex` on MV variables $z_1, \ldots, z_n$. Any statement that is correctly evaluated to `True` or `False` by python (with the built-in function `eval`) is allowed. Internally, the algorithm represents the statement as a Boolean expression in the encoding $\beta(z)$. The range of a MV variable is $\{0, \ldots, k(i)\} \subset \mathbb{Z}$, hence only the maximal values $k(i)$ must be given. To assign to each MV variable its maximal value, a dictionary `cdict={name_z1:k1,...}` will be used. `minexpr` can now be called with these two arguments:

```
>>>import logicfun as lf
>>>bex='a>=1 and b>=2 or a==1 and b>=3 and c<=1'
>>>cdict={'a':3,'b':3,'c':3}
>>>lf.minexpr(bex,cdict)
'a>=1 and b>=2'
```

The following expression is an example of a binary encoding which is already in minimal DNF as a Boolean expression but can be further minimised if multi-merge is applied. (If all maximal values are equal, `cdict` can just be this integer instead of a dictionary.)

```
>>>bex=
'a>=1 and c>=1 or b>=2 and not c>=2 or not a>=3 or not b>=3'
>>>lf.minexpr(bex,3)
'True'
```

7.5.1 Illustrative application to a T cell model

As an illustrative example of a logical network that is given by MV logic target functions, we will consider a T-cell model presented in [Mendoza, 2006]. This model is available on the GINsim webpage (`http://www.ginsim.org/`) and was also translated into a Petri net representation [Chaouiya et al., 2011]. The algorithm presented there is not based on the Boolean representation of logical networks. Boolean minimisation is thus not available as a means to obtain a Petri net of minimal size. In fact, the size of the resulting Petri net depends on the order in which the components are processed in the algorithm. By picking different orderings Petri nets with 25 or more transitions were generated [Chaouiya et al., 2011]. The Petri nets translation of [Palinkas and Bockmayr, 2011], using the multiQMC algorithm, achieves immediately a minimal Petri net, where minimal means that the number of transitions (implicants) and the number of arcs (literals) are minimal. For the T-cell model a minimal Petri net contains 23 transitions. (Above we shortly discussed the additional property that at most one transition is allowed to be enabled in the DNF of the update function. When this is enforced, 24 transitions are needed.)

The model of the T-cell gene regulatory network is provided on the GINsim web page using logical parameters. This is just another format to describe the target functions and can be translated systematically to the representation with binary encodings [Palinkas, 2011]. For example, the logical parameters for the component `IFNgR` are

$$K_{IFNgR}(IFNg^1) = K_{IFNgR}(IFNg^1, SOCS1) = K_{IFNgR}(IFNg^2, SOCS1) = 1,$$

describing all sets of active interactions where the target value of $IFNgR$ is 1. All sets where the target value is 2 are given by $K_{IFNgR}(IFNg^2) = 2$. For all other sets of active interactions, the target value is 0. We can define the dynamics of `IFNgR` by its logical target function $\phi: \mathbb{M}_1 \times \mathbb{M}_2 \to \mathbb{M}_2$. Let f_0, f_1, f_2 be the binary encodings of ϕ. The logical parameters can be translated to Boolean expressions for f_0, f_1, f_2:

- The parameters tell us that `IFNgR` has target value 1 if gene `IFNg` is on level 1 or, if `IFNg` is on level 2 and at the same time `SOCS1` is active. So we get

$$f_1(IFNg, SOCS1) \equiv (IFNg \geq 1) \cdot \overline{(IFNg \geq 2)} + (IFNg \geq 1) \cdot (SOCS1 \geq 1)$$

- The target value is 2 whenever $IFNg$ is on its maximal level 2 but $SOCS1$ is off, so we have

$$f_2(IFNg, SOCS1) \equiv (IFNg \geq 2) \cdot \overline{(SOCS1 \geq 1)}$$

- For f_0 we can derive an expression that is the conjunction $\overline{f_1} \cdot \overline{f_2}$, which gives $(\overline{IFNg \geq 1}) \cdot (\overline{IFNg \geq 2}) + (\overline{IFNg \geq 1}) \cdot (SOCS1 \geq 1)$. Now multi-merge is necessary to reduce this expression to $\overline{(IFNg \geq 1)}$

The update functions for `IFNgR` represented in minimal DNF with the MV-literals introduced in Sec. 7.4.1 are

$$\Psi^{+}_{IFNgR} \equiv \texttt{IFNgR<=1 and IFNg==2 and SOCS1==0} \\ \texttt{or IFNgR==0 and IFNg>=1}$$

for up regulation and

$$\Psi^{-}_{IFNgR} \equiv \texttt{IFNgR>=1 and IFNg==0 or IFNgR==2 and IFNg<=1} \\ \texttt{or IFNgR==2 and SOCS1==1}$$

for down-regulation, as computed with the `Multi-Merge` procedure.

7.6 Discussion

Logical modelling is a very abstract approach for biological systems. The mechanisms which control metabolic or gene regulatory networks on the molecular level are not governed by logical rules but by kinetic parameters and also by stochastic fluctuations (as discussed in the next chapter). Although the logical rules can give a good representation of the phenomenology of the system, they might fail in some cases to capture the dynamics of the system correctly. Still, the logical approach is widely accepted. In fact, for uncovering design principles and motifs of regulatory networks such highly simplified approaches are sometimes more appropriate than quantitative and continuous models. They may help us to extract the principal mechanisms that cause an observed dynamic feature of the biological system. Although the biochemical processes in a cell are very complicated, the resulting dynamical properties of some parts can be rather simple and close to logical switches. In such a case, a complex differential equation might just reproduce a simple logical function and we reduce the mathematical complexity of the model by using this logical description. A further benefit of modelling with logical functions is the natural representation which is easily understandable without any knowledge about the underlying mathematical formalism. Logical expressions are basically an abbreviation of everyday language, with the only difference that they are stated with mathematical precision. Biological knowledge can easily be formulated by logical expressions. However, as the model is processed computationally, new logical formulas arise which contain redundancies and can be very large. Minimisation of logical expression is thus necessary to keep the representation of the model manageable for further treatment. Regarding multi-valued logic, the readability also depends on a convenient formal encoding of the multi-valued variables. Furthermore, a suitable algorithm must be available which allows for fast minimisation of this encoding. A solution was proposed in this chapter.

8 Stochastic modelling with logical networks

8.1 Overview

Stochastic modelling has been applied in various ways to biological systems. The main goal of this chapter is to introduce **stochastic logical networks**, a stochastic extension of logical networks introduced in Chap. 7. Quantitative and probabilistic aspects which were already inherent in the logical network can be revealed thereby and more biological information can be incorporated in the model. However, including quantitative information in an abstract formalism like logical networks raises the question if the resulting model is biologically meaningful. This question will be addressed by comparing stochastic logical networks with mechanistic models, namely ODEs and also the chemical master equation which is a stochastic model of molecular interactions. What the chemical master equation and stochastic logical networks have in common, is that they both define a continuous time Markov chain.

Stochastic logical networks are formally introduced in Sec. 8.3. An example illustrates the extension and the new aspects of the dynamics which can be observed. The extension is then applied to a logical network of a simple regulatory motif. This logical network was obtained from an ODE model and the stochastic logical network recovers the dynamics of the original ODE model. The relationship between the deterministic ODE model and the master equation is discussed in Sec. 8.2. In Sec. 8.3 a short overview of how stochastic logical networks relate to different deterministic or stochastic modelling approaches is given.

A very useful attribute of this stochastic extension of logical networks is that it shares with the master equation the formalism of continuous time Markov chains. As a consequence, we can incorporate the master equation into a stochastic logical network. This way, distinct components and interactions can be modelled by detailed kinetics taking quantitative and stochastic effects into account while keeping the complexity of the model as low as possible. In Sec. 8.4, such a combination of the master equation and a stochastic logical network is illustrated on a model for galactose utilisation in yeast. It allows reproducing stochastic effects which cannot be captured by the purely logical model, and which have been previously observed with a fully kinetic model of much higher complexity.

8.1.1 Background - stochastic modelling

Molecular systems, such as gene regulatory networks, metabolic networks or signal transduction have been modelled in many different ways ranging from stochastic to deterministic and from discrete to continuous methods. Apart from approaches that stick to one particular theoretical framework, hybrid models were developed, e.g. [Alfonsi et al., 2005], which combine

different mathematical descriptions, motivated by the fact that, depending on the biochemical mechanisms, different mathematical models may be appropriate or tractable. Metabolic reactions mostly occur very fast and the amount of substrates and products is much larger. Stochastic modelling is hence not tractable [Puchałka and Kierzek, 2004]. At the same time, the effects of random fluctuations can be expected to be marginal. As a consequence, deterministic ODE models of metabolism can describe the dynamics very accurately. In contrast, ODE models of gene regulation might not capture important stochastic effects caused by the small number of regulatory proteins, which can determine the regulatory dynamics and even cell fate [Elowitz et al., 2002, Oppenheim et al., 2005, Kærn et al., 2005]. To account for these effects, the molecular interactions can be modelled as a stochastic process using the master equation, which describes the probability of molecular interactions. The chemical master equation is a Kolmogorov forward equation and hence defines uniquely a continuous time Markov chain (CTMC), assuming that some technical conditions are satisfied, see [Grimmett and Stirzaker, 1992, p. 242]. Even though this mathematical framework is very different from the ODE description of a biochemical system, both theories are closely related. In contrast to these two mechanistic modelling approaches on the molecular level, the logical networks introduced in Chap. 7 give a very abstract and rather phenomenological description of the interactions.

Probabilistic modelling of abstract logical networks Probabilistic modelling with abstract logical networks has often been implemented using discrete time Markov chains (MCs). In [Kim et al., 2002] a gene regulatory network (GRN) was modelled by a Markov chain, where the transition probabilities between different states of the GRN were derived from microarray data. Stochastic perturbations of single components of a Boolean network were modelled in [Garg et al., 2009] by randomly changing the activity of a single component, which also gives rise to a MC. Most studies in this direction are based on the modelling formalism of probabilistic Boolean networks [Shmulevich et al., 2002] which describes logical networks with binary components (hence Boolean networks) where the regulation of a component is not fixed to one update function (cf. p. 145). Instead, several possible update functions for one component are given together with associated probabilities. A synchronous update of the network is then performed, which means that for each component, one of the target functions is chosen randomly with the associated probability. An extension of probabilistic Boolean networks to asynchronous updates, as well as more complex update schemes, was described in [Merle and Bourdon, 2010]. In [Liang and Han, 2012] a computational approach to simulate probabilistic Boolean networks is presented. Based on *random bit streams* [Han et al., 2014], the transition probabilities between different states are computed by simulation. In contrast, the transition rates in the stochastic logical networks presented in this chapter are already incorporated in the definition of the MC. The method [Liang and Han, 2012] also allows modelling of perturbations of the individual components, implemented by randomly changing the activity of single components, similar to [Garg et al., 2009]. All these approaches from the literature have in common that they describe MCs and that they are aimed to analyse the long-term behaviour. While [Liang and Han, 2012, Kim et al., 2002] are computing the stationary distribution of the CTMC (see p. 17), the robustness of given attractors to random perturbations is analysed in [Garg et al., 2009]. The motivation for these approaches is always the uncertainty of the regulatory interactions and the random fluctuations in the expression levels. The state space of the used formalisms is discrete. Also the time is discretised and not continuous as in CTMCs. These approaches are therefore not comparable with the stochastic logical networks introduced

here, because they differ in the modelling approach as well as in the mathematical formalism.

When continuous time Markov chains are used to model biological systems, this is mostly done in a kinetic approach, similar to the chemical master equation, see e.g. [Calder et al., 2006, Kim et al., 2002, Kwiatkowska et al., 2008, Murray, 1989]. In contrast, stochastic logical networks are not a kinetic approach, but the motivation is to replace the nondeterminism of the abstract logical network model with probabilities for the transitions in the state transition graph and to observe the resulting stochastic and quantitative effects in the dynamics. This model reflects the fact that the exact time point of a regulatory event is always random, which is modelled in a straightforward manner by probabilistic transition rates. In contrast to the above-mentioned modelling approaches, uncertainty about the existence and the type of the regulatory interactions are not considered. Neither are random perturbations of single components.

The practical analysis of stochastic logical networks can be done by probabilistic model checking with the PRISM model checker [Hinton et al., 2006], which can verify and give the probability for dynamic properties of all kinds, short-term behaviour as well as properties of the stationary distribution. These computational methods have been applied in various ways to analyse biochemical systems [Calder et al., 2006, Kim et al., 2002, Yuan et al., 2011].

8.2 Markov chain models of biochemical reactions and regulatory interactions

The chemical master equation and continuous time Markov chains (CTMCs) are introduced formally and the most important properties are discussed. The master equation is a classical application of CTCMs to biological systems. An example is given and the connection of the master equation to deterministic ODE models is discussed. The difference between these two mathematical models lies solely in the fluctuations of molecule amounts in the master equation model. As larger volumes are considered and the amounts of molecules are hence rising, the fluctuations are diminished and the probabilistic model converges to the deterministic ODE model in the so-called thermodynamic limit.

8.2.1 Chemical master equation and CTMCs

The chemical master equation The chemical master equation gives a discrete and stochastic description of reactions or regulatory interactions of molecules. The state of the modelled system is given at every time point by the number of molecules of each species. The rate of an interaction or reaction of several molecules is computationally determined by the collision probability of these molecules [Gillespie, 1976]. Formally, if m species are given, the state space is $U \subset \mathbb{N}_0^m$ and the vector $u \in U$ describes a state, assigning to each species a number of molecules. The chemical master equation can be used to describe metabolic reactions [Levine and Hwa, 2007], but also to model signalling pathways [Calder et al., 2006] and regulatory interactions, e.g. binding of proteins to DNA, phosphorylation of regulatory molecules, etc., see [Thattai and Van Oudenaarden, 2001, Hegland et al., 2007], [De Jong, 2002] and references therein.

Each reaction or interaction, denoted $j = 1, \ldots, n$, has a corresponding rate function (also called propensity function [Gillespie, 1977]) $r_j : U \to \mathbb{R}_{\geq 0}$, which gives the rate of the reaction or interaction in dependence of the state, i.e., the amounts of molecules of the distinct species. The rate $r_j(u)$ is zero, if and only if the reaction or interaction is not possible in state u. If reaction or interaction j occurs in state $u \in U$, the consequence is a transition to a new state $u' \in U$. We assume that at most one reaction or interaction j realises the transition (u, u'). The transitions and their rates define a generator matrix $G \in \mathbb{R}^{U \times U}$ of a CTMC on the state space U (see p. 17). The off-diagonal entries are then $g_{u,u'} = r_j(u)$ or $g_{u,u'} = 0$ in case no reaction or interaction realises this transition. The diagonal entries are by definition $g_{u,u} = -\sum_{v \in U \setminus \{u\}} g_{u,v}$. This definition also holds for a countable infinite state space U, in case the number of outgoing transitions is finite in all states. For every fixed initial state $u_0 \in U$ this CTMC defines a stochastic process $\{X_t\}_{t \in [0,\infty)}$, where $X_t : \Omega \to U$ is a random variable and $P(X_0 = x_0) = 1$ (see p. 17).

The embedded Markov chain If there is a transition from u to u', the waiting time until it triggers is exponentially distributed [Grimmett and Stirzaker, 1992, p. 95, 243] with parameter $g_{u,u'}$ [Baier et al., 2003]. Until the stochastic process triggers, it rests in u. In case u has several outgoing transitions, these are in competition to trigger first (the so-called race condition). The first transition that is triggered determines the next state. The probability that the next state will be u' is given by $\frac{g_{u,u'}}{-g_{u,u}}$. Since $g_{u,u} = -\sum_{v \in U \setminus \{u\}} g_{u,v}$, these probabilities of all the outgoing transitions sum up to one and hence they define a discrete time Markov chain (MC) on the state space U (see p. 17) which is called the embedded Markov chain of the CTMC.

Dynamics of a CTMC: Trajectories, the mean trajectory and the stationary distribution. Here we consider the dynamics of the CTMC or the embedded MC starting at a fixed initial state $u_0 \in U$. That is, we have a stochastic process $\{X_t : t \in T\}$, with $T = \mathbb{R}_{\geq 0}$ and $P(X_0 = u_0) = 1$. A realisation of this process is described by the sequence of states u_i that are visited, starting in u_0, and the waiting times d_i between the transitions. Formally, this gives a sequence of states and the associated waiting times, i.e., $((u_0, d_0), (u_1, d_1), (u_2, d_2), \ldots)$ with $u_i \in U$ and $d_i > 0$, $i = 0, 1, 2, \ldots$. If the state space is infinite or if the transitions constitute cycles, these sequences can be infinite. By neglecting the waiting times we get a sequence $(u_1, u_1, u_2, \ldots)$ of states, describing the dynamics in discrete time steps. To distinguish these two objects, we will call $((u_0, d_0), (u_1, d_1), \ldots)$ a trajectory and $(u_0, u_1, \ldots)$ a sequence of states. While the trajectory is a realisation of the CTMC, the sequence is a realisation of the embedded MC. By construction, the probability of a distinct sequence of states is the same in the CTMC and in the embedded Markov chain. However, the CTMC has an infinite set of trajectories corresponding to one sequence of states.

Model checking - probabilistic extensions Model checking denotes formal methods which can be used to verify properties of the state transition graph (STG) of a logical network or of another model that can be formulated as a transition system (cf. Sec. 7.2). For CTMCs as well as for MCs and related objects, the temporal logic of model checking was extended and computational procedures were developed to perform probabilistic model checking. The computer tool PRISM [Hinton et al., 2006] can carry out these computations. Some features of PRISM will be introduced now and will be used later. In probabilistic model checking we can distinguish between transient properties and steady state properties. Transient properties concern

the system at a certain time point or within a finite time interval, while the so-called steady state properties refer to the stationary distribution of the CTMC or MC (see p. 17). Here we will only deal with transient properties, although steady state properties are also well suited to analyse biological models [Kwiatkowska et al., 2008, Palinkas, 2011]. A theoretical sketch of what model checking is telling us can be given as follows. The complete dynamics of a CTMC with a given initial state is described by the set of all trajectories starting in the initial state. The rates define a probability measure on this set [Baier et al., 2003]. In most cases, a dynamic property can be formulated as a property of a trajectory in the CTMC. The subset of all trajectories with this property has then a probability measure which can be computed with PRISM [Hinton et al., 2006]. The same holds for MCs and the sequences of states, which inherit a probability measure from the transition probabilities [Baier and Katoen, 2008, p. 757]. Furthermore, PRISM is able to consider the mean of all trajectories by using so-called reward structures on the CTMC [Kwiatkowska et al., 2008]. This allows computing at a given time point the expectation value of each component of the states. The trajectory that is constituted by the expectations of all components at all time points will be called the **mean trajectory** here.

Simulating a CTMC For the computational simulation of the CTMCs, the stochastic simulation algorithm [Gillespie, 1977] can be used. In most applications, a large number of simulation runs of the CTMC are carried out in order to approximate the average behaviour, i.e., the mean trajectory. In contrast to the computational procedure of PRISM, this gives just an approximation of the mean values, but it contains all components and all time points in the (finite) interval that was simulated.

The chemical master equation and differential equation models In the 70s, Gillespie gave a theoretical justification for the master equation based on mechanistic considerations and proposed a Monte Carlo method to simulate the master equation [Gillespie, 1976]. The basis of this theoretical derivation was different from the deterministic theory which leads to the classical ODE model [Gillespie, 1976, Oppenheim et al., 2003]. The master equation is a description of an ensemble of molecules that are well-mixed. A state of the system is only specified by the numbers of molecules. Location and momentum are not considered. To account for this missing information, a collision probability for molecules is introduced, based on the current amounts of the different species. This is based on the assumption that the system is well-mixed, meaning that the location of an individual molecule is given in probabilistic terms by a uniform distribution. Today, stochastic effects have been recognized to be necessary for a comprehensive description and understanding of certain regulatory mechanisms [Elowitz et al., 2002, Arkin et al., 1998, Alfonsi et al., 2005] and the master equation, in particular the simulation with Gillespie's algorithm, is widely used.

8.2.2 Example: CTMC model of an autoregulated gene

The mass action kinetics of chemical reactions can be naturally expressed with the master equation. In fact, the formal representation is very similar to the usual ODE model. In the master equation the rate functions and the stoichiometric coefficients describe the changes in concentration, which are given by the derivative in the ODE model. As an example we will consider now a single gene which is autoregulated. This simple regulatory motif is known to promote bistability of the gene expression, see [Alon, 2007, p. 37] or [Thomas et al., 1995].

Differential equation of autoregulation Let x denote the concentration of the gene product. In GRN models it is interpreted as the activity or expression level of the gene. The ODE description is given by the derivative

$$\frac{\mathrm{d}x}{\mathrm{d}t} = (k_a - k_b)\frac{x^h}{\theta^h + x^h} + k_b - k_{deg}x \tag{8.1}$$

We will write this as $\frac{\mathrm{d}x}{\mathrm{d}t} = r_{syn}(x) - r_{deg}(x)$, the function for the synthesis rate is given by the Hill function $r_{syn}(x) = (k_a - k_b)\frac{x^h}{\theta^h + x^h} + k_b$, where h is the Hill parameter, determining the slope of the switch [Alon, 2007, p. 13]. This function describes the switch between a basal rate k_b, in case autoregulation is not active, and the synthesis rate k_a if autoregulation is active. The degradation rate of the gene product is given by $r_{deg}(x) = k_{deg}x$, where k_{deg} is a kinetic parameter. In case of positive autoregulation we have $k_b < k_a$ and the two synthesis rates k_b, k_a determine two different equilibrium points $eq1 = k_b/k_{deg}$ and $eq2 = k_a/k_{deg}$, i.e., $\frac{\mathrm{d}x}{\mathrm{d}t} = 0$ for $x = eq1$ or $x = eq2$. Assuming that $eq1 < \theta < eq2$, the expression level approaches the lower equilibrium $eq1$ as soon as it is below the threshold θ. Above θ the expression tends to $eq2$. Negative autoregulation is given when $k_b > k_a$. If we assume that $k_b > \theta > k_a$, the expression level will tend to θ from every point. In fact, this feature leads to increased stability of the expression level, as will be discussed in Sec. 8.4.1.

