Constraint-based methods for phasing in crystallography

Corinna Melanie Heldt

Dissertation
zur Erlangung des Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
des Fachbereichs Mathematik und Informatik
der Freien Universität Berlin

Freie Universität Berlin
2010

Gutachter:

Prof. Dr. Alexander Bockmayr (Freie Universität Berlin)
Prof. Dr. Alexandre Urzhumtsev (Université de Strasbourg und
Université Henri Poincaré, Nancy)

Tag der Disputation: 14. Dezember 2010

Acknowledgments

It is a pleasure to thank those who made this thesis possible and I express my regards and gratitude to them supporting me in any respect during its completion.

My deepest gratitude is to my advisor, Prof. Dr. Alexander Bockmayr for his encouragement, supervision, expert advice and guidance from the preliminary to the concluding level of this dissertation.

I would like to thank my co-advisor, Prof. Dr. Alexandre Urzhumtsev, for his support already in the early beginning of my thesis-writing, for helpful comments and providing me with useful software.

I am indebted to the former and current members of the Mathematics in Life Sciences Group, especially Dr. Heike Siebert, Dr. Gunnar Klau and Katja Geiger, for providing a stimulating and pleasant working environment, fruitful discussions and support in many different ways.

This thesis has benefited from the comments and criticisms of those who have read various drafts of various parts of it, including Heike, Klaus and Daniel.

My deepest thanks go to my family and friends, especially my husband Daniel, my parents Klaus and Astrid and my sisters Birgit, Inge and Caroline for their continuous encouragement, care and emotional support. This dissertation is dedicated to them.

Contents

Abstract

X-ray crystallography is one of the main methods to establish the three-dimensional structure of biological macromolecules, which is an essential base of structural biology and modern biotechnology.
In an X-ray experiment, one can measure only the magnitudes of the complex Fourier coefficients of the electron density distribution under study, but not their phases. The problem of recovering the lost phases is called the phase problem. As the information provided by the X-ray diffraction experiment is not sufficient to solve this problem, additional information about the electron density distribution has to be considered.
In this work binary integer programming approaches to model different topological properties of proteins are developed. Especially the connectivity constraint, enforcing that the calculated structure consists of at most a given number of connected components, is considered. Using graph theoretical methods and a separation algorithm, a model is worked out. The details of the implementation of the presented additional constraints are described and computational results are presented.

Introduction

Crystallography: (from Greek)

$\kappa\rho\upsilon\sigma\tau\alpha\lambda\lambda o\varsigma$ (krustallos) - ice,crystal
\+ $\gamma\rho\alpha\varphi\varepsilon\iota\nu$ (graphein) - to write

IUCR Online Dictionary of Crystallography

Crystallography is the branch of science devoted to the study of molecular and crystalline structure and properties, with far-reaching applications in mineralogy, chemistry, physics, mathematics, biology and material science [IUCR, 2006].

Knowledge about the three-dimensional structure of biological macromolecules is an essential foundation of structural biology and biotechnology. In X-ray crystallography the arrangement of atoms within a crystal is determined from a three-dimensional representation of the electron density. Various molecules and especially proteins can form crystals and X-ray crystallography is actually one of the main experimental methods to determine such structures. From X-ray experiments one gets *diffraction data* depending on the molecular structure. Then a relation between the this data and the crystal structure, which can be described by the electron density distribution, is searched. With the help of the diffraction data and the usage of mathematical as well as experimental methods, the electron density can be calculated. *Direct methods* use mathematical techniques to compute an electron density map from the diffraction data without any further experiments. The main problem here is the *phase problem*: experiments provide only the intensities of the X-rays diffracted in different directions. With their help the structure factors' magnitudes can be calculated, whereas the information about the phase shift of the structure factors is lost. If the structure factors were known, the electron density could be calculated.
X-ray crystallography can, for example, help to design pharmaceuticals, to understand enzyme mechanisms or different chemical processes or to analyse new materials.

In the first chapter some basics and especially mathematical background of direct methods are presented. Later, methods for solving the phase problem are worked out. A binary integer programming approach, suggested in [Lunin et al., 2002b], is developed in chapter 2. Based on this approach, methods to increase the quality of the solutions are worked out in the following chapters. Information about topological properties of proteins is used and formulated as a binary integer program. In discrete tomography, topological properties of the considered objects are taken into account. In [Bockmayr and Hooker, 2005] some of them are formulated for two-dimensional objects via a binary integer program. In chapter 3 these, as well as new formulations are presented and extended for three-dimensional objects.
In chapter 4, topological properties suitable for proteins, are regarded and binary integer programming models are worked out using methods from graph theory. In the subsequent chapter 5, a model for the whole phasing problem taking topological properties of proteins into account is derived. A different approach to model one of the topological constraints is shown in chapter 7. This approach is based on singular value decomposition and ellipsoidal methods.
A first model has been implemented, the details about the implementation are described in 6. Here, also computational results using real protein data are presented. The effectiveness of the presented model and the increase in the quality of the solutions of the phasing problem, reached by this new method, is shown.