In the limit $h \to \infty$, the switch between the two expression levels occurs immediately, i.e., the Hill kinetics of the synthesis approximates a step function

$$r_{syn}(x) = \begin{cases} k_b, & x < \theta \\ (k_a + k_b)/2, & x = \theta \\ k_a, & x > \theta \end{cases} \quad . \tag{8.2}$$

According to this step function, the synthesis rate is constant on $[0, \theta)$ as well as on (θ, ∞), which means that as long as the threshold is not crossed, the expression x follows the simple dynamics of constant synthesis and linear degradation. In fact, the intermediate value $r_{syn}(\theta) = (k_a + k_b)/2$ can be set to $r_{syn}(\theta) = k_a$ or to $r_{syn}(\theta) = k_b$, since this change in a single point is meaningless in the real system and has no effect on the dynamics.

CTMC of autoregulation The master equation model is based on the same kinetic functions for degradation and autoregulated synthesis as the ODE model. The only difference is that the kinetic functions define transition rates in the CTMC, instead of the derivative of expression levels. The state space is $\mathbb{N}_0$, counting the number X of molecules produced by the gene and present in the system. The degradation of the gene product reduces the number of molecules by one. The rate of this transition depends on the current number X of molecules and is given by $r_{deg}(X) = k_{deg}X$. Synthesis increases the number by one, the rate is given by the same Hill function as in the differential equation, i.e., $r_{syn}(X) = (k_a - k_b)\frac{X^h}{\theta^h + X^h} + k_b$.

Fluctuations around the equilibrium points As a consequence of the Hill kinetics, the synthesis rate is virtually constant apart from a small neighbourhood around the threshold. In particular, assuming that the equilibrium points are separated from the threshold, the molecule amount locally at an equilibrium point is described by a constant synthesis rate and linear degradation. Therefore, the stationary distribution of the CTMC is a Poisson distribution around this point of

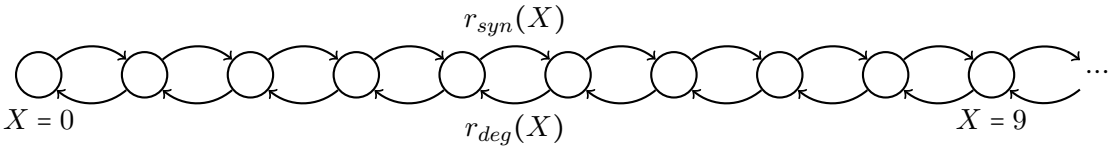


Figure 8.1: The CTMC of the chemical master equation model of synthesis and degradation of one species (e.g. a gene product) represented by a directed graph. The nodes represent the states $u \in U$, which are given by the amount X of molecules of the species. A transition with rate > 0 between two states is indicated by a directed edge. Assuming that the molecules are synthesised and degraded one by one, the only edges are between consecutive states X, $X + 1$. Formally, the CTMC is defined by its generator matrix G, which is in this case given by $g_{X,X+1} = r_{syn}(X)$ and $g_{X+1,X} = r_{deg}(X)$, $X = 1, 2, \ldots, m - 1$, where m is a model parameter implementing an upper bound on the molecule number (an unbounded CTMC with $m = \infty$ is possible as well). All other off-diagonal entries of G are 0 (and the diagonal is determined by the rest of the matrix as explained on page 17). This model is hence completely defined by the functions r_{syn}, r_{deg} and the bound m. Without autoregulation, r_{syn} is constant. The degradation is usually linearly dependent on the concentration, i.e., $r_{deg} = k_{deg}X$ where k_{deg} is a kinetic parameter.

equilibrium [Grimmett and Stirzaker, 1992, p. 251, the *Simple death with immigration* process with $\lambda = k_{syn}$, $\mu = k_{deg}$], see Fig. 8.2.

The thermodynamic limit Consider a biochemical process in a given volume and the corresponding master equation. Since the system is assumed to be well-mixed, the process stays the same, if we model only a part of the volume or if we multiply the whole system. In particular, the concentration and the collision probabilities stay the same. For the mathematical description this change in scale is changing the parameters, because they refer to absolute molecule numbers. Regarding the amount of molecules, scaling the volume by α also scales the number of molecules by α. This has to be taken into account by adjusting the parameters of the master equation which refer to the number of molecules. The threshold θ must be scaled by α, since it gives the number of molecules sufficient to trigger the autoregulation. Degradation was given as a function of the molecule number in the original volume. To take that into account we use $\alpha^{-1}k_{deg}$ as degradation parameter in the transformed model. The molecular dynamics in the CTMC are affected by the change in volume and molecule numbers. Naturally, the random fluctuations are diminished, if a larger volume with a higher number of molecules is considered. For the Poisson distribution around the point of equilibrium this effect is shown in Fig. 8.2.

In the so-called thermodynamic limit $\alpha \to \infty$, that is, scaling the volume and the molecule numbers equally while keeping the concentration constant, the fluctuations in concentration vanish and the dynamics of the master equation becomes deterministic and coincides with the ODE model [Oppenheim et al., 2003, Kurtz, 1978, Kurtz, 1972], which itself is not affected by changes in the volume, since it only deals with concentration values. The derivation of this result in [Kurtz, 1978] is in fact establishing the convergence of the stochastic process given by the CTMC to the mean trajectory of the CTMC and also to the solution of the corresponding ODE in the thermodynamic limit.

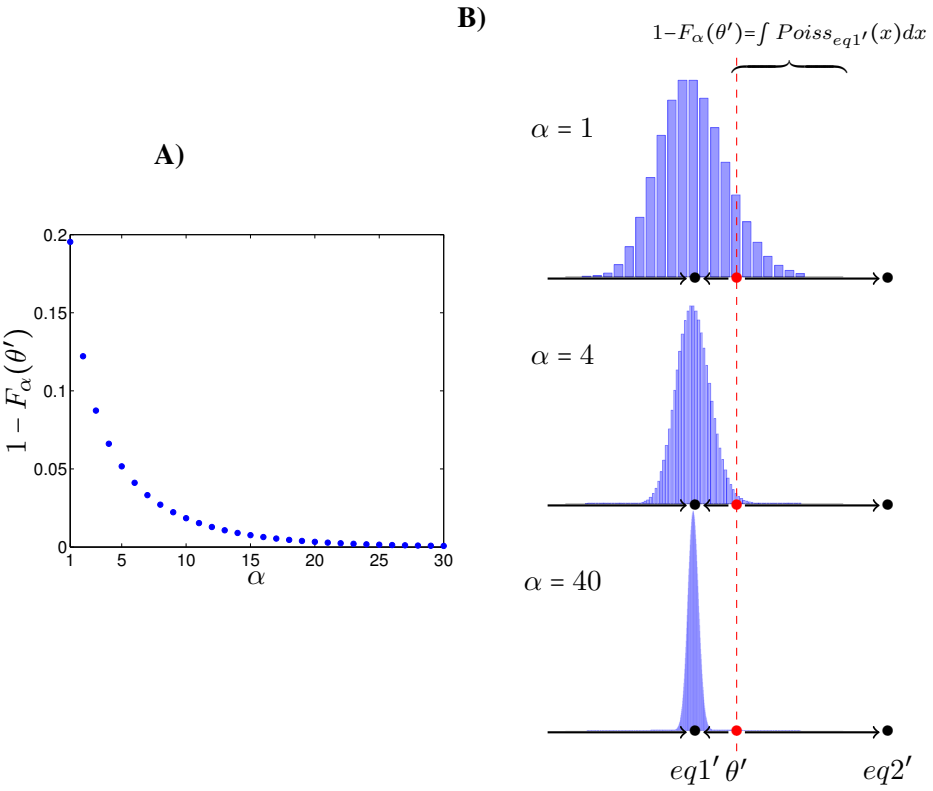


Figure 8.2: The Poisson distributed fluctuations around an equilibrium point $eq1 < \theta$ of a positively autoregulated gene are considered. The random fluctuations are only relevant for the dynamics if there is a significant probability that they cross the threshold. In this case, the random fluctuations would cause the component to change its activity level. Depending on the considered volume, the molecule numbers are different. Accordingly, the equilibrium point is $eq1' = \alpha \cdot eq1$, if the volume is scaled by α and the threshold is $\theta' = \alpha \cdot \theta$. The probability that the fluctuations cross the threshold is given by $P(Y' \geq \theta')$, where Y' is a Poisson distributed random variable with parameter $eq1'$, which means that $eq1'$ is the expectation of Y'. This probability can be obtained as $1-F_\alpha(\theta')$, where F_α is the cumulative distribution function of Y', the index indicating the dependence of Y' on α. **A)** The probability of crossing the threshold for increasing volume, $\alpha = 1, 2, \ldots, 30$. **B)** The Poisson distribution of the fluctuations, i.e., $f_{Y'} = Poiss_{(\alpha \cdot eq1)}$, is shown for $\alpha = 1, 4, 40$ on top of the phase diagram of the ODE (8.1). The probability $P(Y' \geq \theta')$ is the integral over $f_{Y'}$ above the threshold, which is exactly $1 - F_\alpha(\theta')$. This probability is significant if $\alpha = 1$, but with increasing α the fluctuations are considerably reduced in relative amplitude and the probability of crossing the threshold vanishes, as can clearly be seen in A).

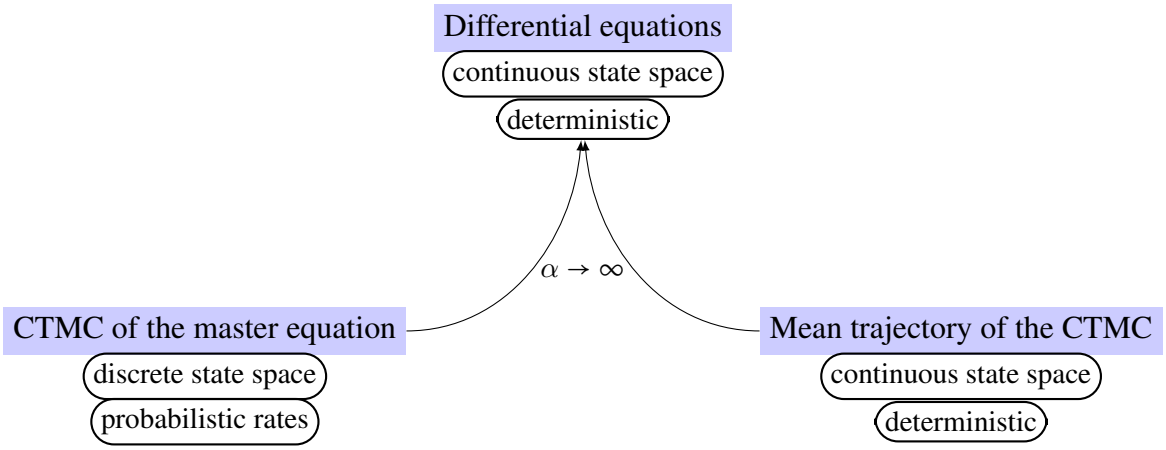


Figure 8.3: In the thermodynamic limit the random fluctuations on the molecular level vanish. As a consequence, the three different descriptions of the dynamics of the biological systems are becoming equal.

8.3 Introduction of stochastic logical networks

A logical network is completely defined by its state transition graph (STG) or, equivalently, by the update functions (cf. p. 145). Dynamic properties can be described in terms of trajectories or of sequences of states and can be verified with model checking. It is straightforward to extend a logical network to a CTMC by assigning an individual rate or velocity to the up- and down-regulation of each component. The resulting model will be called a stochastic logical network. Every edge in the STG inherits a transition rate from the update function. This enriches the dynamics in two ways: First, the transition rates define a probability measure on the space of all feasible sequences of states. Second, waiting times in every state of the sequence give a trajectory of the system in continuous time.

From update functions to the STG The dynamics of a logical network is nondeterministic because a state in the STG can have several successors and hence several possible updates. As already mentioned in Chap. 7, the STG is directly derived from the update functions. Every component j of a logical network has two update functions, Ψ^j_+, Ψ^j_-, as introduced in Sec. 7.5. They define, in terms of Boolean expressions, all states in which the component j is up regulated or down regulated, respectively. Let $i = 1, \ldots, n$ be the components of the logical network. The state space is then $\prod_{i=1}^n \mathbb{M}_{k(i)} = \prod_{i=1}^n \{0, 1, \ldots, k(i)\}$, where $k(i)$ is the maximal value of component i. The nodes of the STG are all states $u \in \prod_{i=1}^n \mathbb{M}_{k(i)}$. Let u, u' be two distinct nodes, then there is an edge from u to u' if and only if there is an $i \in \{1, \ldots, n\}$ and a sign $\sigma \in \{+, -\}$ for up- or down-regulation, such that $\Psi^i_\sigma(u) \equiv 1$ and the state u' fulfils $u'_j = u_j$ for all $j \neq i$, while $u'_i = u_i \, \sigma \, 1$. As a consequence, every edge in the STG corresponds to one distinct update function. If we assign a velocity or rate to the up- and down-regulation of the differ-

ent components, this gives a rate for each update function and as a consequence one rate for each edge in the STG. The STG thus defines a CTMC, which we will call a **stochastic logical network**.

In one state of the STG different update functions are competing to trigger first. The probabilities for each update to trigger define the embedded Markov chain, cf. p. 165. They are only determined by the ratios of the rates. The absolute time information which tells us how long the stochastic process rests in a given state is not contained in the embedded Markov chain. This additional attribute of the continuous time Markov chain can be used to distinguish slow and fast processes in the model.

Scope and justification of stochastic logical networks The abstract modelling approach of logical networks aims at analysing and reproducing the qualitative behaviour of a biological system. However, absolute quantitative aspects are beyond the scope of this approach. The paths through the STG are giving an ordering of the regulatory interactions, but the continuous time introduced in the extension cannot be expected to represent the exact time points of the modelled biochemical events. However, the specification of the rates of the different update functions is a way to incorporate relative information about the velocity of the regulation of the different components. For example, we can implement that gene A is much faster up regulated than gene B, a fact which might be crucial for the regulatory dynamics. In a similar fashion [Fauré et al., 2006] uses a priority order to relate different updates of a logical network, but this approach remains in the nondeterministic framework. Nondeterminism can be seen as the extreme form of uncertainty, where neither probabilistic nor any other information concerning the next event is given. In contrast, the priorities of [Fauré et al., 2006] introduce a fixed hierarchy, which means that the implied temporal order is assumed to be deterministic. At the same time, the choice among one hierarchy class is still nondeterministic. The stochastic logical networks do not enforce an order on the transitions. The probabilistic rates are only distinguishing likely dynamics from unlikely events.

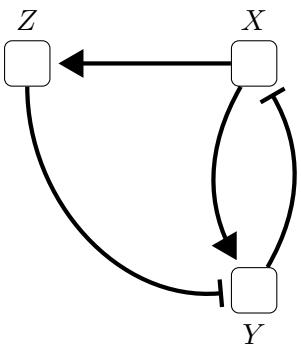
To conclude, the assignment of rates to the update functions is a possibility to incorporate more biological information about the system into the model. However, this is not a kinetic description based on a physical model of molecular interactions, as the ODEs or the master equation.

Definition 8.1. Let a logical network with components $i = 1, \dots, n$, a state space $U \subset \mathbb{N}_0^n$ and update functions $\Psi_i^+, \Psi_i^- : U \to \{0, 1\}$ be given. A **stochastic logical network** is obtained by assigning a fixed rate $r_\pm^i \in \mathbb{R}_{\geq 0}$ to each update function $\Psi_i^\pm$ for $i = 1, \dots, n$, $\pm \in \{-, +\}$.

Refinement of the rates The logical condition of an update function $\Psi_i^\pm$ gives the set of all states where the component i is up regulated resp. down regulated. The assignment of one rate to the update function assumes that the velocity of the up-regulation of i is independent of the current state u. To define the CTMC of the stochastic logical network, we could also assign individual rates $r_\pm^i(u)$ dependent on u. This is usually not practicable due to the size of the STG. However, it is possible to partition the set of states where Ψ_i^+ evaluates to 1 and to assign different rates (but the same update) to the individual parts. For example, we might implement a fast up-regulation with rate 2 for all states where component i is not expressed at all and a slower up-regulation with a rate of 1 if i has already an intermediate expression level. This would roughly mimic the typical saturation curve of gene expression.

Example The extension of a logical network to a stochastic logical network will be illustrated with a small example. In Fig. 8.4 the complete definition of the logical network is given. For every edge the corresponding update function is given. The STG is shown in Fig. 8.5. Assuming that up- and down-regulation of all components takes place with a similar velocity, we assign the same rate $r^i_\pm = r$ to $\Psi^\pm_i$ for all $i = 1, \ldots, n$ and $\pm \in \{+, -\}$. Since absolute time measures are clearly beyond the scope of this modelling approach, the actual value r of the rates is irrelevant and will be set to $r = 1$. The STG of this example network contains three distinct cycles and one fixed point. From every state there is a path leading to the fixed point. In particular, the system is never trapped in one of the cycles, but can only get trapped in the fixed point.

a)



b)

logical target functions		
component	target value	condition
X	0	$Y = 1$
	1	$Y = 0$
Y	0	$(X = 0) \vee (Z = 1)$
	1	$(X = 1) \wedge (Z = 0)$
Z	0	$X = 0$
	1	$X = 1$

c)

update functions			
up	down	condition	rate
Ψ^+_X		$(X = 0) \wedge (Y = 0)$	1
	Ψ^-_X	$(X = 1) \wedge (Y = 1)$	1
Ψ^+_Y		$(Y = 0) \wedge (X = 1) \wedge (Z = 0)$	1
	Ψ^-_Y	$(Y = 1) \wedge ((X = 0) \vee (Z = 1))$	1
Ψ^+_Z		$(Z = 0) \wedge (X = 1)$	1
	Ψ^-_Z	$(Z = 1) \wedge (X = 0)$	1

Figure 8.4: Illustrative example of a logical network. **a)** The interaction graph shows activating (normal arrowheads) and inhibiting (blunt arrowheads) interactions between three components. All components can have binary values. **b)** The target functions for each component describe the regulatory interactions in detail. **c)** The update functions for asynchronous update. They are derived from the target functions and define the STG shown in Fig. 8.5. The last column, with the rates for the update functions, defines the extension of the logical network to a stochastic logical network.

Dynamics of the stochastic logical network In the underlying logical network the nondeterministic dynamics are completely described by the paths in the STG or in other words, by all feasible sequences of states. The extension of a logical network to a stochastic logical network brings many new possibilities to analyse the dynamic properties in quantitative and probabilistic terms. The capabilities of probabilistic model checking of CTMCs were shortly introduced

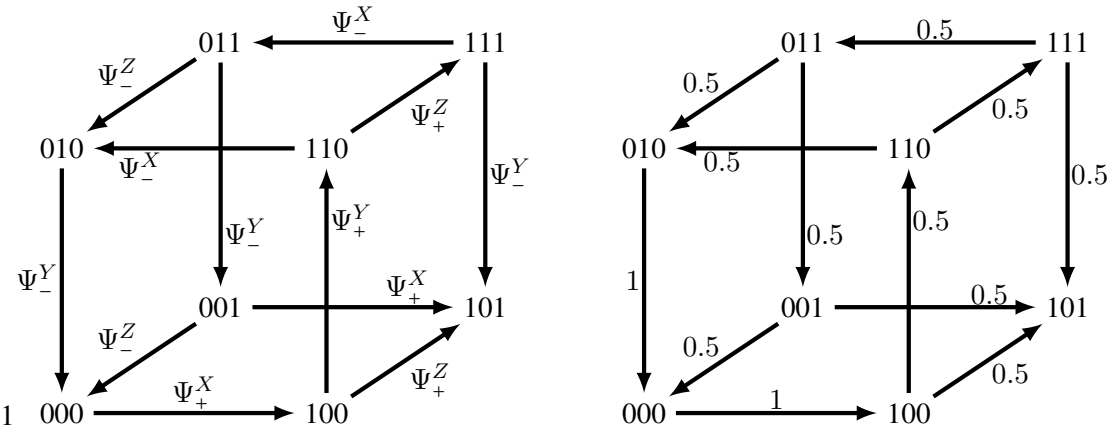


Figure 8.5: The STG of the network given in Fig. 8.4 is shown twice. On the left, every edge is labelled with the update function that defines this edge. All update functions are associated with the rate 1, thus defining a CTMC on the graph. The embedded Markov chain on the STG is given by the transition probabilities shown on the right. They can be derived by taking the ratio of the rate of the transition and the sum of all rates of outgoing edges, see page 165. The state 101 is a fixed point. Furthermore the state transition graph has several cycles: Cycle A: $000 \to 100 \to 110 \to 010 \to 000$ and Cycle B: $000 \to 110 \to 111 \to 011 \to 010 \to 000$ and Cycle C: $000 \to 100 \to 110 \to 111 \to 011 \to 001 \to 000$. However, the dynamic is never trapped in one of the cycles. Instead, from every state the fixed point 101 can be reached. For the probabilities this means that traversing a cycle once has a probability < 1. It is 0.5^2 for cycle A, 0.5^4 for cycle B and 0.5^5 for cycle C. As a consequence, the probability of traversing these cycles several times is getting extremely small. In contrast, the probability of being in the fixed point after only two transitions (starting in 000) is already 0.5.

on page 165 and will be illustrated on the example. Using the PRISM model checker we are able to compute probabilities for certain events and the expected time of different regulatory processes. Starting in the state 000 e.g., we can identify the probability that the trajectory will satisfy the following:

(i) component Z is never active
(ii) the state 010 is never reached
(iii) Y is activated before Z
(iv) it takes at least twice as much time until Z is activated as it takes for first activation of Y
(v) component Z is active for a larger ratio of time than Y.

Many interesting properties can be formulated in terms of sequences of states of the embedded MC, thus neglecting the waiting times in the states. Here the first three examples are properties which refer only to the sequence of states, while the last two properties take the continuous waiting time into account and are hence properties of trajectories of the CTMC of the stochastic logical network. In fact, (iv) is a specification of (iii), quantifying the time difference instead of just giving the order of the events. (i) is fulfilled by the infinite sequence cycling the four states with $Z = 0$. However, the probability of this infinite cycle is zero and every other sequence clearly contains a state with $Z = 1$. This example shows how the probabilistic view is enhancing the analysis compared to the nondeterministic perspective, which does not distinguish (i) from (ii), although it is just a matter of time until Z is activated and (i) not valid anymore, while (ii) is clearly satisfied by many paths and has probability > 0.62. (iii) is satisfied if and only if the

second transition is $100 \to 110$. This has probability 0.5, because the first transition from the initial state is fixed to $000 \to 100$.

Computing these probabilities by hand is only possible for such simple networks and properties of course. The continuous time properties (iv), (v) are already much more difficult to verify, but can easily be handled by the computational methods available with the PRISM model checker. (v) is an example of a long-term property which refers to the infinite time interval $[0, \infty)$ and can be directly verified if the stationary distribution is available.

8.3.1 The mean trajectory

As already mentioned, the PRISM model checker also allows computing the values of the mean trajectory at distinct time points. For stochastic logical networks we can therefore obtain the expectation of the expression levels of the components at a time point t. Identifying the expectation at a sufficient number of time points gives a good approximation of the average dynamics of the components in continuous time. Note that the resulting dynamics is completely continuous. While a component can only take values in the discrete set $\{0, 1, \ldots, k(i)\}$, the expectation can possibly take any value in $[0, k(i)]$. The expectation of a component i with maximal activation level $k(i)$ as a function of time is hence given by $E(i)\colon [0, \infty) \to [0, k(i)]$. For the network of Fig. 8.4 the mean trajectory, consisting of the expectations of the three components, is shown in Fig. 8.6.

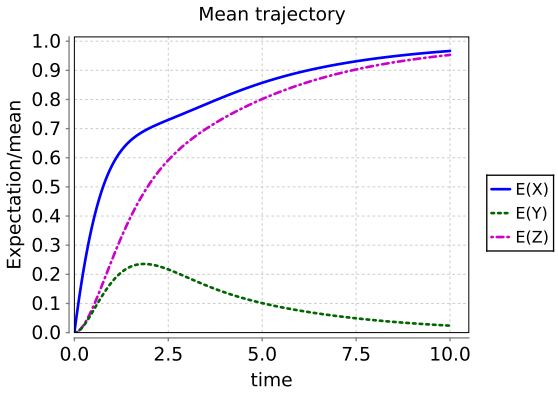


Figure 8.6: The mean trajectory of the stochastic logical network of Fig. 8.4. For the three components X, Y, Z the expectation value was computed at 500 equidistant time points from 0 to 10, which gives a good approximation of the functions $E(X), E(Y), E(Z)\colon [0, 10] \to [0, 1]$, i.e., the expectation values of the individual components at all time points. The convergence to the fixed point 101 is clearly visible. However, since arbitrarily long trajectories with alternating values of the three components are possible, the mean values are only slowly approaching the values of the fixed point.

8.3.2 Relationship between different models

Hill functions are naturally approximated by step functions. In fact, a step function is a logical function, as soon as we can define the preimages of the distinct values by logical formulas (as e.g. the step function (8.2) for autoregulation). Since a Hill function converges to such a step function, the logical network model of gene regulation can be seen as a discrete approximation

of the regulatory kinetics in ODE models, see [Thomas et al., 1995] and [Alon, 2007, p. 13]. This translation of the differentiable Hill functions into step functions, and hence into logical functions, can be reversed. In [Krumsiek et al., 2010] the Hill function was generalised in order to give a differentiable function that reproduces the switch-like behaviour of a given logical function. A comparison between different modelling formalisms is given in Fig. 8.7. Between the smooth Hill functions and the step functions, which can be interpreted as logical functions, also intermediate formalisms as e.g. piecewise linear functions can be considered. In [Jamshidi, 2012] a comprehensive treatment of the translation of a model between these different formalisms is given. The classical ODE format of a biochemical system is closely related to the stochastic model given by the master equation and the two descriptions coincide in the thermodynamic limit. It has to be noted that the master equation model is computationally not tractable for large volumes, because the state space, which is counting the molecule numbers, is exploding. In contrast, the stochastic logical networks are based on a minimal state space necessary to represent the relevant expression or concentration levels of the components. The stochasticity of the master equation constitutes itself in the fluctuations around the equilibrium values. These fluctuations in turn make the regulatory switches, caused by crossing the thresholds, random events. In stochastic logical networks, the regulatory switches are stochastic as well, but they are based on assigned probabilities and not on a kinetic model.

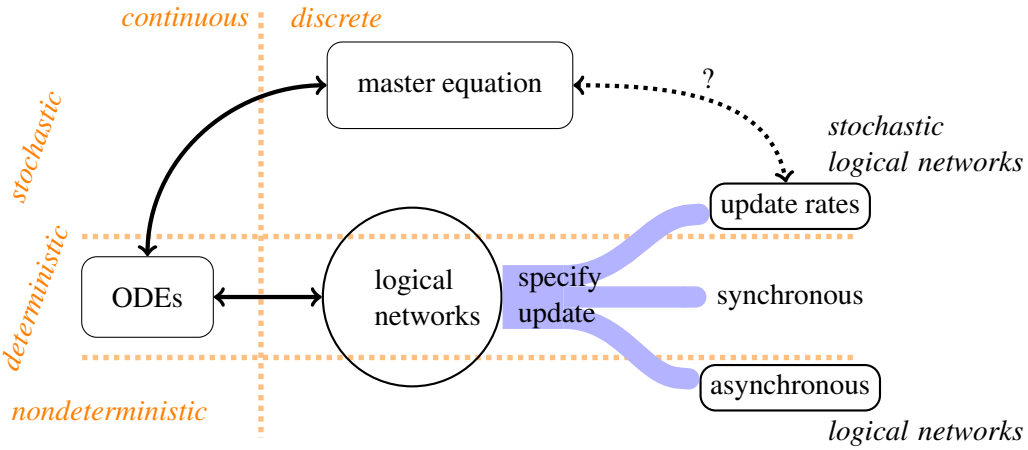


Figure 8.7: Different modelling frameworks for gene regulatory networks, signalling or metabolic networks. Depending on the kind of update, logical networks can have deterministic, nondeterministic or stochastic dynamics. Also the master equation model of a biochemical system is discrete and stochastic. The relationship between ODEs and logical descriptions of regulatory interactions was already subject to several studies and [Thomas et al., 1995, Krumsiek et al., 2010] give formal methods to get from one to the other, respectively.

8.3.3 Example of the feed-forward loop

A feed-forward loop (see Fig. 8.8) is a network motif presented in [Alon, 2007, Sec. 4.7,p. 57f]. It consists of three genes. The dynamical characteristic of this motif is that it produces pulse-like gene expression of one of the genes, see Fig. 8.9. This effect is achieved by a delayed repression of the gene.

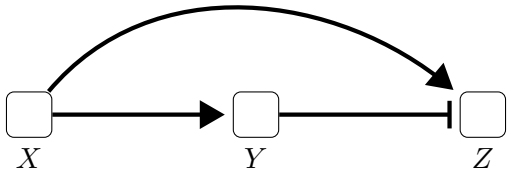


Figure 8.8: Feed-forward loop, a regulatory motif to generate a pulse-like expression of gene Z. Once gene X is active, it activates Z, but at the same time Y causes a delayed repression of Z.

In [Alon, 2007, Sec. 4.7.2] a kinetic model is given, consisting of the synthesis and degradation of the proteins Y and Z. In the following this model will be presented and compared to a stochastic logical network of the motif. The concentration of X is assumed to be constant and we thus have only the four processes:

$$\begin{aligned} 1: \ \varnothing \longrightarrow Y, \qquad & -1: \ Y \longrightarrow \varnothing, \\ 2: \ \varnothing \longrightarrow Z, \qquad & -2: \ Z \longrightarrow \varnothing \end{aligned}$$

The kinetic parameters are k_1 for synthesis of Y and k_{-1}, k_{-2} for degradation of Y and Z, respectively. For synthesis of Z the rate is k_{2a} if the gene is not repressed and k_{2b} if it is repressed by Y. The expression of Y has the equilibrium level k_1/k_{-1}. Gene Z has two equilibria, k_{2b}/k_{-2} in the repressed case and k_{2a}/k_{-2} in the unrepressed case. The threshold of the expression level of Y, above which the repression is effective, is denoted by θ. The resulting deterministic ODE model is given as follows [Alon, 2007, p. 60]:

$$\begin{aligned} \frac{\mathrm{d}Y}{\mathrm{d}t} &= k_1 - k_{-1} \cdot Y \\ \frac{\mathrm{d}Z}{\mathrm{d}t} &= (k_{2a} - k_{2b}) \cdot \left(1 + (Y/\theta)^h\right)^{-1} + k_{2b} - k_{-2} \cdot Z \end{aligned} \tag{8.3}$$

The first term of $\frac{\mathrm{d}Z}{\mathrm{d}t}$, namely $(k_{2a} - k_{2b}) \cdot (1 + (Y/\theta)^h)^{-1} + k_{2b}$ is the Hill function for the regulatory switch. In the limit $h \to \infty$ the Hill function approximates (except at θ, cf. p. 167) the step function $r_{synZ}(Y) = \begin{cases} k_{2a}, & Y \le \theta \\ k_{2b}, & Y > \theta \end{cases}$.

A logical network of the feed-forward loop It is straightforward to give the logical network that describes the feed-forward loop of Fig. 8.8. The gene that produces X is always active. To represent the relevant concentration levels of Y, two levels suffice. For the gene producing Z we have to distinguish between the basal expression level, if transcription is initiated by X but repressed by Y, and the upper expression level, in case transcription is activated by X and not repressed. Hence, three levels are necessary.

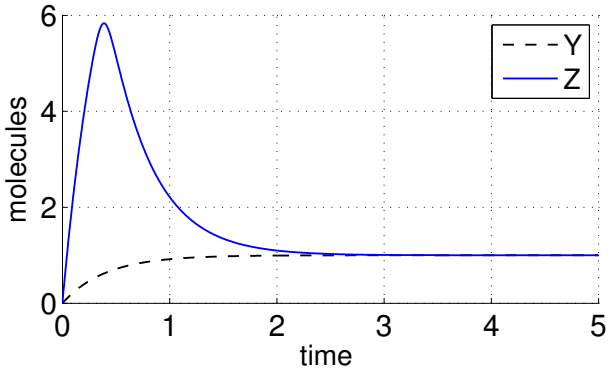


Figure 8.9: Pulse like gene activation of gene Z, generated by a feed-forward loop. The solution of the ODEs (8.3) is shown. Parameter values are $k_1 = 2.5$, $k_{2a} = 25$, $k_{2b} = 2.5$ and $k_{-1} = k_{-2} = 2.5$, threshold $\theta = 2.5/4 = 0.625$ and a Hill coefficient of $h = 20$. The initial values are $Y = Z = 0$. The switch-like regulation of Z, implemented by the Hill function above, leads to the pulse of Z expression before the concentration comes to rest in the lower equilibrium, k_{2b}/k_{-2} under repression by Y.

target functions		
component	target value	condition
Y	0	$X = 0$
	1	$X = 1$
Z	0	$X = 0$
	1	$(X = 1) \wedge (Y = 1)$
	2	$(X = 1) \wedge (Y = 0)$

update functions			
up	down	condition	rate
Ψ_Y^+		$(Y = 0) \wedge (X = 1)$	1
	Ψ_Y^-	$(Y = 1) \wedge (X = 0)$	1
Ψ_Z^+		$(Z = 0) \wedge (X = 1) \vee$ $(Z \leq 2) \wedge (X = 1) \wedge (Y = 0)$	10
	Ψ_Z^-	$(Z = 2) \wedge (Y = 1)$ $\vee (Z \geq 1) \wedge (X = 0)$	1

Table 8.1: The logical network of the feed-forward loop is given by the target functions. The asynchronous and unitary update is then derived and given by the update functions. Together with the rates in the last column this defines the stochastic logical network. The kinetic parameters of the ODE model were just scaled by 0.4, to get the rate 1 for degradation of Y and Z as well as for the repressed synthesis of Z and a rate of 10 for the unrepressed synthesis.

The logical networks of the feed-forward loop and its dynamics are fully described by Tab. 8.1. The formulation of the target functions is the straightforward implementation of the regulatory interactions. The four update functions $\Psi_Y^\pm$, $\Psi_Z^\pm$ describe the up- and down-regulation of the genes Y and Z. The extension to a stochastic logical network is simply given by assigning rates to all update functions. These rates define the velocity of the regulatory switch. It is natural to use the parameters for synthesis and degradation in the ODE model for the corresponding rates in the stochastic logical network (here these parameters were scaled by 0.4). Using the PRISM model checker [Hinton et al., 2006] we computed the mean trajectory of the network. The CTMC has infinitely many trajectories, but in fact only three different corresponding sequences of states, see Fig. 8.10. The mean trajectory is shown in Fig. 8.11 (A). The dynamics is in remarkable concordance with the solution of the deterministic ODE model, Fig. 8.9. However, it is not immediately clear if the single trajectories of the CTMC are close to the mean or not. The convergence of the mean of Z to $Z = 1$ in Fig. 8.11 (A) for example, could

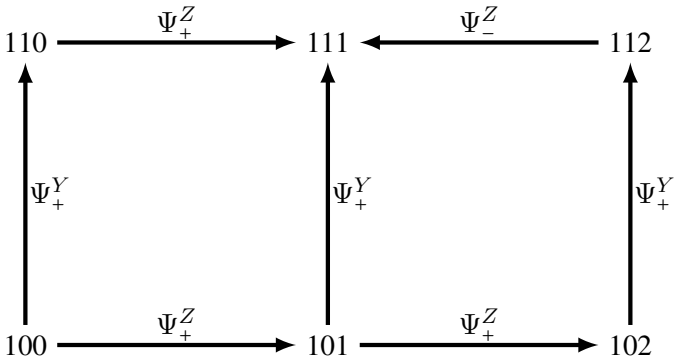


Figure 8.10: The STG of the logical network of the feed-forward loop. Component Z can take three values, $0, 1, 2$. Since X is a constant input and we are only interested in the case where X is active, states with $X = 0$ are neglected. From the initial state 100, only three possible sequences exist and they all lead to the fixed point 111. Every edge is labelled with the update function which defines the edge. The associated rates of the update functions define a CTMC on the STG.

also result from two sets of trajectories: One set with $Z = 2$ and another with $Z = 0$ in the long run. If these sets of trajectories both had the probability measure 0.5, the mean trajectory would converge to $Z = 1$. To ensure that the trajectories are really concentrated around the mean, the probability for a distinct expression level is computed and shown in Fig. 8.11 (B). These results confirm that almost all trajectories converge quickly to $Y = 1$, $Z = 1$ (in this simple model this can also be deduced by looking at the STG in Fig. 8.10).

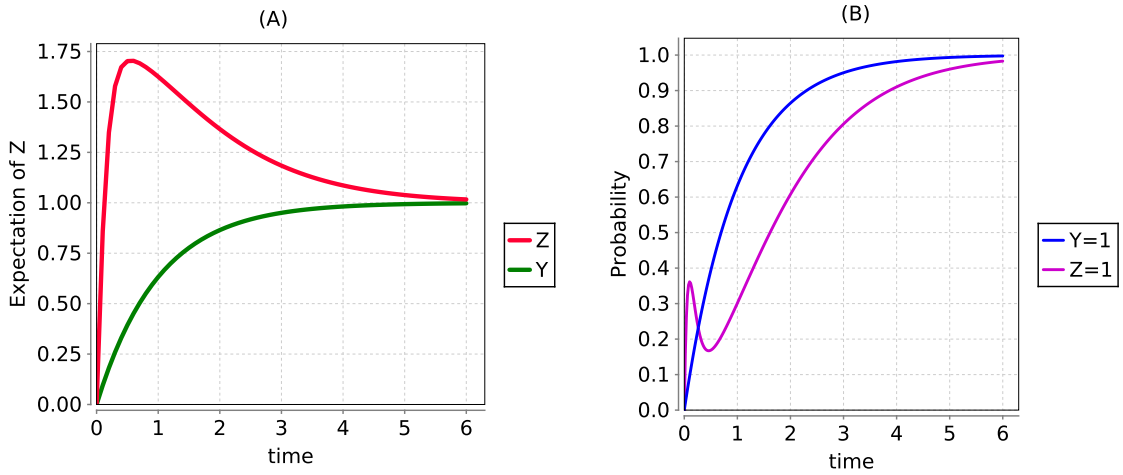


Figure 8.11: (A) The mean trajectory shows the expectation of the expression levels of the genes Y and Z. The characteristic pulse like dynamics of the feed-forward loop is clearly reproduced by the stochastic logical network. (B) To verify that the individual trajectories of the CTMC, i.e., the realisations of the stochastic process, are in fact concentrated around their mean, the probability of $Y = 1$ and of $Z = 1$ is shown. The mean trajectory is converging to $Z = Y = 1$ and the probability of $Y = 1$, $Z = 1$ is converging to 1, thus confirming that almost all individual trajectories are indeed reaching the state $Z = Y = 1$ in the time interval $[0, 6]$.

Conclusion It is important to note that, in contrast to the master equation, stochastic fluctuations around the equilibrium level are not included in the stochastic logical network model. Every equilibrium level is represented by one state and a change between different states can only be caused by a regulatory switch. That is, the corresponding logical condition must be fulfilled for a transition to occur and then the time point of the switch is random.

The mean trajectory of the CTMC of a stochastic logical network gives rise to a continuous dynamics. The concordance with the solution of the ODE model of the feed-forward loop is due to the short time scale and the simple regulatory structure, which results in a very limited set of possible dynamics. The time scale is short, because any trajectory ends in the fixed point $X = Y = Z = 1$ after few switches only. In case of more complex regulatory dynamics, the mean trajectory cannot be expected to reproduce the ODE evolution. In particular if e.g. two fixed points exist, such that the probability to reach one of them is 1 and each is reached with a significant probability, then the mean trajectory would converge to a convex combination of these two fixed points which might not correspond to any (reachable) state. In the long run, even trajectories corresponding to the same sequence of states can lead to a meaningless mean trajectory. The reason is that the trajectories of one sequence are similar at the beginning, but with progressing time they are significantly shifted. As a consequence, the characteristics of the sequence will be lost in the mean. For example, the mean trajectory of an oscillating component would not oscillate indefinitely, but converge to the mean of the oscillation.

Nevertheless, in case of network dynamics which end up quickly in one unique fixed point, as the feed-forward loop, the logical model can capture the quantitative characteristics of the dynamics without considering the kinetics of the interactions. Probabilistic model checking offers various possibilities to elucidate properties that are not captured by the mean trajectory (cf. Fig. 8.11) and hence to compensate for the mentioned shortcomings.

8.4 Including the master equation in a stochastic logical network

Negative autoregulation is a network motif that can make the regulation more robust to stochastic fluctuations. This kinetic effect cannot be captured with a logical model, but with a small master equation model which gives here a CTMC with 10 expression levels for the gene. After looking at this master equation model in isolation, this model will be combined with a stochastic logical network model of a galactose switch. Two genes in this regulatory network are modelled by the master equation. This allows examining the stabilising effect of the interplay of the autoregulation of these two genes. The expectation of the dynamic behaviour is computed for varying signal strength of galactose sensing. Despite its low complexity, the combined model demonstrate a quantitative stochastic effect which was also identified in a detailed model of high complexity.

8.4.1 Negative autoregulation

The kinetic modelling of autoregulation was shortly discussed on p. 166. While positive autoregulation establishes bistability, negative autoregulation promotes robustness in gene expres-

sion. The equilibrium point of expression is given by the threshold θ of negative autoregulation. This is derived from the kinetic model in [Alon, 2007, p. 31f], where it is further argued that θ is a biochemical parameter that varies much less from cell to cell than the production rate k_{syn} and this leads to less variation in the expression level between cells. Increased robustness in negatively autoregulated cells could also be shown experimentally [Becskei and Serrano, 2000]. Another effect of negative autoregulation is an acceleration of the response of an activated gene, which means that it takes a shorter time from transcription activation until the concentration of the gene product reaches its steady state. A fast response requires a high synthesis rate k_{syn} of the gene product. As a consequence, also the steady state concentration will be high, see [Alon, 2007] for details. This dependency can be resolved by negative autoregulation, because the response time is then determined by k_{syn} as before, but the steady state concentration is fixed by θ. While the first effect of negative autoregulation, the robustness, depends on the stochastic fluctuations in parameters, the second one, the response time, depends on the kinetics and can also be observed in an ODE model.

Master equation for negative autoregulation In Fig. 8.1 it was explained how a master equation model of a single gene product that is synthesized and degraded is translated into a CTMC. The states of the CTMC are given by X, the number of molecules of the gene product. This CTMC is completely defined by m, the upper bound of X, and two functions, r_{syn} and r_{deg}, which assign a rate of synthesis and degradation to each state X. In the case of negative autoregulation the synthesis rate is given by the Hill function $r_{syn}(X) = (k_a - k_b) \cdot (1 + (X/\theta)^h)^{-1} + k_b$ for $X < m$ and with $k_b < k_a$. The autoregulation threshold θ is defined as the concentration where repression reduces transcription by 50% [Alon, 2007, p. 32]. At the upper bound the synthesis rate is $r_{syn}(m) = 0$. For the discrete CTMC model we will use the step function that is obtained by taking the limit $h \to \infty$ (cf. p. 167). The bounded master equation model for a negatively autoregulated gene is then given by the rate functions

$$r_{syn}(X) = \begin{cases} k_a, \ X \leq \theta \\ k_b, \ m > X > \theta \\ 0, \ X = m \end{cases} \qquad \text{and} \qquad r_{deg}(X) = k_{deg} X, \tag{8.4}$$

where m is the bound on the molecule number. For comparison, we also look at a similar model without autoregulation, given by the rate functions

$$r_{syn}(X) = \begin{cases} k_{syn}, \ X < m \\ 0, \ X = m \end{cases} \qquad \text{and} \qquad r_{deg}(X) = k_{deg} X, \tag{8.5}$$

where the kinetic parameter k_{syn} determines the constant rate function. The basal synthesis rate k_b is set to zero. To observe the effect of fluctuations of the synthesis parameters, which can vary from cell to cell, k_a and k_{syn} are varied and the stationary distributions of the CTMCs are computed, see Fig. 8.12. To compute the stationary distributions it suffices to solve a system of linear equalities defined by the generator matrix, see [Grimmett and Stirzaker, 1992, p. 244, (20)*, additional linear equalities are necessary to ensure that the distribution has only nonnegative values and sums up to 1].