Chapter 1

Some Crystallographic and Mathematical Basics

> Willst Du ins Unendliche schreiten,
> Geh nur im Endlichen nach allen Seiten.
> Willst Du Dich am Ganzen erquicken,
> So musst Du das Ganze im Kleinsten
> erblicken.
>
> Johann Wolfgang von Goethe

1.1 Introduction to X-ray Crystallography

1.1.1 X-ray-scattering

X-ray crystallography is a method to construct three-dimensional structure models of crystals by determining the arrangement of electrons. A *crystal* is a homogeneous solid in which a particular arrangement of atoms is repeated along all three spatial dimensions. From experiments one gets *diffraction data* depending on the molecular structure of the regarded crystal. X-rays are scattered exclusively by the electrons in the atoms, so a relation between the measured intensities of the beams diffracted at the object in question and the crystal structure which can be described by its electron density distribution is searched. Electron densities represent probabilistically where electrons can be found in the considered molecule. Crystallography can also be used to solve crystal structures of proteins, which are irregular and can be very large. Max Perutz and Sir John Kendrew were the first scientists to solve protein structures by X-ray crystallography. They were awarded

the Nobel prize in chemistry in 1962 for finding the structure of sperm whale myoglobin [Kendrew, 1964; Perutz, 1964].
Crystallography is nowadays used as a technique in structure determination and structure-based drug design. The knowledge about the three-dimensional structure of a protein can be used for the retrieval and design of ligands. More detailed information about drug design based on crystallography containing also concrete examples can be found in [Mannhold et al., 2003].

The first step on the way to estimate a crystal's electron density is the collecting of crystallographic data. This data is obtained with the help of a *diffractometer* or a synchrotron: an X-ray beam is diffracted by the crystal of interest into many discrete beams, each of them produces a reflection on a detector, which leads to a *diffraction pattern*, see Figure 1.1. The crystal and the film are rotated, the reflections are recorded, such that the resulting diffraction pattern is a three-dimensional one. With the help of this diffraction–data and using mathematical as well as experimental methods, the electron–density distribution can be derived.