As the parameters are varied, the distributions and in particular the mean values of the autoregulated model are less shifting compared to the model without autoregulation. This result

is also obtained by analysing the corresponding ODE model [Alon, 2007, Sec. 4.7.2]. The results of the master equation models in Fig. 8.12 are furthermore showing that also for fixed k_a, the random fluctuations around the equilibrium are reduced, because we observe a higher concentration of the stationary distributions around the mean values (i.e. the equilibrium points) for the model with autoregulation (see also [Bundschuh et al., 2003] for a detailed discussion of the stochastic effects of negative autoreguation).

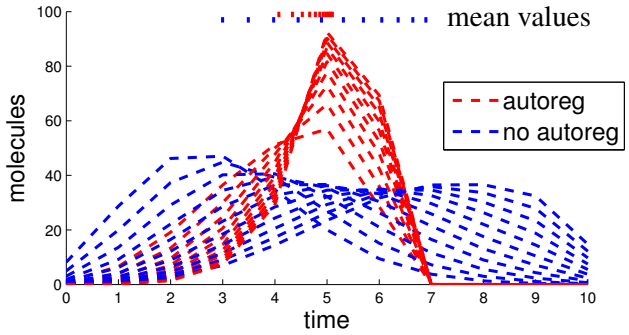


Figure 8.12: Stationary distributions of two different master equation models of synthesis and degradation of a single gene with varying parameters. In blue, the results for the model without autoregulation given by (8.5) with $m = 10$ are plotted. The constant synthesis rate k_{syn} was varied between 0.25 and 0.75. With autoregulation the CTMC is given by (8.4) with $m = 10, \theta = 5$ and $k_b = 0$. The unrepressed synthesis rate k_a was varied between 0.5 and 1.5 (in the negatively autoregulated case the synthesis rate is often higher to enable fast activation of the gene [Alon, 2007, Sec. 3.4]). The resulting stationary distributions are plotted in red. For all distributions, the mean values, which coincide with the equilibrium point $X = k_{syn}/k_{deg}$ resp. $X = k_a/k_{deg}$ [Grimmett and Stirzaker, 1992, p. 95,251], are indicated above the plots. Apparently, negative autoregulation leads to a higher density of the distribution around the mean, which means that for a fixed synthesis rate k_a the fluctuations in the concentration of the gene product are reduced. Furthermore, variations in k_a (among different cells) are shifting the distributions of autoregulated X much less than without autoregulation, which is also reflected in the mean values.

8.4.2 Logical modelling of galactose utilisation in yeast

Introduction Glucose is the main carbon source for yeast cells. However, if glucose is not available, the cells can use alternative substrates as carbon source. The budding yeast *Saccharomyces cerevisiae* can use galactose instead of glucose. The metabolic pathway to transform galactose to glucose-1-phosphate, named Leloir pathway, is catalysed by the enzymes of the GAL gene family. If glucose is not available but galactose is present in the medium, the GAL genes are activated on the transcriptional level by a network of activation and repression. This complex construction constitutes a robust switch that allows for quick activation of the galactose pathway. The complex regulation of the Leloir pathway was subject to extensive research on the experimental level as well as in the field of mathematical modelling. While [de Atauri et al., 2004] and [Berkhout et al., 2013] analyse quantitative and stochastic behaviour of an ODE model representing the core of the regulation of the Leloir pathway, a larger logical model of glucose repression in yeast was presented in [Christensen et al., 2009], where the Leloir pathway is only a minor part.

Regulatory mechanism of galactose sensing and uptake The Leloir pathway, consisting of the uptake of galactose, the conversion to galactose-1-phosphate and then to glucose-1-phosphate, is catalysed by the structural genes GAL1, GAL2, GAL7 and GAL10. Depending on the regulation, these structural genes (further on just abbreviated as GalS) can be in an activated state, which means that all enzymes for the Leloir pathway are produced. Otherwise they can be in a repressed state or in an unrepressed state that is ready for induction, i.e., production of the enzymes can be initiated immediately. These structural genes are transcriptionally activated by gal4p, the product of GAL4. This activation can be prohibited in several ways. *Repression:* In the presence of glucose, GAL4 is inactivated, such that no protein gal4p is produced and can bind to DNA to initiate transcription of GalS. *Inhibition:* gal4p is present and could bind to DNA for activation of the structural genes. However, activation is still inhibited by gal80p which binds to gal4p, so that it cannot activate the structural genes GalS. This is the case if no glucose and also no galactose is present and the cell is for example feeding on glycerol instead. Inhibition by gal80p can be relieved quickly by the regulatory protein gal3p, which then leads to an activation of the pathway. The mechanisms of the induction by gal3p are not completely understood at the moment. Whether the protein gal3p is the inducer itself or if it instead synthesizes an inducer from galactose, is not known. However, we can say that gal3p together with galactose leads to induction [Lohr et al., 1995, Johnston, 1987]. The difference between the glucose-repression and the gal80p-inhibition lies in the time it takes for activation of the structural genes. It takes 3-5 [h] if glucose was present as repressor and only 10-20 [min] if the activation was only inhibited by gal80p. This means that the cell can switch quickly (10-20 [min]) from glycerol to galactose as substrate, but needs a long time (3-5 [h]) to switch from glucose to galactose, [Johnston, 1987]. The transcription of GAL3 is activated, similar to the structural genes, by gal4p. Induction can only occur if galactose is present in the cell. The entry of galactose depends on gal2p and gal1p. In case the GAL structural genes are not expressed, galactose entry must still be possible on a very low scale, so that induction can take place. The activation of the structural genes will then lead to higher galactose uptake and hence reinforce their own expression. Once the Leloir pathway is active, gal3p is not necessary anymore, because gal1p can also mediate activation of the structural genes [Lohr et al., 1995], [Johnston, 1987].

Building an abstract model of the galactose regulation In the abstract logical network model of gene regulation, a gene is not distinguished from the protein it produces. The components of the regulatory network will be denoted $gal3$, $gal4$, $gal80$, $galS$, representing the genes as well as the corresponding proteins. Furthermore, regulatory input is given by the components $gal(ext)$, denoting external galactose, which is the signal for $gal3$ activation, and external glucose $glc(ext)$, which acts as repressor of $gal4$. The Leloir pathway can be integrated into the logical model as a metabolic network [Palinkas and Bockmayr, 2011]. The abstraction of the complex regulatory mechanisms to a discrete logical model is not unique. For example, it is a matter of interpretation if the components $gal80$ and $glc(ext)$ are acting together on the component $gal4$, inhibiting or activating it and gal4 is then the sole regulator of the structural genes $galS$, or if instead $glc(ext)$ and $gal80$ act together with $gal4$ on $galS$ directly. Both variants are a reasonable abstraction of the biochemical processes.

A crucial property of the galactose switch is that induction can be triggered much faster in the unrepressed (no glucose) and uninduced state (no galactose) than in the repressed state. The reason is that in absence of the repressor, $gal4$ can already bind to the promoter which

initiates transcription of the structural genes. It is just not possible to start transcription due to the inhibition by $gal80$. As soon as $gal80$ is released by $gal3$, transcription can be initiated. To reflect this in the abstract model, we let $glc(ext)$ regulate the state of $gal4$ and let $gal80$ be a regulator of $galS$. If glucose repression is released, the component $gal4$ can be activated, and the structural genes only thereafter. In the logical model dynamics, represented by the state transition graph (STG, see p. 170), this means that two transitions are needed for pathway activation, whereas one transition suffices from an uninduced state where the component $gal4$ is already active.

The inducer is in reality activated by internal galactose. However, the transport of galactose is only activated after induction. It is assumed that there is either a slow constitutive transport process or that basal GAL2 expression enables the first entry of galactose which can trigger induction [Lohr et al., 1995]. We could model this by implementing both transport processes as reactions of the metabolic network of the Leloir pathway, where one of them is permanently enabled. This is an example where the rates of the stochastic logical network can be used to incorporate an important biological aspect naturally in the model. In this case the rate of the two transport processes could be adjusted to implement the difference in velocity.

Another possibility is to use external instead of internal galactose concentration in the condition for the activation of $gal3$ (as inducer). To justify this twist, we look at the STG of the two variants. Since one transport reaction is always enabled, there is a transition from every state s with external galactose present to a state s' with internal galactose present. Such a transition represents the constitutive transport process that does not require $galS$ expression. From s' there is a transition to a state s'' where $gal3$ is active. Altogether we have from every state s in the STG where external galactose is present, a path of at most two transitions, which leads to a state s'', where $gal3$ is active. Such a path of length 2 is replaced by a single transition from s to s'', if we let $gal3$ be regulated by external galactose instead of internal. Also here the rates of a stochastic logical network can serve to reflect the process behind this single transition in a small rate.

A logical network model To give a logical network model of the GRN, it suffices to specify the regulation of each component by logical target functions as described in Sec. 7.2. In its simplest form, each component is represented by a binary (Boolean) variable, that is, each component can be in an active state (gene is expressed, signal is present) or in an inactive state. The scheme of the model in Fig. 8.13 is an incomplete representation of the logical model. The target functions given in Tab. 8.2 describe the regulation of each component in detail.

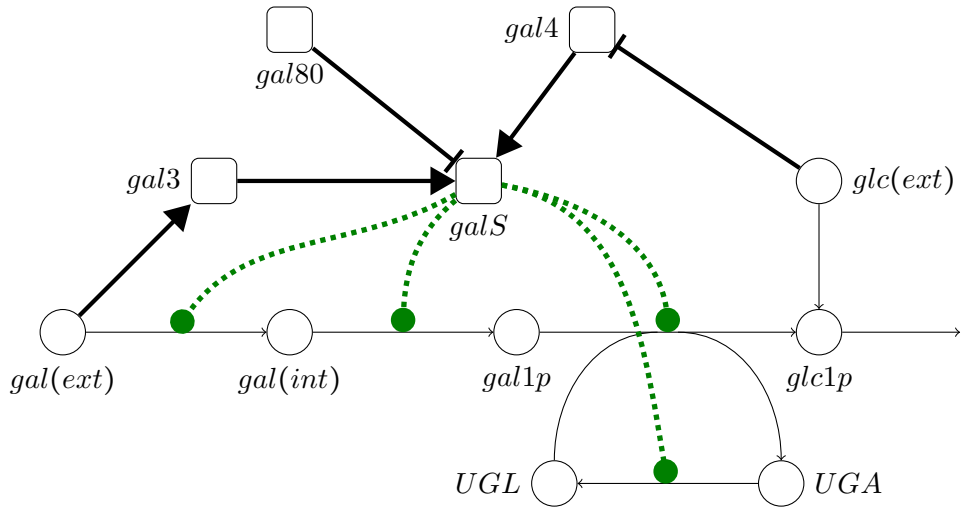


Figure 8.13: A graphical representation of the integrated model of the regulation of galactose utilisation. The genes are drawn as boxes and the metabolites as circles. The thick black arrows indicate the interactions of the GRN (blunt arrowhead for inhibition), the thin arrows represent the metabolic reactions and the dotted green arrows stand for the production of enzymes by the structural genes $galS$ which catalyse the pathway. Here we mainly discuss the dynamics of the GRN, but an integrated logical model of GRN and metabolism can also be given formally [Palinkas and Bockmayr, 2011].

target functions		
component	target value	condition
gal3	0	$gal(ext) = 0$
	1	$gal(ext) = 1$
gal4	0	$glc(ext) = 1$
	1	$glc(ext) = 0$
gal80	0	false
	1	true
	2	false
galS	0	$(gal4 = 0) \vee (gal80 = 2) \vee (gal3 = 0) \wedge (gal80 \geq 1)$
	1	$(gal4 = 1) \wedge (gal80 \leq 1) \wedge (gal3 = 1) \vee (gal4 = 1) \wedge (gal80 = 0)$

Table 8.2: Logical target functions that describe the interactions of the logical network of galactose utilisation given by the interaction graph in Fig. 8.13.

Together with the stoichiometry of the metabolic network, an integrated model of gene regulation and metabolism can be derived [Palinkas and Bockmayr, 2011]. The stoichiometry is already given by the scheme in Fig. 8.13, where all coefficients are ± 1. However, here we will focus on the GRN since the dynamics of the metabolic network is rather simple and completely determined by the three components $galS$, $gal(ext)$ and $glc(ext)$ of the GRN. To define the dynamics of the GRN, the update type of the logical network has to be specified (see Sec. 7.2 or page 170). Once all these specifications are given, the translation of the model into a Petri net representation as well as into a PRISM model of the corresponding stochastic logical net-

work can be done automatically by the procedure described in [Palinkas and Bockmayr, 2011]. This procedure generates in the first step the update functions. They are given in Tab. 8.3 for asynchronous and unitary updates. Since the gene $gal80$ is reported to be negatively autoregulated, thus allowing a short response time and keeping the expression to a moderate level (see Sec. 8.4.1), this component was implemented with three levels and a constant target function. This component is so far just a constant input into the GRN. However, the autoregulation will play a role in the next section. With this abstract model, the basic properties of the galac-

update functions		
up	down	condition
Ψ_+^{gal3}		$(gal3 = 0) \wedge (gal(ext) \geq 1)$
	Ψ_-^{gal3}	$(gal3 = 1) \wedge (gal(ext) = 0)$
Ψ_+^{gal4}		$(gal4 = 0) \wedge (glc(ext) = 0)$
	Ψ_-^{gal4}	$(gal4 = 1) \wedge (glc(ext) \geq 1)$
Ψ_+^{gal80}		$(gal80 = 0)$
	Ψ_-^{gal80}	$(gal80 = 2)$
Ψ_+^{galS}		$(galS = 0 \wedge gal4 = 1 \wedge gal80 = 0) \vee (galS = 0 \wedge gal4 = 1 \wedge gal3 = 1)$
	Ψ_-^{galS}	$(galS = 1 \wedge gal3 = 0 \wedge gal80 = 1) \vee (galS = 1 \wedge gal4 = 0)$

Table 8.3: From the target functions, Tab. 8.1, the update functions $\Psi_\pm^c$, describing asynchronous, unitary update of every component c of the GRN, can be derived.

tose switch, as described above, are preserved. The dynamics of this logical model is quickly summarized. Looking at the dynamics by initial state we get:

- In the initial state $glc(ext) = 0$, $gal(ext) = 0$, $gal80 = 1$, the structural genes $galS$ are repressed by $gal80$, but $gal4$ is present and can start transcription of $galS$ as soon as the repression is released, that is, as soon as $gal3$ is activated.
- In the initial state $glc(ext) = 1$, $gal(ext) = 0$, $gal80 = 1$, the structural genes $galS$ are repressed and in addition $gal4$ is not expressed. To activate $galS$ two regulatory events are hence required: activation of $gal3$ and of $gal4$.
- In the initial state $glc(ext) = 1$, $gal(ext) = 1$, $gal80 = 1$, the structural genes are not repressed by $gal80$, but for transcription, $gal4$ is missing. As soon as $gal4$ is activated, $galS$ will also be switched on.
- In the initial state $glc(ext) = 0$, $gal(ext) = 1$, $gal80 = 1$, galactose is the only substrate and $galS$ will be activated immediately.

The differences in the time it takes to release expression by $gal3$ activation and the time to initiate transcription by $gal4$ activation can be incorporated in a stochastic logical network of this model by adjusting the rates of $gal4$ up-regulation and $gal3$ up-regulation, i.e., of the update functions Ψ_+^{gal4}, Ψ_+^{gal3}.

8.4.3 A combined probabilistic model of logical and kinetic dynamics

The aim is now to combine the extension to the stochastic logical network with detailed kinetic modelling of two components, namely the components $gal3$ and $gal80$. Apart from the regulation by external galactose, the gene $gal3$ is also positively autoregulated. The interplay of

this positive autoregulation of $gal3$ and the negative autoregulation of $gal80$ is assumed to have a stabilising effect on the galactose switch [Johnston, 1987, de Atauri et al., 2004]. We will incorporate the two autoregulatory interactions by modelling the components $gal80$ and $gal3$ with the master equation in 10 expression levels.

Master equation models To define the master equation model of a single gene as a CTMC, it suffices to give rate functions for synthesis and degradation. The master equation model of negative autoregulation for the gene $gal80$ was already given by the rate functions in (8.4). For the gene $gal3$, the synthesis rate depends on the galactose signal $gal(ext)$. Furthermore, the positive autoregulation causes a switch in synthesis rate when the threshold θ^{gal3} of $gal3$ concentration is surpassed. This is implemented by a piecewise linear step function of the form

$$r_{syn}^{gal3}(gal3, gal(ext)) = \begin{cases} \max\left((gal(ext) - d), 0\right), & X \leq \theta^{gal3} \\ gal(ext) + d, & m > X > \theta^{gal3} \\ 0, & X \geq m \end{cases},$$

i.e., the synthesis rate is proportional to the signal $gal(ext)$ and shifted by $d > 0$ or by $-d < 0$ depending on the autoregulation. The degradation is modelled analogously to $gal80$, with parameter k_{deg}^{gal3}. For these two components the CTMCs of the master equation are thus completely described. The remaining components of the GRN are modelled with only two expression levels as a stochastic logical network. The complete CTMC of this combined model can be directly derived from Tab. 8.4 as follows. The components $\{gal3,\ gal80,\ gal4,\ galS,\ gal(ext),\ glc(ext)\}$ of the GRN model with their resolution in 10 resp. 2 expression levels give the state space $U = \{0, 1, \ldots, 9\}^2 \times \{0, 1\}^4$. If two states u, u' are identical except in one component c where $u'_c = u_c \pm 1$ and the condition listed in Tab. 8.4 for the update $c\pm$ (leftmost column) is fulfilled in the state u, then the transition from u to u' is occurring with the rate $g_{u,u'}$ (right columns). Different rates are given for the model with autoregulation and the model without autoregulation. All other off-diagonal entries $g_{u,u'}$ of the generator matrix of the CTMC are zero (cf. Fig. 8.1). Tab. 8.4 thus completely describes two different CTMCs on the same state space.

The impact of the autoregulation on the activation of the structural genes Since the activation of $galS$, and hence of the galactose pathway, depends on the signal of external galactose, $gal(ext)$, we will vary the strength of this signal in the model and observe the resulting dynamic behaviour. The interacting regulators $gal3$ and $gal80$ are modelled with the same range and the regulation of $galS$ can hence be deduced by comparing their expression levels directly. When the concentration of $gal80$ surpasses the concentration of $gal3$, we assume that not all of the inhibitor can be bound and the expression of $galS$ will be repressed. The update function for the activation of $galS$ is hence given by the expression $(galS = 0) \wedge (gal4 = 1) \wedge (gal3 \geq gal80 + 1)$. To make the effect of the autoregulation of the components $gal3$ and $gal80$ visible, the model is also considered without autoregulation. This means that the two components are modelled by the master equation with rate functions which do not depend on the concentration of the own product. These rate functions are also given in Tab. 8.4.

8.4.4 Computational analysis of the combined model

With these two variants of the combined model we will now examine the effect of the autoregulation of $gal80$ and $gal3$ on the sensitivity of the galactose switch to the signal strength.

update functions and rates			
update	condition	rate, with autoregulation	rate, no autoreg.
$gal3+$	$(gal3 \leq \theta^{gal3})$	$\max((gal(ext) - d), 0)$	$gal(ext)$
$gal3+$	$(gal3 < 9) \wedge (gal3 > \theta^{gal3})$	$gal(ext) + d$	
$gal3-$	$(gal3 > 0)$	$k_{deg}^{gal3} \cdot gal3$	$k_{deg}^{gal3} \cdot gal3$
$gal80+$	$(gal80 \geq \theta^{gal80}) \wedge (gal80 < 9)$	k_b^{gal80}	constant
$gal80+$	$(gal80 < \theta^{gal80})$	k_a^{gal80}	
$gal80-$	$(gal80 > 0)$	$k_{deg}^{gal80} \cdot gal80$	$k_{deg}^{gal80} \cdot gal80$
$gal4+$	$(gal4 = 0) \wedge (glc(ext) = 0)$	constant	constant
$gal4-$	$(gal4 = 1) \wedge (glc(ext) = 1)$	constant	constant
$galS+$	$(galS = 0) \wedge (gal4 = 1) \wedge (gal3 \geq gal80)$	constant	constant
$galS-$	$(galS = 1) \wedge (gal3 < gal80) \vee (galS = 1) \wedge (gal4 = 0)$	constant	constant

Table 8.4: The update functions for this probabilistic model are given in several parts, in order to represent the varying rates conveniently. In the leftmost column the component and direction of regulation is indicated. A condition defines a set of states where the corresponding update occurs with the rate given in the columns on the right. The components $gal80$ and $gal3$ are modelled with the master equation and expression levels from 0 to 9 representing the amounts of regulatory protein. The rate functions in the two columns on the right define the master equations for $gal80$ and $gal3$. The other components are modelled as a stochastic logical network with the rates given in the right columns as well. A simple model, where the autoregulation is neglected, as well as a model including the autoregulation of $gal80$ and $gal3$, are considered. With the rates for these two models, this table completely describes two CTMCs as explained in the main text above. These CTMCs were then analysed with the PRISM model checker, see Fig. 8.14.

Ideally, the genetic switch defines a threshold of the signal of external galactose and activates the pathway if and only if the threshold is surpassed. The activity of $galS$ in dependence of the signal $gal(ext)$ would then be a step function. Robustness means that despite the stochasticity in the regulatory interactions, the dynamics are close to this ideal case, i.e., there is a small probability of false activation when the signal $gal(ext)$ is below some threshold, and also of missing activation when the threshold is surpassed. To test this notion of robustness, the signal strength $gal(ext)$ was varied between 0 (no external galactose) and 4 (very strong galactose signal) and different indicators for pathway activation were computed. The parameters of the master equations of the components $gal3$ and $gal80$ were chosen such that the dynamics of the autoregulation is captured within the small range of 10 expression levels. For $gal3$, the exact values are $k_{deg}^{gal3} = 0.2$ and $d = 0.3$ and the threshold was set to $\theta^{gal3} = 4$. For the component $gal80$ we set the degradation rate to $k_{deg}^{gal80} = 0.2$ as well. The synthesis is given by $k_a^{gal80} = 2$ and $k_b^{gal80} = 0$. The threshold was set to $\theta^{gal80} = 5$. The rates for the logical components are constant and were set to 1. With these parameters the master equations represent $gal3$ and $gal80$ with 10 expression levels and model the dynamics of positive and negative autoregulation, respectively. To neglect the autoregulation in the master equations, we use a constant synthesis rate for $gal80$ and for $gal3$ a rate proportional to the signal $gal(ext)$. This is the kinetic correspondence to the

logical model, where autoregulation is neglected.

Comparison of the dynamics with and without autoregulation The basic dynamical properties, as listed above on p. 185, are of course preserved in the refined model, no matter if autoregulation was included in the synthesis rates of the master equations or not. The differences between these two variants are merely in the quantitative and stochastic properties of the switch. To examine these properties with and without autoregulation, four indicators were analysed with the PRISM model checker [Hinton et al., 2006]. The initial state in all computations is given by $glc(ext) = 0$, no external glucose, $gal80 = 5$, moderate abundance of repressor, $gal3 = gal4 = 0$ and the structural genes are inactive, $galS = 0$. We are interested in the activation of $galS$, since this tells us if the pathway for galactose utilisation is switched on. In Fig. 8.14 A) the expected

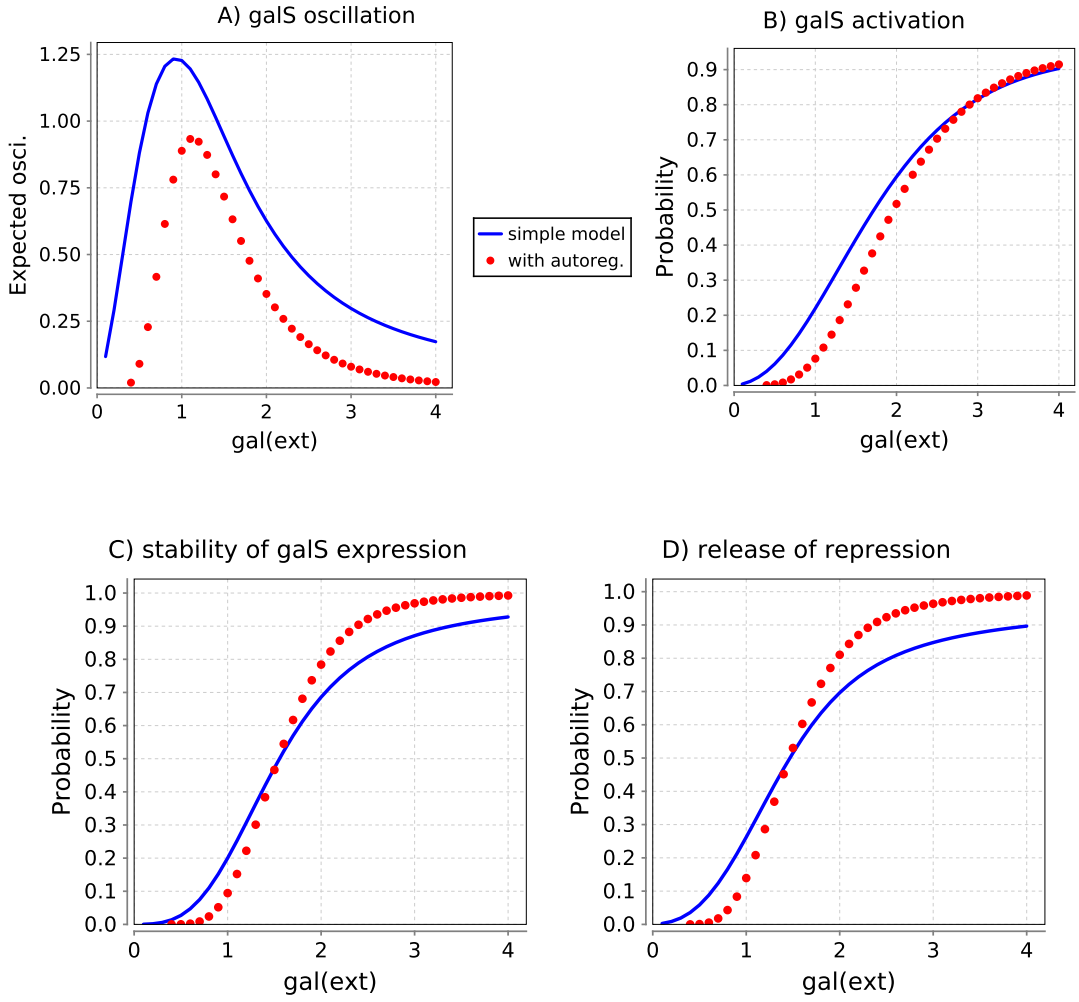


Figure 8.14: Indicators of stability of the galactose genetic switch are plotted for the models with and without autoregulation of $gal80$ and $gal3$. For the model with autoregulation, the curves of the indicators of $galS$ activation show then a more switch-like behaviour. Also the expected number of oscillations is at least 20% lower at all values of signal strength.

number of oscillations of the expression level of $galS$ was computed. Since we are starting in the initial state $galS = 0$, an oscillation is completed whenever the update Ψ_{-}^{galS} is executed. In the PRISM model checker we assign a reward of 1 to this update and have then the possibility to compute the expected number of executions of this update (i.e., the expected reward). An oscillation of the structural genes $galS$ indicates that the switch is unstable. It is clear that the instability will reach its maximum at an intermediate signal strength just on the threshold of pathway activation. The model with autoregulation has a similar distribution of the oscillations, but at any computed signal strength the amount of oscillations is at least 20% lower than in the model without autoregulation. The plot thus reveals a significantly increased stability of the switch, if $gal3$ and $gal80$ autoregulation is included in the model. A similar indicator is computed in Fig. 8.14 C), where the probability that $galS$ is permanently active during a fixed time interval shortly after induction is shown. The curve for the model with autoregulation is closer to a step function and this means closer to the ideal switch. False activation of $galS$ while the signal is low is less probable, but regular activation, following a strong signal, is more likely than in the model without autoregulation. Similar results were obtained for the two remaining indicators. The probability for quick first activation of $galS$ (within a short time interval after induction) was computed in Fig. 8.14 B) and the competition between $gal80$ and $gal3$, which determines $galS$ activation and was examined in Fig. 8.14 D), where the probability of $gal3 \geq gal80$, which means that all inhibitors are bound by $gal3$, is plotted.

Conclusion All computational results confirm that the model with autoreguation fulfils the condition of robustness, as formulated above, to a significantly higher degree than the model without autoregulation. The effect of the autoregulation of the two components $gal3$ and $gal80$ on the stability of the galactose switch was also analysed in [de Atauri et al., 2004] and an increase in stability was observed as well. The model used in [de Atauri et al., 2004] is based on differential equations and contains detailed and fully stochastic representations of the underlying regulatory mechanisms. This model is therefore of high complexity. The investigation here shows that it is possible to capture individual aspects of this biochemical system with a model of an overall low complexity. It is clear that such an approach cannot replace a fully mechanistic description as given in [de Atauri et al., 2004]. However, apart from the possibility to reproduce certain quantitative and stochastic behaviours, the development of simplified abstract models has a value on its own because it helps to identify key mechanisms that lead to certain dynamical features.

8.5 Discussion

The straightforward extension of a logical network as defined in Chap. 7 to a CTMC is termed stochastic logical network here. It should not be confused with probabilistic Boolean networks [Shmulevich et al., 2002] which are based on different assumptions and lead to discrete Markov chains. This chapter tried to grasp the relationship between the modelling approach of stochastic logical networks and the classical master equation of biochemical systems. On the one hand, they share the same mathematical format, namely CTMCs. On the other hand, they are based on different derivations. While the master equation is a mechanistic model based on considerations about the collision probabilities in a well-mixed reaction system, the logical models are often derived as phenomenological models. The contrary is the case in the relationship between ODEs and the master equation. Here the mathematical format is different, but the

ODE model as well as the master equation are based on very similar physical considerations. It turns out that the difference between these models lies essentially in the ”resolution” of the molecular amounts. The continuous concentration values in the ODE model can be seen as an infinite resolution, while the absolute molecule numbers are still present in the master equation. By taking the thermodynamic limit, the master equation reaches the infinite resolution of the ODE model and the stochastic fluctuations vanish. As a consequence, the two models coincide in the thermodynamic limit. It is clear that stochastic logical networks and the master equation are drastically different and we cannot expect to find a general coherence between them. Since the regulatory switches in probabilistic logical models are enabled by a fixed logical rule, the model is only stochastic in the time points of the switches. It accounts thus rather for cell to cell random variations of the kinetic parameters governing the regulatory interactions than for the fluctuations of expression levels and concentrations in a single cell. This chapter aimed to show that stochastic logical networks are able, despite their high abstraction level, to capture some features of regulatory networks, which usually require quantitative and stochastic models of much higher complexity to be reproduced. The rates on the update functions are also a possibility to include more biological information, in particular quantitative aspects, in the logical model.

The application to the galactose utilisation model gave an illustration of how the minimalistic logical modelling approach can be combined with the master equation. Due to the concordance in the mathematical format, this combination is straightforward and very flexible. It is hence well suited to model different parts of a biochemical system with different precision. Given an abstract logical model, we have the possibility to refine it step by step, incorporating kinetic description of critical interactions. In the galactose model, we integrated a master equation model of the autoregulation of two genes into the stochastic logical network. With this ’minimal’ extension, the model captured the stabilising effect of these features.

The mean trajectory of the stochastic logical network can reproduce the quantitative behaviour of a simple regulatory pattern. However, this holds only for short-term dynamics of networks with simple dynamical properties. In other cases, probabilistic model checking still enables us to verify many aspects of the short-term as well as of the long-term behaviour of the stochastic process and to complement the information given by the mean trajectory.

9 Outlook: Constraint-based modelling approaches for the interactions between gene regulation and metabolism

9.1 Overview

In models of gene regulation of metabolism, as they were discussed in the preceding chapters, the gene expression determines the concentration of enzymes which in turn control the flux rates. This control can be formalised by defining a maximal flux rate which depends linearly on the enzyme concentration. In constraint-based models the dependence on the enzymes can be represented by the bounds on the flux distributions. This allows the natural integration of transcriptional regulation into constraint-based models. In the other direction, the regulatory input from the metabolism to the gene regulatory network (GRN) is determined by the metabolite concentrations. It is not evident how to include this feedback regulation into a constraint-based model. A strategy which circumvents the problem is to deal only with relative values. Instead of absolute levels of gene expression or concentrations, only the information whether a gene is up- or down-regulated and whether the concentration of an enzyme or metabolite is increased or decreased is used as input or computed as output, respectively. This relative information only makes sense in a dynamic model, as e.g. the resource allocation model presented in Chap. 3. Two such approaches are presented in this chapter, both are working in the constraint-based modelling framework. The first approach uses the correlation between flux rates of two reactions to predict interactions in the GRN which are mediated by the metabolic network. This can be seen as the constraint-based version of a similar method based on metabolic control analysis. Second, an approach is proposed to deduce concentration changes from alterations in flux rates and enzymatic activity.

9.2 The flux space has more to give than single flux modes

An often criticised aspect of FBA and similar methods is the fact that the result is just a single flux mode. Particularly in cases where many different optimal solutions exist, a single flux mode gives only an arbitrarily chosen detail of the whole picture. Hence, it is a promising alternative to consider regions of the flux space instead. However, in high dimensions the representation and computation of such regions is a difficult computational task.

Limitations of optimisation-based methods The flux space is the set of all flux distributions that are feasible according to all biological aspects that are considered in the model, i.e., translated into constraints. Methods which are based on optimisation, as e.g. FBA, compute distinct elements of this set as optimal solutions, but the flux space itself, as a geometric object representing the capabilities of the metabolic network, is not at all explored. Many aspects of the biological model are thus neglected. As an example, consider the computation of flux variability (FVA, see Sec. 2.2.2) in a metabolic network with flux space F. For a reaction j this gives an interval $[slb_j, sub_j]$ of the feasible flux rates. If $c \in [slb_j, sub_j]$ we know that a flux mode v with $v_j = c$ exists, but we do not know if there are many flux modes of this kind or if this describes rather an exotic behaviour which is very unlikely in practice. Formally, we can consider the intersection $F \cap \{v \in \mathbb{R}^n : \|v_j - c\| \leq \varepsilon\}$ for some small $\varepsilon > 0$. If the intersection is small, we can conclude that a flux rate v_j close to c can only be obtained under narrow conditions on the flux rates of the other reaction in the network. This can be interpreted in the way that the flux rate $v_j = c$ is not robust, i.e., a small change in some part of the network is likely to force the flux rate of reaction j to deviate from this value. In fact, it is often the case in practice that only a part of the interval $[slb_j, sub_j]$ corresponds to intersections with significant volume and the rest of the interval represents marginal parts of the flux space, see e.g. Fig. 9.3. It is questionable whether these marginal parts can be considered as feasible flux rates.

Computing regions of flux modes that have certain properties, as e.g. optimality, was applied for different purposes [Almaas et al., 2004, Braunstein et al., 2008, Bordel et al., 2010, Schellenberger and Palsson, 2009, Price et al., 2004b]. The plus of information that these methods bring comes with a plus in computational complexity. A convenient and straightforward way to compute regions of flux modes with given properties and to obtain the volume of these regions is to sample the whole flux space. We can then sort for all sample flux modes which fulfil certain conditions. By counting samples, the volume can be determined relative to the volume of the whole flux space or relative to some superset with known volume. The difficult part here is to obtain a sufficiently large number of uniformly distributed samples of the flux space or of a subset.

9.2.1 Sampling

Curse of dimensionality It is easy to sample uniformly from a box $\prod_{i=1}^{n}[a_i, b_i] \subset \mathbb{R}^n$, since it suffices to sample every coordinate $x_i \in [a_i, b_i]$ uniformly. For most other regions $F \subset \mathbb{R}^n$ it is not possible to directly obtain uniformly distributed samples in a similar fashion. Instead, we can consider a box which contains F, and generate random samples from this box. Afterwards the samples that lie outside of F are discarded. This so-called rejection sampling is applicable if containment $x \in F$ can be tested quickly and the ratio of the volume of F to the volume of the box is not too small, so that a sufficient amount of samples remains. In particular, if the dimension of F is smaller than n, this method fails because $vol(F) = 0$ and the probability that a sample point lies in F is zero. For a flux space $F \subset \mathbb{R}^n$, containment can be easily tested by verifying the defining linear (in-)equality constraints $Sv = 0,\ lb \leq v \leq ub$. However, the second condition concerning the volume is not fulfilled in relevant cases. The problem here is a geometric phenomenon that arises in high dimensions. Let us look at two simple geometric objects, a cube and a ball, because we can explicitly compute their volume. For the n-dimensional unit ball the volume goes to zero as $n \to \infty$ (although it increases until $n = 5$). Independent of the dimension, the volume of the unit cube is 1 and it is the smallest cube that contains the unit

ball. Therefore the ratio of the volume of the ball and the smallest circumscribing box goes to zero as the dimension goes to infinity. Such phenomena arising in high dimensions and causing computational difficulties are sometimes called *curse of dimensionality* [Lee and Verleysen, 2007, p. 6]. When the flux space is fitted into a box, the same effect is encountered. First of all, the flux space $F := \{v \in \mathbb{R}^n : Sv = 0,\ lb \leq v \leq ub\}$ has dimension $h = n - \mathrm{rank}(S)$ and lives in the subspace $\mathrm{null}(S) = \{x \in \mathbb{R}^n : Sx = 0\} \subset \mathbb{R}^n$. Since $h < n$ in relevant cases, the volume of F as a subset of $\mathbb{R}^n$ is thus 0. This problem can be fixed by projecting $\mathrm{null}(S)$ onto $\mathbb{R}^h$. The projection of F has than dimension h (given that no reaction is blocked, i.e., $slb_j < sub_j$, $j = 1, \ldots, n$). The smallest circumscribing box in $\mathbb{R}^h$, containing the flux space F, can be given by FVA in the following way. Let $A \in \mathbb{R}^{h \times m}$ be the matrix that gives the projection of $\mathrm{null}(S)$ onto R^h, by $x \mapsto Ax$. To determine the minimal (maximal) value of the l-th component of the projection of F, the optimisation problem $a_l\,(b_l) := \min\,(\max)\{A_{l,}.x : x \in F\}$ is solved (where $A_{l,}.$ is the l-th row). The box is then given by $\prod_{l=1}^{h}[a_l, b_l] \subset \mathbb{R}^h$. The problem of dimensionality of the subset $F \subset \mathbb{R}^n$ is then fixed. However, for a large dimension h the problem that the volume of the projection of the flux space is vanishing compared to the volume of the box in $\mathbb{R}^h$ remains. In the core metabolic network introduced in Chap. 4 we have $n = 46$ and $h = 35$ and the ratio of the volume of the projection of the flux space to the volume of the smallest circumscribing box is so small that virtually no samples from the box are also elements of the flux space.

Random walks that converge to the uniform distribution An alternative method for sampling from a region $F \subset \mathbb{R}^n$ is the so-called Hit-and-run sampling [Kaufman and Smith, 1998]. This is basically a random walk on F. This method requires an oracle which can determine for a given point $x \in \mathbb{R}^n$ if $x \in F$ or not. Since the oracle is called many times during the computational procedure of Hit-and-run sampling it must be sufficiently fast. Given an initial element x^1, a random direction and a random distance in chosen to get the next element. The element given by this direction and distance is tested for containment by the oracle. If it is contained in F it is accepted as the next element x^2 and the iteration continues. Otherwise, another element is chosen randomly and tested until a next element $x^2 \in F$ is found.

The probability $P(x^i = y)$ describes a probability density on F and the goal of the method is that in the limit $i \to \infty$ this probability is constant for all $y \in F$, i.e., the uniform distribution is approximated [Lovász and Vempala, 2006]. In [Kaufman and Smith, 1998] a variation was proposed, where the choice of the direction is not uniform, but biased to directions where many samples were found previously. It is called Artificial Centering Hit-and-Run (ACHR). This modification is in particular countering the problem that classical Hit-and-Run sampling can get stuck in high dimensional corners [Lovász and Vempala, 2006] (the reason is again a *curse of dimensionality* phenomenon: in a corner of a cube for example, the proportion of directions pointing outside of the cube with dimension n is $1 - 0.5^n$.) While the classical Hit-and-Run sampling is rigorously proven to converge to the uniform distribution on a polytope, see [Lovász and Vempala, 2006], the ACHR method is not shown to converge. However, it was verified empirically that ACHR converges very fast to the uniform distribution, see [Kaufman and Smith, 1998], where the fast convergence of ACHR was shown experimentally for rectangular regions and simplices. For the analysis of metabolism, ACHR has been applied to flux spaces of various metabolic networks [Almaas et al., 2004, Schellenberger and Palsson, 2009, Yang et al., 2010] and is also implemented in the cobra toolbox [Schellenberger et al., 2011b]. An improved implementation of ACHR for flux cone sampling has been proposed recently [Megchelenbrink

et al., 2014]. Concerning futile cycles, sampling thermodynamically feasible flux modes is discussed in [Reimers, 2014], see also [Xie, 2012], where a variant of ACHR is proposed which avoids flux modes with futile cycles.

Correlation of reactions If a sufficient amount of samples of the flux space is available, this enables to easily analyse the correlation of sets of reactions [Reed and Palsson, 2004, Xi et al., 2009, Schellenberger et al., 2011b]. For pairs of reactions this can be seen as a relaxation of FCA (Sec. 2.2.2). Since the samples only give a random selection of elements, we cannot deduce a coupling relation with certainty. However, the projection of the samples on a reaction pair reveals the predominant relationship between the two reactions. This might show a strong relation which is not detected by FCA, see for example Fig. 9.3. Another relaxation of the notion of flux coupling (FCA), based on elementary modes, has been proposed in [Marashi, 2011].

9.3 Metabolism as part of the gene regulatory network

Constraint-based methods describe the metabolism only in terms of flux rates. Metabolite concentrations are assumed to be constant (steady-state assumption) and are not specified. Also the regulatory feedback from the metabolic network to the GRN must hence be derived from the flux rates. The straightforward approach used by the method *regulated FBA* [Covert et al., 2001] is to associate a high concentration of a metabolite with a high flux rate of the reactions that consume or produce it. This is based on simple mass action kinetics and can be used to implement the regulatory feedback in terms of flux rates. However, reaction kinetics is often much more complex and this simple correlation is in many cases not given. Another possible perspective on the interplay between a metabolic networks and the GRN is to identify interactions in the GRN that are mediated by the metabolic network. The idea is that regulatory input given by the expression of metabolic genes determines the state of the metabolic network and therewith also the regulatory feedback to the GRN. Altogether the metabolic genes thus have a regulatory impact on other genes, which is mediated by the metabolism. We will discuss methods that predict these metabolic mediated interactions in the GRN.

9.3.1 Determining metabolic mediated interactions between two genes

A method based on metabolic control analysis

In [Baldazzi et al., 2010] a method is presented to derive signed interactions (up- and down-regulating interactions) between regulatory components of a gene regulatory network, which are mediated by the metabolic network. The metabolic network is given by a kinetic ODE model. However, only the signs of certain partial derivatives must be given, while the specific values of the kinetic parameters are not required. A sophisticated computational procedure is then used to determine the propagation of the regulatory effects in the metabolic network and

to predict the regulatory feedback from the metabolism to the GRN. This procedure includes the symbolic inversion of a matrix containing partial derivatives. As a consequence, it is not possible to retrace how the results of this method are obtained even for minimal examples. The method is illustrated on a simple network which we will also discuss here. It is a simplified model of glycolysis, see Fig. 9.1. Hexose-6-phosphate (H6P) can be processed to PEP but can also bind to FruR which is thereby inactivated. This inactivation is represented by the reversible reaction 10 which describes the conversion of H6P and free FruR to bound FruR. Regulatory input into the GRN is only given by free FruR. Therefore, the flux v_{10}, being the only effector on the concentration of free FruR, determines the regulatory feedback from the metabolic network to the GRN. The method [Baldazzi et al., 2010] determines the signs (activation, inhibition, no effect) of interactions of the two genes, fbaA and pykF with free FruR, the regulatory component of the metabolic network. A kinetic model of the network is formulated with ODEs. The parameters are not exactly specified. To apply the method it suffices to specify the sign of the partial derivatives of flux rates. These signs just indicate if a regulatory interaction from a gene-product to a reaction is activating or inhibiting.