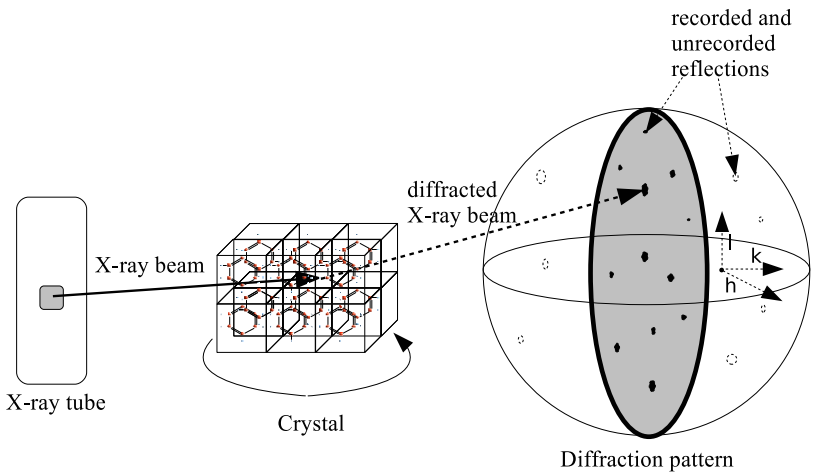


Figure 1.1: Diffractometer

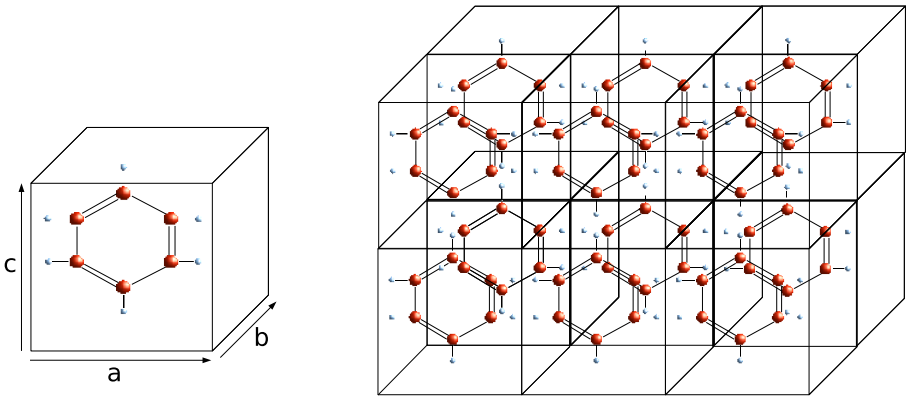
Direct methods use mathematical methods to derive the electron-density map from the diffraction data without any further experiments. The main problem here is the *phase–problem* which will be the main topic of this work. In 1985, Hauptmann and Karle were awarded the Nobel prize in chemistry for their work on solving the phase–problem with direct methods, [Hauptman, 1992; Karle, 1992].
The basics about X-ray diffraction presented in this chapter are further devel-

oped e.g. in [Barrow, 1979; Borchardt-Ott, 2002; Drenth, 1994; Giacovazzo, 1998; Giacovazzo et al., 2002; Rhodes, 1993; Sherwood, 1976].

1.1.2 Unit cell, real and reciprocal space

Vectors, matrices as well as higher-dimensional arrays will be noted with bold letters, $\mathbf{x}^T$ is the transpose of an array $\mathbf{x}$, as well as $\mathbf{A}^T$ is the transpose of a matrix $\mathbf{A}$. The scalar product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted by $\mathbf{x}^T\mathbf{y}$.

An ideal crystal consists of identical molecules, respectively complexes of molecules repeated periodically in all three dimensions, cf. Figure 1.2b. This means that a 3-dimensional parallelepiped can be found containing such a complex of molecules which builds up the whole crystal if it is repeatedly stacked together in all three dimensions. This parallelepiped, in general, is not unique. It is defined by the length of its edges as well as the angles between them and is called *unit cell* (see Figure 1.2). These base units generate a 3-dimensional *crystal lattice*.



(a) Unit cell (b) Crystal

Figure 1.2: Unit cells building a crystal: a) a unit cell, b) crystal built of unit cells

The unit cell's volume will be denoted with V_{cell}. Let $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$ span the unit cell. Then every vector $\mathbf{r} \in \mathbb{R}^3$ can be written in this basis, i.e., $\mathbf{r} = x_1\mathbf{a}+x_2\mathbf{b}+x_3\mathbf{c}$, where $\mathbf{x} = (x_1, x_2, x_3)^T \in \mathbb{R}^3$ is the vector of coordinates of $\mathbf{r}$ with respect to the basis $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$. The vector space defined by this basis is called *real space*. A vector $\mathbf{x} \in \mathbb{R}^3$ is inside the unit cell, see Figure 1.2a, if and only if $\mathbf{x} \in V = [0, 1]^3$.

The electron density distribution in a crystal can be described by a real function $\rho(\mathbf{r}) : \mathbb{R}^3 \to \mathbb{R}$. The electron density distribution function has three linearly independent periods corresponding to the length of the unit cell's edges, i.e.,

$$\rho(\mathbf{r}) = \rho(\mathbf{r} + k_1 \cdot \mathbf{a} + k_2 \cdot \mathbf{b} + k_3 \cdot \mathbf{c}), k_1, k_2, k_3 \in \mathbb{Z}. \tag{1.1}$$

This means, if the electron density distribution values in the unit cell are known, its values in the whole crystal are known, due to periodicity.

For the vector of coordinates $\mathbf{x} = (x_1, x_2, x_3)$, the electron density function has integer periods in all three directions:

$$\rho(\mathbf{x}) = \rho(\mathbf{x} + \mathbf{k}), \ \forall \mathbf{x} \in V, \ \forall \mathbf{k} \in \mathbb{Z}^3. \tag{1.2}$$

The crystal lattice defined by the lattice points $\mathbf{p} \in \mathbb{R}^3$

$$\mathbf{p} = r\mathbf{a} + s\mathbf{b} + t\mathbf{c}, \ r, s, t \in \mathbb{Z} \tag{1.3}$$

is called real lattice, in contrast to the reciprocal lattice which will be introduced later.