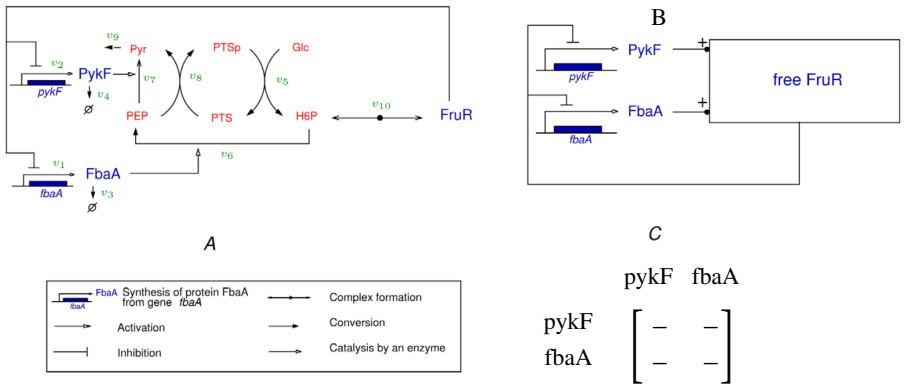


Figure 9.1: **A:** Schematic representation of a small example from [Baldazzi et al., 2010]. Genes and their products are distinguished by capitalising the name of the protein. Free FruR can bind with H6P to give 'FruR'. This process is given by the reversible reaction 10. The concentration of free FruR determines the activation of the genes pykF and fbaA. The stoichiometry of reaction 6 is 1·H6P → 2·PEP. **B:** From the dynamics in the metabolic network the regulatory effect of the proteins PykF and FbaA on the concentration of free FruR are determined by the method. **C:** The result is represented in a sign matrix of the interactions between the genes pykF and fbaA. They both have an inhibiting effect on themselves as well as on the other gene.

9.3.2 A method using the flux space of a constraint-based model

Based on the association of flux rates with metabolite concentrations and regulatory feedback to the GRN as in *regulated FBA* (rFBA) [Covert et al., 2001], we can use the flux space of a constraint-based model to find gene interactions that are mediated by the metabolic network. In rFBA, a regulatory interaction from a metabolite to the regulatory network becomes active, if the flux rate of the associated reaction is above a given threshold. The concentration of the metabolite is then assumed to be high enough to be effective in regulation.

Let i be a reaction which is regulated by gene A of the GRN and j a reaction which is associated with a metabolite that is a regulator of gene B of the GRN. To determine the regulatory effect that gene A exerts on gene B, we can identify the dependence of the flux rate of j on the flux rate of i. Flux coupling describes some cases of dependence qualitatively (see Def. 2.5). But also if a pair of reactions is uncoupled, there might be a strong correlation between the flux rates, which is revealed when the flux space is examined. This is done by projecting the flux space on the two reactions i and j under consideration. Formally the projection can be defined as

$$\pi_{ij}(F) := \{x \in \mathbb{R}^2 \,:\, \exists v \in F, \text{ s.t. } v_i = x_1,\ v_j = x_2\}.$$

The projection gives a 2-dimensional polytope, the region of all possible pairs of flux rates of i and j that can occur together in a flux mode of F. Some projections of very simple networks are given in Fig. 9.2.

For a given $x \in \pi_{ij}(F)$ consider the ε-ball $B_\varepsilon(y) := \{y \in \mathbb{R}^2 : \|y - x\| \leq \varepsilon\}$. Similar to the above considerations about the robustness of the flux rate of a single reaction, it is interesting here to know how many flux modes are projected to $B_\varepsilon(y)$. In practice this question can be answered by simply projecting a large number of samples with π_{ij}. The resulting sample concentration in the projection gives valuable additional information about the correlation of the two flux rates. In Fig. 9.3 top left e.g. the projected samples of the core metabolic network of Chap. 4 reveal that the directionally coupled reactions are effectively almost fully coupled, i.e., a flux through PGKf that is significantly higher than that of PDH is possible but extremely unlikely. Also in the other plots, large parts of the projection are almost empty. This means that flux modes with such combinations of flux rates are feasible but only if the flux rates of all other reactions lie in certain narrow ranges.

Exact computation of volumes of regions of a flux space is in general rarely tractable, but if a large number of uniformly distributed samples from the flux space is available, it can be well estimated [Braunstein et al., 2008].

Computation of flux space projection

The projection $\pi_{ij}(F) \subset \mathbb{R}^2$ can be equivalently written as an infinite union of 1-dimensional intervals, namely

$$\pi_{ij}(F) = \bigcup_{e \in [slb_i, sub_i]} \{e\} \times [a_j(e), b_j(e)],$$

where $a_j(e), b_j(e)$ are the minimal and maximal flux rates of reaction j, given that $v_i = e$, i.e.,

$$\begin{aligned} a_j(e) &= \min\{v_j \,:\, v \in F,\ v_i = e\} \\ b_j(e) &= \max\{v_j \,:\, v \in F,\ v_i = e\} \end{aligned} \tag{9.1}$$

To compute the projections in practice, we can take equidistant points $e_1, \ldots, e_q \in [slb_i, sub_i]$ and compute the intervals $\{e_k\} \times [vmin(e_k), vmax(e_k)]$ for $k = 1, \ldots, q$ by solving the LPs (9.1). An approximation of $\pi_{ij}(F)$ is then obtained by taking the convex hull of this finite set of 1-dimensional intervals.

Projection of sampled flux modes and regression analysis

Once uniformly distributed samples of flux modes $v \in F$ are available, the projection on the two reactions i, j is explicitly given by the corresponding vector components v_i, v_j. The distribution of these sample points in the projection of the flux space is adding important information.

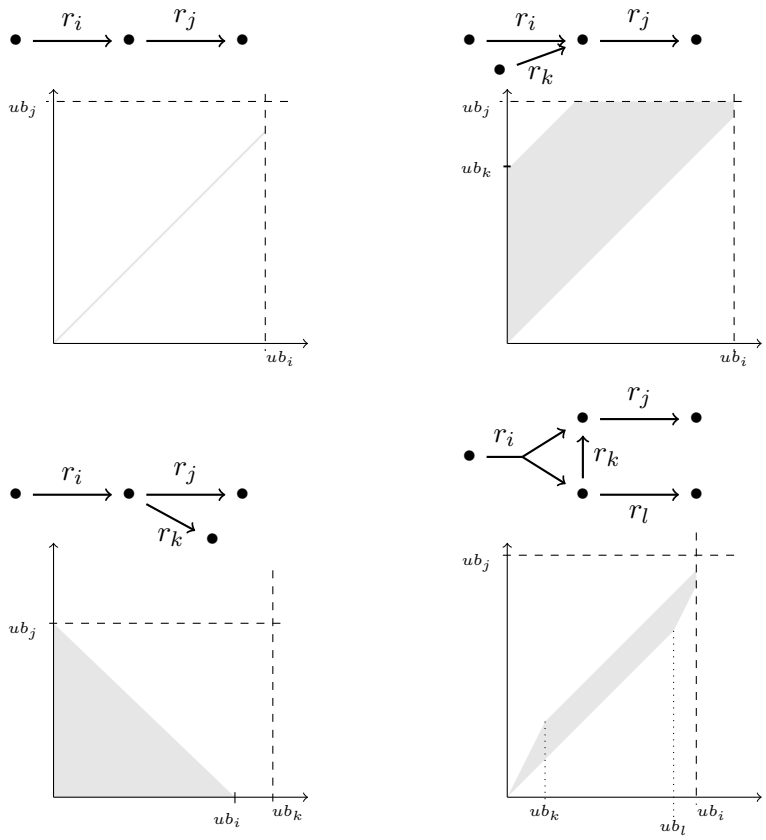


Figure 9.2: Projection on two reactions. The grey area indicates those points that are the projection of a flux mode of the depicted network. In very simple networks, as the four examples here, the projection can be determined exactly by hand. In case of two fully coupled reactions the projection is a line, see top left.

Fig. 9.3 illustrates the approach on the small core metabolic network that was introduced in Chap. 4. To derive a simple description of the dependence of reaction i on reaction j, a linear as well as a piecewise linear regression of the sample points in the projection is given. Let the projected samples be denoted by $s^1, \ldots, s^N$, $s^k = (x^k, y^k) \in \pi_{ij}(F) \subset \mathbb{R}^2$, $k = 1, \ldots, N$. The linear regression is a linear function $f\colon \mathbb{R} \to \mathbb{R}$, $f(x) = ax + b$, where the parameters a and b can be determined explicitly, see e.g. [Georgii, 2007]. To obtain piecewise linear regression, the interval $[slb_i, sub_i]$ of feasible flux rates of reaction i is partitioned into h intervals I_l, $l = 1, \ldots, h$, given by $I_l := [slb_i + \frac{l-1}{h}(sub_j - slb_j), slb_i + \frac{l}{h}(sub_j - slb_j)]$. The set of samples, $\{s^1, \ldots, s^N\}$ is partitioned accordingly into sets S^l, consisting of those samples $s^k = (x^k, y^k)$ with $x^k \in I_l$, $l = 1, \ldots, h$. A linear regression line is then computed for each of the h sets S^l. The resulting piecewise linear approximation of the sample points is drawn in red in Fig. 9.3 to capture nonlinear dependence between the reaction rates. A larger h gives a more detailed picture of the relationship between the two fluxes. However, in each interval the number of samples points must remain large enough to ensure the quality of the regression.

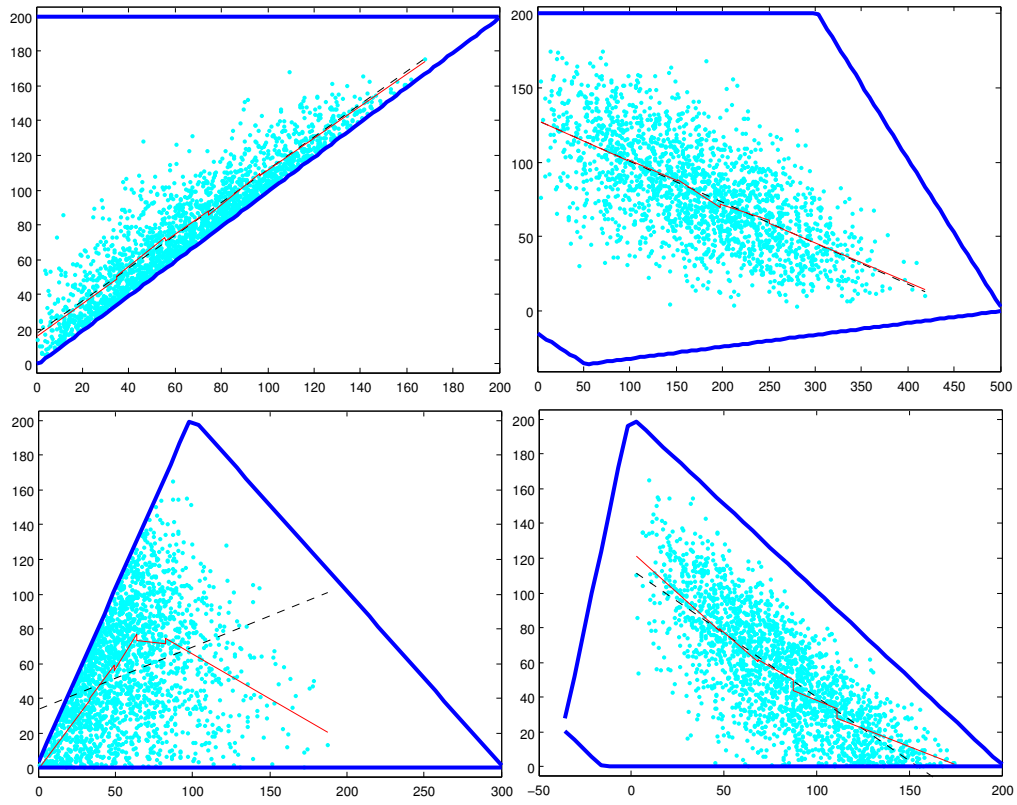


Figure 9.3: Illustrative examples of some projections of the linear flux space $\mathcal{F}$ of the core metabolic network of Chap. 4 without the resource allocation model. The dependence between reactions is shown for the following pairs of reactions from left to right and top to bottom: PGKf over PDH, TK2f over OxP, TCA over OxP2 and TCA over TK2f. The projections of 2000 sampled flux modes are plotted as dots. These samples were obtained with the ACHR-sampling method implemented in the COBRA toolbox [Schellenberger et al., 2011b] for Matlab. A linear regression line as well as a piecewise linear regression is computed to reveal the type of dependence between the two reactions. The projections of the flux space were computed with FVA at 40 equidistant flux rates of reaction i. It is given by its boundary plotted as a thick blue line. These examples show that the distribution of the samples gives important information that is not provided by flux coupling.

9.3.3 Comparison of the constraint-based and the metabolic control analysis approach

In Fig. 9.4 the results of the constraint-based approach to the network of [Baldazzi et al., 2010] are discussed. The projections of the two pairs of reactions of interest are shown. The effect of the flux through reaction 7 on the flux of reaction 10 is predicted to be negative, i.e., more FruR is released if the flux v_7 is increased. The same result was obtained in [Baldazzi et al., 2010] where a positive sign for the interaction from PykF (catalysing reaction 7) to the amount of free FruR was deduced by the method, see Fig. 9.1 B). On the other hand, the results differ for the effect of FbaA, i.e., the associated flux v_6, on the flux v_{10} determining the amount of free FruR. In the constraint-based model the effect can be positive or negative. In fact, the sign depends on the ratio of the fluxes v_7 and v_8. In the right plot of Fig. 9.4 we see that blocking reaction 6 also blocks reaction 10 and a high flux through reaction 6 enables in both directions a high flux through reaction 10. This effect is caused by the steady-state assumption and can be

explained as follows. Binding of FruR depends on H6P. On the one hand, reaction 6 produces PEP which is a substrate of reaction 8, and since reactions 8 and 5 are fully coupled, production of H6P depends on v_8. Therefore, a high flux v_6 is enabling a high production of H6P. On the other hand, reaction 6 is consuming H6P. Due to the stoichiometry of reaction 6 (H6P $\rightarrow$ 2$\cdot$ PEP) production of H6P can be twice as much as the consumption by v_6. Therefore, the H6P availability depends on the choice between reactions 7 and 8 for PEP consumption. If PEP is mostly consumed by v_7, then more H6P is consumed by v_6 than can be produced by v_5. In this case, the amount of unbound FruR increases. In the other direction, if v_8 is consuming more PEP than v_7, then more H6P is produced than consumed by v_6, and remaining H6P must bind to free FruR.

This possible positive effect on FruR binding by v_6 appears only in the constraint-based approach. It is not predicted in the metabolic control analysis used in [Baldazzi et al., 2010] where only a negative effect is predicted.

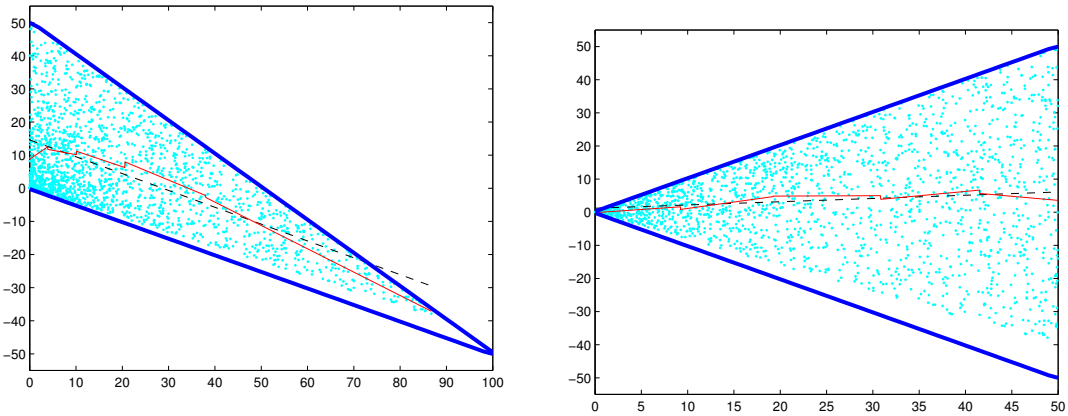


Figure 9.4: Results of the constraint-based approach. See Fig. 9.3 for explanation of the shown information. The plots show the projections on reaction 10 (FruR binding) against reaction 7 on the left and against reaction 6 on the right. In this small network the samples are distributed rather uniformly and are hence not adding new information to the projection. (As in Fig. 9.3, the COBRA toolbox [Schellenberger et al., 2011b] was used to generate samples.) These flux space projections show that a high flux through v_7 is forcing reaction v_{10} to go in the negative direction, i.e., the bounded FruR is released. If v_7 is close to zero, FruR is mostly being bound, i.e., $v_{10} \geq 0$. For reaction 6, the situation is different. A high flux v_6 enables a high flux v_{10} in both directions. If FruR is bound or released depends on whether reaction 7 or 8 is the main consumer of PEP as explained in the main text.

9.3.4 Discussion

The constraint-based approach is based on the same assumption as the method rFBA [Covert et al., 2001], namely that the flux rates are in correlation with the concentration of the reactants and the latter can hence be deduced from the flux rates. The approach of [Baldazzi et al., 2010] is based on a kinetic description of the network and on metabolic control analysis, see e.g. [Heinrich and Schuster, 1996, Sec. 5]. The kinetic parameters are neglected, instead the sign of certain partial derivatives must be given. In fact, these signs give a qualitative description of regulatory interactions between species and reactions. The case study of the small example

from [Baldazzi et al., 2010] revealed that the constraint-based approach does not always predict the same effect. But on the other hand, the results were also not contradicting. The difference was in fact that the constraint-based method gave an ambiguous result, where the method of [Baldazzi et al., 2010] predicted an interaction with negative sign. The question, which method is preferable for a given model, is not tackled here. The complex procedure of the method of [Baldazzi et al., 2010], involving symbolic inversion of a matrix, makes it hard to retrace the emergence of the final result from the individual features of the model. For the constraint-based approach this could be done by considering the implications of the steady-state assumption in the small network.

9.4 Relative description of concentrations in constraint-based models

Consideration of metabolic concentrations in constraint-based models is hindered by the circumstance that the flux space only specifies flux rates and neglects metabolite concentrations. In the case where the regulation of the metabolism by the GRN is modelled, this is not a big problem, since the regulatory control by enzymes can be naturally translated to bounds on flux rates as e.g. done in the method proposed in Chap. 3. The predicted dynamic behaviour computed by this method is given as a sequence of steady state phases. Each phase is characterised by a gene expression pattern and a flux distribution. In this section, an outlook is given, how such a dynamic picture of metabolic activity can be used to deduce some information about the corresponding dynamics of the metabolite concentrations. This information is only relative and qualitative. For a given metabolite we can predict, whether its concentration is increasing or decreasing or unchanged in the transition from one phase to the next. Based on these results, the feedback from the metabolic network to the GRN can then be modelled.

9.4.1 Reaction kinetics - an interplay of flux rate, enzyme activity and metabolite concentration

The kinetics that governs a reaction depends on the type of the reaction and the involved species, see [King and Altman, 1956] or [Beard and Qian, 2008, Sec. 4]. Essentially, the reaction rate is determined by the enzyme activity and the concentration of the reactants. The dependence of the enzyme activity is in general linear, but the dependence on the reactants is mostly nonlinear. However, we can say that the concentration of substrates has usually a positive effect on the reaction rate, while the concentration of a product of the reaction has a negative effect on the rate. Combining this rough and qualitative information about the effects of the enzyme concentrations E_j and the concentrations c_i for reactions $j = 1, \ldots, n$ and metabolites $i = 1, \ldots, m$, we will make some deductions in a relative and dynamic setting. Dynamic, because we consider transitions between two steady state phases as for example in the sequences of flux modes $v^1, \ldots, v^k$ in Chap 4. Relative, because we only refer to increase or decrease of a flux rate or a concentration of an enzyme or metabolite in the transition from one phase to the next.

We consider one fixed transition between two phases of flux distributions v^k and v^{k+1}. Formally, we distinguish the case that the flux of a reaction j changes the direction in the transition, i.e., $\text{sign}(v_j^k) = -\text{sign}(v_j^{k+1}) < 0$. In this case it is clear that, independent of the enzyme concentration, some substrates of j must have decreased their concentration or some products increased and vice versa, if $\text{sign}(v_j^k) = -\text{sign}(v_j^{k+1}) > 0$.

For the case that the flux did not change the direction, we define formally the changes in flux rates as well as in enzyme activity by

$$\begin{aligned} \Delta E_j &:= \text{sign}(E_j^{k+1} - E_j^k) \in \{-1, 0, 1\} \text{ and} \\ \Delta v_j &:= \text{sign}(|v_j^{k+1}| - |v_j^k|) \in \{-1, 0, 1\}. \end{aligned} \tag{9.2}$$

The fluxes v_j^{k+1} and v_j^k have the same sign and the definition does not distinguish whether these are positive or negative, because we assume that the enzyme activity bounds the flux rate equally in both directions. If the concentrations of the reactants are constant, the equivalence $E_j^{k+1} < E_j^k \Leftrightarrow v_j^{k+1} < v_j^k$ holds and this means that $\Delta E_j = \Delta v_j$. In the other direction, the question is what we can deduce about changes in metabolite concentration, given the changes in flux rate and enzyme activity. The prediction will be based on simple and evident rules. Looking at one reaction j, three cases can be distinguished:

- $\Delta E_j = \Delta v_j \Rightarrow$ no inference possible,
- $\Delta E_j < \Delta v_j \Rightarrow$ either some substrate must have *increased* its concentration or, if j is reversible, some product *decreased* its concentration
- $\Delta E_j > \Delta v_j \Rightarrow$ either some substrate must have *decreased* its concentration or, if j is reversible, some product *increased* its concentration

Such an approach was applied before to describe qualitatively, how a single gene perturbation can propagate regulatory changes in a GRN [Veber et al., 2008].

The conditions given by the second and third case can be formulated as linear constraints on integer variables $\Delta c_i \in \{-1, 0, 1\}$. These variables represent the possible changes of the concentrations, i.e., $\Delta c_i = \text{sign}(c^{k+1} - c_i^k)$, where c_i^k is the concentration of metabolite i in phase k for $i = 1, \ldots, m$. The set of reactions j where $\Delta E_j < \Delta v_j$ will be denoted $J^<$. In the index set $J^>$ we collect all reactions j which either change the direction of flux, i.e., $\text{sign}(v_j^k) = -\text{sign}(v_j^{k+1}) \neq 0$, or where $\Delta E_j > \Delta v_j$. When formalising the rules above, one has to take into account that the roles of substrates and products depend on the direction of the flux. We use the binary sign function given by $\text{sign}^+(v_j^k) = 1$ if $v_j^k \geq 0$ (i.e., zero counts as positive) and otherwise, for $v_j^k < 0$, $\text{sign}^+(v_j^k) = -1$. The conditions can then be formulated by the following constraints:

$$\begin{aligned} \text{sign}^+(v_j^k) \cdot \sum_i \text{sign}(s_{i,j}) \Delta c_i \leq -1, \text{ for } j \in J^< \\ \text{sign}^+(v_j^k) \cdot \sum_i \text{sign}(s_{i,j}) \Delta c_i \geq 1, \text{ for } j \in J^> \end{aligned}$$

The constraints for $j \in J^<$ ($j \in J^>$) are slightly stronger than the condition formulated in case two (three) above. The constraint for $j \in J^<$ states that the number of substrates that increased and products that decreased their concentration is strictly larger than the number of substrates that decreased and products that increased their concentration. While the implication of one single condition is not likely to restrict the possible concentration changes Δc significantly, the interplay of all constraints is likely to impose significant restrictions, see e.g. Fig. 9.5.

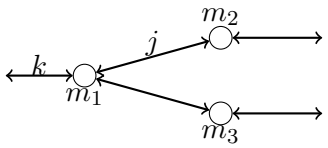


Figure 9.5: Example network to illustrate the deduction of changes in metabolite concentrations. The positive directions of the reactions are from left to right. In a transition between two metabolic phases, we suppose that no flux rate inverted its sign and that all fluxes go in the positive direction. Furthermore assume that $\Delta E_j > \Delta v_j$ and $\Delta E_k > \Delta v_k$. From $\Delta E_j > \Delta v_j$ we can conclude then that either m_1 decreases its concentration or m_2 increases. Since also $\Delta E_k > \Delta v_k$, we deduce that m_2 increased its concentration.

9.4.2 Implications on feedback

The goal of this analysis of possible concentration changes is to make deductions about the feedback to the GRN, which is determined by the concentration of certain regulatory metabolites. The constraints above define a space $H_k \subset \{-1, 0, 1\}^m$ of feasible concentration changes in the transition from phase k to phase $k + 1$. To infer predictions about the regulatory feedback, the space H_k has to be analysed. If i is a metabolite with regulatory impact in the GRN, then we want to know if the constraints restrict Δc_i to a subset of $\{-1, 0, 1\}$ or not. In case $\Delta c_i = 1$ is enforced, we can conclude that the regulatory feedback of metabolite i is increasing in the transition from phase k to phase $k + 1$. Similarly, the contrary effect can be concluded if $\Delta c_i = -1$. In case $\Delta c_i \geq 0$, we can at least exclude a decrease in the regulatory effect of metabolite i and similar an increase can be excluded if $\Delta c_i \leq 0$. In practice, the possible ranges of Δc_i for $i = 1, \ldots, m$ can easily be determined by solving integer linear programming (ILP) problems or with the help of Answer Set Programming [Lifschitz, 2002, Gebser et al., 2007].

9.5 Discussion

This last chapter gave an outlook on some approaches that can serve for modelling the regulatory feedback from the metabolic network to the gene regulatory network (GRN). Like in the previous chapters, the main interest here is on constraint-based methods. Regarding the approach of modelling the transcriptional regulation of metabolism, presented in Chap. 3, the last section here is of most interest, because it describes a possibility to fill the gap of feedback regulation within the constraint-based framework and is designed for application to sequences of flux modes. The approaches discussed in the Sec. 9.3 give a static description of interactions between genes, which are mediated by the metabolic network. This is useful if the focus of interest is on the gene regulation.

In contrast to the regulation of reaction rates by metabolic genes, which can be naturally translated into the constraint-based framework, the question of concentration(-changes) leads to quite intricate computational procedures (in [Baldazzi et al., 2010]) or to uncertain assumptions (in rFBA [Covert et al., 2001] and the approach presented here in Sec. 9.3). It is rather difficult to make sure that an appropriate description of the biological system under consideration is preserved. This is an essential weak point of constraint-based modelling compared to differential equations.

Bibliography

[Alberts et al., 1998] Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., and Watson, J. D. (1998). Molecular biology of the cell. *Garland, New York, Reference Edition*.

[Alfonsi et al., 2005] Alfonsi, A., Cancès, E., Turinici, G., Di Ventura, B., and Huisinga, W. (2005). Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems. In *ESAIM: Proceedings*, volume 14, pages 1–13. EDP Sciences.

[Almaas et al., 2004] Almaas, E., Kovacs, B., Vicsek, T., Oltvai, Z., and Barabási, A. (2004). Global organization of metabolic fluxes in the bacterium escherichia coli. *Nature*, 427(6977):839–843.

[Alon, 2007] Alon, U. (2007). *An introduction to systems biology: design principles of biological circuits*. CRC Press.

[Arkin et al., 1998] Arkin, A., Ross, J., and McAdams, H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells. *Genetics*, 149(4):1633–1648.

[Baier et al., 2003] Baier, C., Haverkort, B., Hermanns, H., and Katoen, J.-P. (2003). Model-checking algorithms for continuous-time markov chains. *Software Engineering, IEEE Transactions on*, 29(6):524–541.

[Baier and Katoen, 2008] Baier, C. and Katoen, J. (2008). *Principles of model checking*. MIT Press.

[Bailey, 1999] Bailey, J. (1999). Lessons from metabolic engineering for functional genomics and drug discovery. *Nature biotechnology*, 17(7):616–618.

[Baldazzi et al., 2010] Baldazzi, V., Ropers, D., Markowicz, Y., Kahn, D., Geiselmann, J., and De Jong, H. (2010). The carbon assimilation network in escherichia coli is densely connected and largely sign-determined by directions of metabolic fluxes. *PLoS computational biology*, 6(6):e1000812.

[Bartl et al., 2013] Bartl, M., Kötzing, M., Schuster, S., Li, P., and Kaleta, C. (2013). Dynamic optimization identifies optimal programmes for pathway regulation in prokaryotes. *Nature communications*, 4.

[Bartlett et al., 1988] Bartlett, K., Brayton, R., Hachtel, G., Jacoby, R., Morrison, C., Rudell, R., Sangiovanni-Vincentelli, A., and Wang, A. (1988). Multi-level logic minimization using implicit don't cares. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 7(6):723–740.

[Beard et al., 2002] Beard, D., Liang, S., and Qian, H. (2002). Energy balance for analysis of complex metabolic networks. *Biophysical journal*, 83(1):79–86.

[Beard and Qian, 2008] Beard, D. A. and Qian, H. (2008). *Chemical biophysics: quantitative analysis of cellular systems*. Cambridge University Press.

[Beaulieu and Xie, 2004] Beaulieu, N. and Xie, Q. (2004). An optimal lognormal approximation to lognormal sum distributions. *Vehicular Technology, IEEE Transactions on*, 53(2):479–489.

[Becskei and Serrano, 2000] Becskei, A. and Serrano, L. (2000). Engineering stability in gene networks by autoregulation. *Nature*, 405(6786):590–593.

[Beg et al., 2007] Beg, Q., Vazquez, A., Ernst, J., De Menezes, M., Bar-Joseph, Z., Barabási, A.-L., and Oltvai, Z. (2007). Intracellular crowding defines the mode and sequence of substrate uptake by escherichia coli and constrains its metabolic activity. *Proceedings of the National Academy of Sciences*, 104(31):12663–12668.

[Berkhout et al., 2013] Berkhout, J., Teusink, B., and Bruggeman, F. (2013). Gene network requirements for regulation of metabolic gene expression to a desired state. *Scientific reports*, 3.

[Bertsimas and Tsitsiklis, 1997] Bertsimas, D. and Tsitsiklis, J. (1997). *Introduction to linear optimization*. Athena Scientific Belmont.

[Bisschop, 2010] Bisschop, J. (2010). *AIMMS-optimization modeling*. Paragon Decision Technology, Haarlem.

[Bockmayr et al., 2001] Bockmayr, A., Weispfenning, V., et al. (2001). Solving numerical constraints. *Handbook of automated reasoning*, pages 751–842.

[Bordel et al., 2010] Bordel, S., Agren, R., and Nielsen, J. (2010). Sampling the solution space in genome-scale metabolic networks reveals transcriptional regulation in key enzymes. *PLoS computational biology*, 6(7):e1000859.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[Braunstein et al., 2008] Braunstein, A., Mulet, R., and Pagnani, A. (2008). Estimating the size of the solution space of metabolic networks. *BMC bioinformatics*, 9(1):240.

[Brayton et al., 1990] Brayton, R., Hachtel, G., and Sangiovanni-Vincentelli, A. (1990). Multilevel logic synthesis. *Proceedings of the IEEE*, 78(2):264–300.

[Brayton and Khatri, 1999] Brayton, R. K. and Khatri, S. P. (1999). Multi-valued logic synthesis. In *VLSI Design, International Conference on*, pages 196–196. IEEE Computer Society.

[Bundschuh et al., 2003] Bundschuh, R., Hayot, F., and Jayaprakash, C. (2003). The role of dimerization in noise reduction of simple genetic networks. *Journal of Theoretical Biology*, 220(2):261–269.

[Burgard et al., 2004] Burgard, A., Nikolaev, E., Schilling, C., and Maranas, C. (2004). Flux coupling analysis of genome-scale metabolic network reconstructions. *Genome research*, 14(2):301–312.

[Burgard et al., 2003] Burgard, A., Pharkya, P., and Maranas, C. (2003). Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and bioengineering*, 84(6):647–657.

[Calder et al., 2006] Calder, M., Vyshemirsky, V., Gilbert, D., and Orton, R. (2006). Analysis of signalling pathways using continuous time markov chains. In *Transactions on Computational Systems Biology VI*, pages 44–67. Springer.

[Chan and Ji, 2011] Chan, S. and Ji, P. (2011). Decomposing flux distributions into elementary flux modes in genome-scale metabolic networks. *Bioinformatics*, 27(16):2256–2262.

[Chaouiya et al., 2011] Chaouiya, C., Naldi, A., Remy, E., and Thieffry, D. (2011). Petri net representation of multi-valued logical regulatory graphs. *Natural Computing*, 10(2):727–750.

[Chaouiya et al., 2004] Chaouiya, C., Remy, E., Ruet, P., and Thieffry, D. (2004). Qualitative modelling of genetic networks: from logical regulatory graphs to standard petri nets. In *Applications and Theory of Petri Nets 2004*, pages 137–156. Springer.

[Christensen et al., 2009] Christensen, T., Oliveira, A., and Nielsen, J. (2009). Reconstruction and logical modeling of glucose repression signaling pathways in Saccharomyces cerevisiae. *BMC systems biology*, 3(1):7.

[Chu et al., 2011] Chu, D., Barnes, D. J., and von der Haar, T. (2011). The role of trna and ribosome competition in coupling the expression of different mrnas in saccharomyces cerevisiae. *Nucleic acids research*, 39(15):6705–6714.

[Chubukov et al., 2013] Chubukov, V., Uhr, M., Le Chat, L., Kleijn, R. J., Jules, M., Link, H., Aymerich, S., Stelling, J., and Sauer, U. (2013). Transcriptional regulation is insufficient to explain substrate-induced flux changes in bacillus subtilis. *Molecular systems biology*, 9(1).

[Chubukov et al., 2012] Chubukov, V., Zuleta, I. A., and Li, H. (2012). Regulatory architecture determines optimal regulation of gene expression in metabolic pathways. *Proceedings of the National Academy of Sciences*, 109(13):5127–5132.

[Clarke et al., 1999] Clarke, E. M., Grumberg, O., and Peled, D. (1999). *Model checking*. MIT press.

[Covert et al., 2001] Covert, M., Schilling, C., and Palsson, B. (2001). Regulation of gene expression in flux balance models of metabolism. *Journal of theoretical biology*, 213(1):73–88.

[Covert et al., 2008] Covert, M., Xiao, N., Chen, T., and Karr, J. (2008). Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli. *Bioinformatics*, 24(18):2044.

[Cox and Hammes, 1983] Cox, B. and Hammes, G. (1983). Steady-state kinetic study of fatty acid synthase from chicken liver. *Proceedings of the National Academy of Sciences*, 80(14):4233–4237.

[Crama and Hammer, 2011] Crama, Y. and Hammer, P. L. (2011). *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press.

[DasGupta, 2010] DasGupta, A. (2010). *Fundamentals of probability: a first course*. Springer.

[de Atauri et al., 2004] de Atauri, P., Orrell, D., Ramsey, S., and Bolouri, H. (2004). Evolution of design principles in biochemical networks. *Systems biology*, 1(1):28–40.

[De Figueiredo et al., 2009] De Figueiredo, L., Podhorski, A., Rubio, A., Kaleta, C., Beasley, J., Schuster, S., and Planes, F. (2009). Computing the shortest elementary flux modes in genome-scale metabolic networks. *Bioinformatics*, 25(23):3158–3165.

[De Jong, 2002] De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103.

[Dingel and Milenkovic, 2008] Dingel, J. and Milenkovic, O. (2008). A list-decoding approach for inferring the dynamics of gene regulatory networks. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 2282–2286. IEEE.

[Durot et al., 2009] Durot, M., Bourguignon, P.-Y., and Schachter, V. (2009). Genome-scale models of bacterial metabolism: reconstruction and applications. *FEMS microbiology reviews*, 33(1):164–190.

[Eaves and Freund, 1982] Eaves, B. C. and Freund, R. (1982). Optimal scaling of balls and polyhedra. *Mathematical Programming*, 23(1):138–147.

[Edwards et al., 2002a] Edwards, J., Covert, M., and Palsson, B. (2002a). Metabolic modelling of microbes: the flux-balance approach. *Environmental Microbiology*, 4(3):133–140.

[Edwards et al., 2002b] Edwards, J., Ramakrishna, R., and Palsson, B. (2002b). Characterizing the metabolic phenotype: a phenotype phase plane analysis. *Biotechnology and bioengineering*, 77(1):27–36.

[Elowitz et al., 2002] Elowitz, M., Levine, A., Siggia, E., and Swain, P. (2002). Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186.

[Fauré et al., 2006] Fauré, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):e124–e131.

[Feist and Palsson, 2010] Feist, A. and Palsson, B. (2010). The biomass objective function. *Current opinion in microbiology*, 13(3):344–349.

[Fell et al., 1986] Fell, D., Small, A., and Rankin, J. (1986). Fat synthesis in adipose tissue. an examination of stoichiometric constraints. *Biochem. J*, 238:781–786.

[Fristedt and Gray, 1997] Fristedt, B. and Gray, L. (1997). *A modern approach to probability theory*. Springer.

[Fukuda and Prodon, 1996] Fukuda, K. and Prodon, A. (1996). Double description method revisited. *Combinatorics and computer science*, pages 91–111.

[Garg et al., 2009] Garg, A., Mohanram, K., Di Cara, A., De Micheli, G., and Xenarios, I. (2009). Modeling stochasticity and robustness in gene regulatory networks. *Bioinformatics*, 25(12):i101–i109.

[Gebser et al., 2007] Gebser, M., Schaub, T., and Thiele, S. (2007). Gringo: A new grounder for answer set programming. In *Logic Programming and Nonmonotonic Reasoning*, pages 266–271. Springer.

[Genrich et al., 2001] Genrich, H., Küffner, R., and Voss, K. (2001). Executable Petri net models for the analysis of metabolic pathways. *International Journal on Software Tools for Technology Transfer (STTT)*, 3(4):394–404.

[Georgii, 2007] Georgii, H. (2007). *Stochastik: Einführung in die Wahrscheinlichkeitstheorie und Statistik*. Walter De Gruyter Inc.

[Gershenson, 2003] Gershenson, C. (2003). Classification of random boolean networks. *Artificial Life*, 8:1–8.

[Gillespie, 1976] Gillespie, D. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434.

[Gillespie, 1977] Gillespie, D. (1977). Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361.

[Glass and Kauffman, 1973] Glass, L. and Kauffman, S. (1973). The logical analysis of continuous, non-linear biochemical control networks. *Journal of theoretical Biology*, 39(1):103–129.

[Goelzer et al., 2011] Goelzer, A., Fromion, V., and Scorletti, G. (2011). Cell design in bacteria as a convex optimization problem. *Automatica*, 47(6):1210–1218.

[Goldstein and Bockmayr, 2013] Goldstein, Y. and Bockmayr, A. (2013). A lattice-theoretic framework for metabolic pathway analysis. In *Computational Methods in Systems Biology*, pages 178–191. Springer.

[Goodyear et al., 1991] Goodyear, L., Hirshman, M., Smith, R., and Horton, E. (1991). Glucose transporter number, activity, and isoform content in plasma membranes of red and white skeletal muscle. *American Journal of Physiology-Endocrinology And Metabolism*, 261(5):E556–E561.

[Griffiths and Harris, 2011] Griffiths, P. and Harris, J. (2011). *Principles of algebraic geometry*, volume 52. John Wiley & Sons.

[Grimmett and Stirzaker, 1992] Grimmett, G. and Stirzaker, D. (1992). *Probability and random processes*. Oxford University Press.

[Griva et al., 2009] Griva, I., Nash, S., and Sofer, A. (2009). *Linear and Nonlinear Optimization*. SIAM, 2nd edition.

[Grötschel et al., 1993] Grötschel, M., Lovász, L., and Schrijver, A. (1993). *Geometric algorithms and combinatorial optimization. 1988*. Springer, Berlin.

[Han et al., 2014] Han, J., Chen, H., Liang, J., Zhu, P., Yang, Z., and Lombardi, F. (2014). A stochastic computational approach for accurate and efficient reliability evaluation. *IEEE Trans. Computers*.

[Hegland et al., 2007] Hegland, M., Burden, C., Santoso, L., MacNamara, S., and Booth, H. (2007). A solver for the stochastic master equation applied to gene regulatory networks. *Journal of computational and applied mathematics*, 205(2):708–724.

[Heinrich and Schuster, 1996] Heinrich, R. and Schuster, S. (1996). *Regulation of cellular systems*. Chapman and Hall.

[Hertz and Dienel, 2005] Hertz, L. and Dienel, G. (2005). Lactate transport and transporters: general principles and functional roles in brain cells. *Journal of neuroscience research*, 79(1-2):11–18.

[Hinton et al., 2006] Hinton, A., Kwiatkowska, M., Norman, G., and Parker, D. (2006). PRISM: A tool for automatic verification of probabilistic systems. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 441–444.

[Hlavička and Fišer, 2001] Hlavička, J. and Fišer, P. (2001). Boom: a heuristic boolean minimizer. In *Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*, pages 439–442. IEEE Press.

[Hoffmann and Holzhütter, 2009] Hoffmann, S. and Holzhütter, H.-G. (2009). Uncovering metabolic objectives pursued by changes of enzyme levels. *Annals of the New York Academy of Sciences*, 1158(1):57–70.

[Hoffmann et al., 2006] Hoffmann, S., Hoppe, A., and Holzhütter, H.-G. (2006). Composition of metabolic flux distributions by functionally interpretable minimal flux modes (minmodes). *Genome informatics International Conference on Genome Informatics*, 17:195–207.

[Hoppe et al., 2007] Hoppe, A., Hoffmann, S., and Holzhütter, H.-G. (2007). Including metabolite concentrations into flux balance analysis: thermodynamic realizability as a constraint on flux distributions in metabolic networks. *BMC systems biology*, 1(1):23.

[Jamshidi, 2012] Jamshidi, S. (2012). *Comparing discrete, continuous and hybrid modelling approaches of gene regulatory networks*. PhD thesis, Freie Universität Berlin.

[Jensen et al., 2011] Jensen, P., Lutz, K., and Papin, J. (2011). Tiger: Toolbox for integrating genome-scale metabolic models, expression data, and transcriptional regulatory networks. *BMC systems biology*, 5(1):147.

[Johnson, 1975] Johnson, D. B. (1975). Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84.

[Johnston, 1987] Johnston, M. (1987). A model fungal gene regulatory mechanism: the GAL genes of Saccharomyces cerevisiae. *Microbiological reviews*, 51(4):458.

[Kærn et al., 2005] Kærn, M., Elston, T. C., Blake, W. J., and Collins, J. J. (2005). Stochasticity in gene expression: from theories to phenotypes. *Nature Reviews Genetics*, 6(6):451–464.

[Karlebach and Shamir, 2008] Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780.

[Karmarkar, 1984] Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM.

[Kaufman and Smith, 1998] Kaufman, D. and Smith, R. (1998). Direction choice for accelerated convergence in hit-and-run sampling. *Operations Research*, 46(1):84–95.

[Kim et al., 2002] Kim, S., Li, H., Dougherty, E. R., Cao, N., Chen, Y., Bittner, M., and Suh, E. B. (2002). Can markov chain models mimic biological regulation? *Journal of Biological Systems*, 10(04):337–357.

[King and Altman, 1956] King, E. and Altman, C. (1956). A schematic method of deriving the rate laws for enzyme-catalyzed reactions. *The Journal of physical chemistry*, 60(10):1375–1378.

[Klamt and Stelling, 2003] Klamt, S. and Stelling, J. (2003). Two approaches for metabolic pathway analysis? *TRENDS in Biotechnology*, 21(2):64–69.

[Klipp et al., 2002] Klipp, E., Heinrich, R., and Holzhütter, H.-G. (2002). Prediction of temporal gene expression. *European journal of biochemistry*, 269(22):5406–5413.

[Königsberger, 1997] Königsberger, K. (1997). *Analysis 2*. Springer.

[Königsberger, 2001] Königsberger, K. (2001). *Analysis 1*. Springer.

[Krumsiek et al., 2010] Krumsiek, J., Pölsterl, S., Wittmann, D. M., and Theis, F. (2010). Odefy-from discrete to continuous models. *BMC bioinformatics*, 11(1):233.

[Kümmel et al., 2006] Kümmel, A., Panke, S., and Heinemann, M. (2006). Putative regulatory sites unraveled by network-embedded thermodynamic analysis of metabolome data. *Molecular systems biology*, 2(1).

[Kurtz, 1972] Kurtz, T. (1972). The relationship between stochastic and deterministic models for chemical reactions. *The Journal of Chemical Physics*, 57(7):2976–2978.

[Kurtz, 1978] Kurtz, T. (1978). Strong approximation theorems for density dependent markov chains. *Stochastic Processes and Their Applications*, 6(3):223–240.

[Kwiatkowska et al., 2008] Kwiatkowska, M., Norman, G., and Parker, D. (2008). Using probabilistic model checking in systems biology. *ACM SIGMETRICS Performance Evaluation Review*, 35(4):14–21.

[Larhlimi and Bockmayr, 2006] Larhlimi, A. and Bockmayr, A. (2006). A new approach to flux coupling analysis of metabolic networks. In *Computational Life Sciences II*, pages 205–215. Springer.

[Larhlimi and Bockmayr, 2009] Larhlimi, A. and Bockmayr, A. (2009). A new constraint-based description of the steady-state flux cone of metabolic networks. *Discrete Applied Mathematics*, 157(10):2257–2266.

[Lavagno et al., 1990] Lavagno, L., Malik, S., Brayton, R., and Sangiovanni-Vincentelli, A. (1990). Mis-mv: optimization of multi-level logic with multiple-values inputs. In *Computer-Aided Design, 1990. ICCAD-90. Digest of Technical Papers., 1990 IEEE International Conference on*, pages 560–563. IEEE.

[Lawler, 1964] Lawler, E. (1964). An approach to multilevel boolean minimization. *Journal of the ACM (JACM)*, 11(3):283–295.

[Lee et al., 2008] Lee, J., Gianchandani, E., Eddy, J., and Papin, J. (2008). Dynamic analysis of integrated signaling, metabolic, and regulatory networks. *PLoS computational biology*, 4(5):e1000086.

[Lee and Verleysen, 2007] Lee, J. and Verleysen, M. (2007). *Nonlinear dimensionality reduction*. Springer.

[Levine and Hwa, 2007] Levine, E. and Hwa, T. (2007). Stochastic fluctuations in metabolic pathways. *Proceedings of the National Academy of Sciences*, 104(22):9224–9229.

[Liang and Han, 2012] Liang, J. and Han, J. (2012). Stochastic boolean networks: an efficient approach to modeling gene regulatory networks. *BMC systems biology*, 6(1):113.

[Lifschitz, 2002] Lifschitz, V. (2002). Answer set programming and plan generation. *Artificial Intelligence*, 138(1):39–54.

[Lohr et al., 1995] Lohr, D., Venkov, P., and Zlatanova, J. (1995). Transcriptional regulation in the yeast GAL gene family: a complex genetic network. *The FASEB Journal*, 9(9):777–787.

[Lovász and Vempala, 2006] Lovász, L. and Vempala, S. (2006). Hit-and-run from a corner. *SIAM Journal on Computing*, 35(4):985–1005.

[MacLean and Studholme, 2010] MacLean, D. and Studholme, D. (2010). A boolean model of the pseudomonas syringae hrp regulon predicts a tightly regulated system. *PloS ONE*, 5(2):e9101.

[Mahadevan et al., 2002] Mahadevan, R., Edwards, J., and Doyle III, F. (2002). Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophysical journal*, 83(3):1331–1340.

[Mahadevan and Schilling, 2003] Mahadevan, R. and Schilling, C. (2003). The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic engineering*, 5(4):264–276.

[Maier et al., 2009] Maier, T., Güell, M., and Serrano, L. (2009). Correlation of mRNA and protein in complex biological samples. *FEBS letters*, 583(24):3966–3973.

[Marashi, 2011] Marashi, S.-A. (2011). *Constraint-based Analysis of Substructures of Metabolic Networks*. PhD thesis, Freie Universität Berlin.

[McAdams and Arkin, 1998] McAdams, H. and Arkin, A. (1998). Simulation of prokaryotic genetic circuits. *Annual review of biophysics and biomolecular structure*, 27(1):199–224.

[McAdams and Shapiro, 1995] McAdams, H. and Shapiro, L. (1995). Circuit simulation of genetic networks. *Science*, 269(5224):650–656.

[McCluskey, 1956] McCluskey, E. J. (1956). A Way to Simplify Truth Functions. *Bell Syst. Tech. J.*, 35(6):1417–1444.

[Megchelenbrink et al., 2014] Megchelenbrink, W., Huynen, M., and Marchiori, E. (2014). optgpsampler: an improved tool for uniformly sampling the solution-space of genome-scale metabolic networks. *PLoS ONE*, 9(2): e86587.

[Megiddo, 1987] Megiddo, N. (1987). On the complexity of linear programming. *Advances in economic theory*, pages 225–268.

[Mendoza, 2006] Mendoza, L. (2006). A network model for the control of the differentiation process in Th cells. *Biosystems*, 84(2):101–114.

[Merle and Bourdon, 2010] Merle, T. and Bourdon, J. (2010). Complex update strategies for probabilistic boolean networks. *Proceedings of WSCB 2010*.

[Müller and Bockmayr, 2013] Müller, A. and Bockmayr, A. (2013). Fast thermodynamically constrained flux variability analysis. *Bioinformatics*, 29(7):903–909.

[Müller et al., 2014] Müller, S., Regensburger, G., and Steuer, R. (2014). Enzyme allocation problems in kinetic metabolic networks: Optimal solutions are elementary flux modes. *Journal of theoretical biology*, 347:182–190.

[Murata, 2002] Murata, T. (2002). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

[Murray, 1989] Murray, J. (1989). *Mathematical Biology*. Springer-Verlag, New York.

[Müssel et al., 2010] Müssel, C., Hopfensitz, M., and Kestler, H. (2010). BoolNet - an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, 26(10):1378–1380.

[Nagoshi et al., 2005] Nagoshi, E., Brown, S., Dibner, C., Kornmann, B., and Schibler, U. (2005). Circadian gene expression in cultured cells. *Methods in enzymology*, 393:543–557.

[Navid and Almaas, 2012] Navid, A. and Almaas, E. (2012). Genome-level transcription data of yersinia pestis analyzed with a new metabolic constraint-based approach. *BMC systems biology*, 6(1):150.

[Øksendal, 2007] Øksendal, B. (2007). *Stochastic differential equations*. Springer.

[Oppenheim et al., 2005] Oppenheim, A., Kobiler, O., Stavans, J., Court, D., and Adhya, S. (2005). Switches in bacteriophage lambda development. *Annu. Rev. Genet.*, 39:409–429.

[Oppenheim et al., 2003] Oppenheim, I., Shuler, K., and Weiss, G. (2003). Stochastic and deterministic formulation of chemical rate equations. *The Journal of Chemical Physics*, 50(1):460–466.

[Oyarzún et al., 2009] Oyarzún, D., Ingalls, B., Middleton, R., and Kalamatianos, D. (2009). Sequential activation of metabolic pathways: a dynamic optimization approach. *Bulletin of mathematical biology*, 71(8):1851–1872.

[Palinkas, 2011] Palinkas, A. (2011). *Integrierte Modelle von Stoffwechsel und Genregulation mit Petri-Netzen*. Diplomarbeit, Freie Universität Berlin.

[Palinkas and Bockmayr, 2011] Palinkas, A. and Bockmayr, A. (2011). Petri Nets for Integrated Models of Metabolic and Gene Regulatory Networks. *Workshop on Constraint Based Methods for Bioinformatics 2011*, page 45.

[Palinkas et al., 2015] Palinkas, A., Bulik, S., Bockmayr, A., and Holzhütter, H.-G. (2015). Sequential Metabolic Phases as a Means to Optimize Cellular Output in a Constant Environment. *PloS ONE*, 10(3).

[Paulraj and Sumathi, 2010] Paulraj, S. and Sumathi, P. (2010). A comparative study of redundant constraints identification methods in linear programming problems. *Mathematical Problems in Engineering*, 2010.

[Pfeiffer et al., 1999] Pfeiffer, T., Nu, J., Montero, F., Schuster, S., et al. (1999). Metatool: for studying metabolic networks. *Bioinformatics*, 15(3):251–257.

[Price et al., 2013] Price, M. N., Deutschbauer, A. M., Skerker, J. M., Wetmore, K. M., Ruths, T., Mar, J. S., Kuehl, J. V., Shao, W., and Arkin, A. P. (2013). Indirect and suboptimal control of gene expression is widespread in bacteria. *Molecular Systems Biology*, 9:660.

[Price et al., 2002] Price, N., Famili, I., Beard, D., and Palsson, B. (2002). Extreme pathways and kirchhoff's second law. *Biophysical journal*, 83(5):2879.

[Price et al., 2004a] Price, N., Reed, J., and Palsson, B. (2004a). Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature Reviews Microbiology*, 2(11):886–897.

[Price et al., 2004b] Price, N., Schellenberger, J., and Palsson, B. (2004b). Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies. *Biophysical Journal*, 87(4):2172.

[Puchałka and Kierzek, 2004] Puchałka, J. and Kierzek, A. M. (2004). Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophysical Journal*, 86(3):1357–1372.

[Qian and Beard, 2005] Qian, H. and Beard, D. (2005). Thermodynamics of stoichiometric biochemical networks in living systems far from equilibrium. *Biophysical chemistry*, 114(2):213–220.

[Quine, 1952] Quine, W. (1952). The problem of simplyfying truth functions. *Am. Math. Mon.*, 59(8):521–531.

[Quine, 1955] Quine, W. (1955). A way to simplify truth functions. *Am. Math. Mon.*, 62(9):627–631.

[Reed and Palsson, 2004] Reed, J. and Palsson, B. (2004). Genome-scale in silico models of e. coli have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states. *Genome Research*, 14(9):1797–1805.

[Reed et al., 2003] Reed, J., Vo, T., Schilling, C., and Palsson, B. (2003). An expanded genome-scale model of escherichia coli k-12 (ijr904 gsm/gpr). *Genome Biol*, 4(9):R54.

[Reimers, 2014] Reimers, A. C. (2014). *Metabolic Networks, Thermodynamic Constraints, and Matroid Theory*. PhD thesis, Freie Universität Berlin.

[Richard, 2010] Richard, A. (2010). Negative circuits and sustained oscillations in asynchronous automata networks. *Advances in Applied Mathematics*, 44(4):378–392.

[Rockafellar, 1997] Rockafellar, R. T. (1997). *Convex analysis*. Princeton university press.

[Roostalu et al., 2008] Roostalu, J., Jõers, A., Luidalepp, H., Kaldalu, N., and Tenson, T. (2008). Cell division in escherichia coli cultures monitored at single cell resolution. *BMC microbiology*, 8(1):68.

[Rousset et al., 1981] Rousset, M., Zweibaum, A., and Fogh, J. (1981). Presence of glycogen and growth-related variations in 58 cultured human tumor cell lines of various tissue origins. *Cancer Research*, 41(3):1165–1170.

[Rudell and Sangiovanni-Vincentelli, 1987] Rudell, R. and Sangiovanni-Vincentelli, A. (1987). Multiple-valued minimization for PLA optimization. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 6(5):727–750.

[Saez-Rodriguez et al., 2007] Saez-Rodriguez, J., Simeoni, L., Lindquist, J., Hemenway, R., Bommhardt, U., Arndt, B., Haus, U., Weismantel, R., Gilles, E., Klamt, S., et al. (2007). A logical model provides insights into T cell receptor signaling. *PLoS Comput Biol*, 3(8):e163.

[Sapra et al., 2003] Sapra, S., Theobald, M., and Clarke, E. (2003). Sat-based algorithms for logic minimization. In *Computer Design, 2003. Proceedings. 21st International Conference on*, pages 510–517. IEEE.

[Schellenberger et al., 2011a] Schellenberger, J., Lewis, N. E., and Palsson, B. (2011a). Elimination of thermodynamically infeasible loops in steady-state metabolic models. *Biophysical journal*, 100(3):544–553.

[Schellenberger and Palsson, 2009] Schellenberger, J. and Palsson, B. Ø. (2009). Use of randomized sampling for analysis of metabolic networks. *Journal of biological chemistry*, 284(9):5457–5461.

[Schellenberger et al., 2011b] Schellenberger, J., Que, R., Fleming, R., Thiele, I., Orth, J. D., Feist, A., Zielinski, D., Bordbar, A., Lewis, N., Rahmanian, S., Kang, J., Hyduke, D., and Palsson, B. Ø. (2011b). Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox v2. 0. *Nature protocols*, 6(9):1290–1307.

[Schibler and Naef, 2005] Schibler, U. and Naef, F. (2005). Cellular oscillators: rhythmic gene expression and metabolism. *Current opinion in cell biology*, 17(2):223–229.

[Schilling et al., 2000] Schilling, C., Letscher, D., and Palsson, B. (2000). Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of theoretical biology*, 203(3):229–248.

[Schomburg et al., 2002] Schomburg, I., Chang, A., and Schomburg, D. (2002). Brenda, enzyme data and metabolic information. *Nucleic acids research*, 30(1):47–49.

[Schöning, 1992] Schöning, U. (1992). *Logik für Informatiker*. BI-Wissenschaftsverlag.

[Schrijver, 1998] Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons.

[Schuetz et al., 2007] Schuetz, R., Kuepfer, L., and Sauer, U. (2007). Systematic evaluation of objective functions for predicting intracellular fluxes in escherichia coli. *Molecular systems biology*, 3(1).

[Schuster and Holzhütter, 1995] Schuster, R. and Holzhütter, H.-G. (1995). Use of mathematical models for predicting the metabolic effect of large-scale enzyme activity alterations. *European Journal of Biochemistry*, 229(2):403–418.

[Schuster et al., 2011] Schuster, S., de Figueiredo, L., Schroeter, A., and Kaleta, C. (2011). Combining metabolic pathway analysis with evolutionary game theory. explaining the occurrence of low-yield pathways by an analytic optimization approach. *Biosystems*, 105(2):147–153.

[Schuster et al., 2000] Schuster, S., Fell, D., and Dandekar, T. (2000). A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature biotechnology*, 18(3):326–332.

[Schuster and Hilgetag, 1994] Schuster, S. and Hilgetag, C. (1994). On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, 2(02):165–182.

[Schuster et al., 2002a] Schuster, S., Hilgetag, C., Woods, J., and Fell, D. (2002a). Reaction routes in biochemical reaction systems: algebraic properties, validated calculation procedure and example from nucleotide metabolism. *Journal of mathematical biology*, 45(2):153–181.

[Schuster et al., 2002b] Schuster, S., Pfeiffer, T., Moldenhauer, F., Koch, I., and Dandekar, T. (2002b). Exploring the pathway structure of metabolism: decomposition into subnetworks and application to mycoplasma pneumoniae. *Bioinformatics*, 18(2):351–361.

[Shannon, 1938] Shannon, C. E. (1938). A symbolic analysis of relay and switching circuits. *American Institute of Electrical Engineers, Transactions of the*, 57(12):713–723.

[Shlomi et al., 2011a] Shlomi, T., Benyamini, T., Gottlieb, E., Sharan, R., and Ruppin, E. (2011a). Genome-scale metabolic modeling elucidates the role of proliferative adaptation in causing the warburg effect. *PLoS computational biology*, 7(3):e1002018.

[Shlomi et al., 2011b] Shlomi, T., Benyamini, T., Gottlieb, E., Sharan, R., and Ruppin, E. (2011b). Genome-scale metabolic modeling elucidates the role of proliferative adaptation in causing the warburg effect. *PLoS computational biology*, 7(3):e1002018.

[Shmulevich et al., 2002] Shmulevich, I., Dougherty, E., Edward, R., Kim, S., and Zhang, W. (2002). Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274.

[Shoval et al., 2012] Shoval, O., Sheftel, H., Shinar, G., Hart, Y., Ramote, O., Mayo, A., Dekel, E., Kavanagh, K., and Alon, U. (2012). Evolutionary trade-offs, pareto optimality, and the geometry of phenotype space. *Science*, 336(6085):1157–1160.

[Siñao et al., 2005] Siñao, E., Remy, E., Thieffry, D., and Chaouiya, C. (2005). Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in E. Coli. *Bioinformatics*, 21.

[Silverman et al., 2010] Silverman, S., Petti, A., Slavov, N., Parsons, L., Briehof, R., Thiberge, S., Zenklusen, D., Gandhi, S., Larson, D., Singer, R., and Botstein, D. (2010). Metabolic cycling in single yeast cells from unsynchronized steady-state populations limited on glucose or phosphate. *Proceedings of the National Academy of Sciences*, 107(15):6946–6951.

[Slavov et al., 2011] Slavov, N., Macinskas, J., Caudy, A., and Botstein, D. (2011). Metabolic cycling without cell division cycling in respiring yeast. *Proceedings of the National Academy of Sciences*, 108(47):19090–19095.

[Smallbone et al., 2007] Smallbone, K., Simeonidis, E., Broomhead, D., and Kell, D. (2007). Something from nothing- bridging the gap between constraint-based and kinetic modelling. *Febs Journal*, 274(21):5576–5585.

[Soh and Inoue, 2010] Soh, T. and Inoue, K. (2010). Identifying Necessary Reactions in Metabolic Pathways by Minimal Model Generation. In *ECAI*, pages 277–282.

[Steggles et al., 2007] Steggles, L., Banks, R., Shaw, O., and Wipat, A. (2007). Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach. *Bioinformatics*, 23(3):336.

[Tamura et al., 2009] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M. (2009). Compiling finite linear csp into sat. *Constraints*, 14(2):254–272.

[Terzer and Stelling, 2008] Terzer, M. and Stelling, J. (2008). Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics*, 24(19):2229–2235.

[Thattai and Van Oudenaarden, 2001] Thattai, M. and Van Oudenaarden, A. (2001). Intrinsic noise in gene regulatory networks. *Proceedings of the National Academy of Sciences*, 98(15):8614–8619.

[Thieffry and Thomas, 1995] Thieffry, D. and Thomas, R. (1995). Dynamical behaviour of biological regulatory networks - II. Immunity control in bacteriophage lambda. *Bulletin of Mathematical Biology*, 57(2):277–297.

[Thomas, 1973] Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of theoretical biology*, 42(3):563–585.

[Thomas et al., 1995] Thomas, R., Thieffry, D., and Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks - I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, 57(2):247–276.

[Topham et al., 1986] Topham, C. M., Matthews, B., and Dalziel, K. (1986). Kinetic studies of 6-phosphogluconate dehydrogenase from sheep liver. *European Journal of Biochemistry*, 156(3):555–567.

[Van Ham, 1979] Van Ham, P. (1979). How to deal with variables with more than two levels. In *Kinetic Logic a Boolean Approach to the Analysis of Complex Regulatory Systems*, pages 326–343. Springer.

[Varma and Palsson, 1994a] Varma, A. and Palsson, B. (1994a). Metabolic flux balancing: Basic concepts, scientific and practical use. *Biotechnology*, 12.

[Varma and Palsson, 1994b] Varma, A. and Palsson, B. (1994b). Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli W3110. *Applied and environmental microbiology*, 60(10):3724.

[Veber et al., 2008] Veber, P., Guziolowski, C., Le Borgne, M., Radulescu, O., and Siegel, A. (2008). Inferring the role of transcription factors in regulatory networks. *BMC bioinformatics*, 9(1):228.

[Villa and Sangiovanni-Vincentelli, 1990] Villa, T. and Sangiovanni-Vincentelli, A. (1990). Nova: state assignment of finite state machines for optimal two-level logic implementation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 9(9):905–924.

[Wagner and Urbanczik, 2005] Wagner, C. and Urbanczik, R. (2005). The geometry of the flux cone of a metabolic network. *Biophysical journal*, 89(6):3837.

[Wortel et al., 2014] Wortel, M. T., Peters, H., Hulshof, J., Teusink, B., and Bruggeman, F. J. (2014). Metabolic states with maximal specific rate carry flux through an elementary flux mode. *FEBS Journal*, 281(6):1547–1555.

[Xi et al., 2009] Xi, Y., Chen, Y.-P. P., Cao, M., Wang, W., and Wang, F. (2009). Analysis on relationship between extreme pathways and correlated reaction sets. *BMC bioinformatics*, 10(Suppl 1):S58.

[Xie, 2012] Xie, L. (2012). Avoid internal loops in steady state flux space sampling. *arXiv preprint*, 1210.5234.

[Yang et al., 2010] Yang, L., Mahadevan, R., and Cluett, W. R. (2010). Designing experiments from noisy metabolomics data to refine constraint-based models. In *American Control Conference (ACC), 2010*, pages 5143–5148. IEEE.

[Yanofsky, 2000] Yanofsky, C. (2000). Transcription attenuation: once viewed as a novel regulatory strategy. *Journal of bacteriology*, 182(1):1–8.

[Yuan et al., 2011] Yuan, Q., Pang, J., Mauw, S., Trairatphisan, P., Wiesinger, M., and Sauter, T. (2011). A study of the pdgf signaling pathway with prism. *arXiv preprint arXiv:1109.1367*.

[Zevedei-Oancea and Schuster, 2003] Zevedei-Oancea, I. and Schuster, S. (2003). Topological analysis of metabolic networks based on Petri net theory. *In silico biology*, 3(3):323–345.

[Zhang et al., 2008] Zhang, R., Shah, M., Yang, J., Nyland, S., Liu, X., Yun, J., Albert, R., and Loughran, T. (2008). Network model of survival signaling in large granular lymphocyte leukemia. *Proceedings of the National Academy of Sciences*, 105(42):16308.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel genutzt habe. Die vorliegende Arbeit ist in keinem anderen Promotionsverfahren eingereicht worden.

Berlin, den 9. Januar 2015

Aljoscha Palinkas