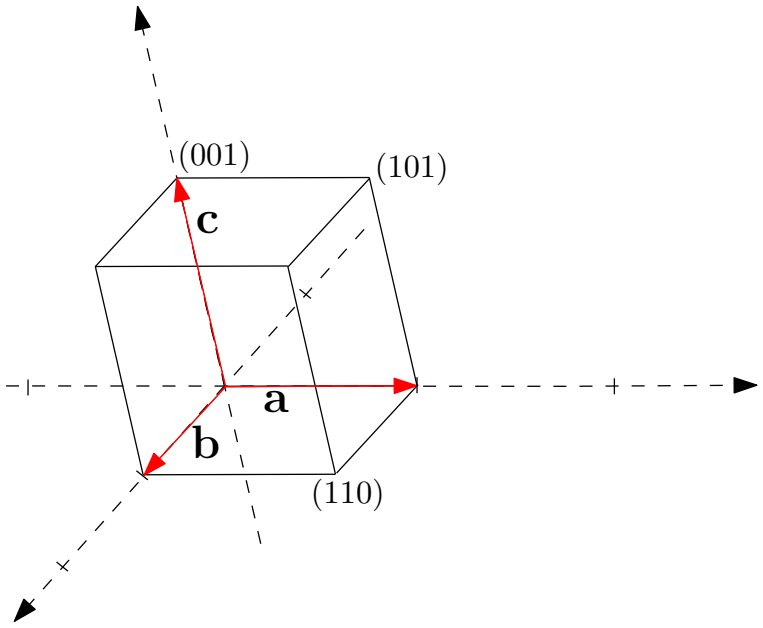


Figure 1.3: Unit cell with the basis vectors of the real lattice

1.1.3 Bragg's law

A reflection on the diffraction pattern is only produced by reflected waves of significant intensity. To reach significant intensities, maximal constructive interference of the diffracted X-ray-waves is necessary. Therefore we will now derive a condition ensuring maximal constructive interference.
According to Figure 1.4 let d be the interplanar distance, i.e., the minimal distance between the layers of the regarded crystal. The path distance of two waves, diffracted at two different layers will be denoted by δ . As one can easily see, the path distance δ is given by

$$\delta = 2d \sin\theta. \tag{1.4}$$

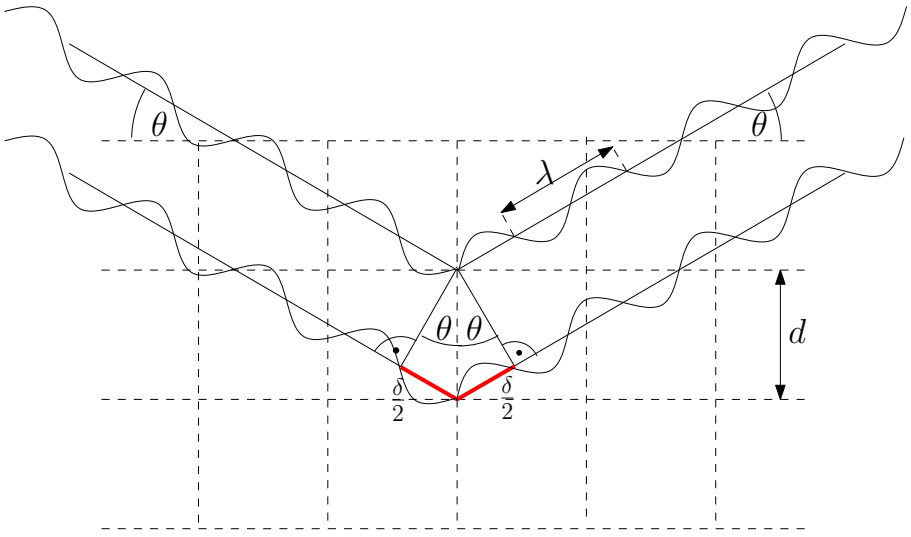


Figure 1.4: Bragg's law

Maximal constructive interference is reached, if the path distance is an integer multiple of the wavelength λ.

$$2d \sin\theta = n\lambda, \ n \in \mathbb{Z}. \tag{1.5}$$

Only waves satisfying this condition result in a reflected wave of significant intensity which can produce a reflex on the film for the diffraction pattern. Notice, that the interplanar distances have to be in the same order of magnitude like the wave lengths of the X-rays to get diffraction [Barrow, 1979]. The smaller the interplanar distance, the bigger is the resolution of the diffraction pattern.

1.1.4 Ewald sphere

Figure 1.5 shows Bragg's law for the incoming X-ray $\mathbf{r_{in}}$ and the outgoing X-ray $\mathbf{r_{out}}$, both having an angle of θ to the reflecting Bragg plane B. As the angle between the incoming X-ray and the corresponding Bragg plane B varies between $0°$ and $90°$, the vector $\mathbf{r_{out}}$ representing the outgoing X-ray ends on the surface of a sphere, the *Ewald sphere*. Figure 1.5 shows a spheric section of the Ewald sphere. Due to this construction σ is perpendicular to the plane B.

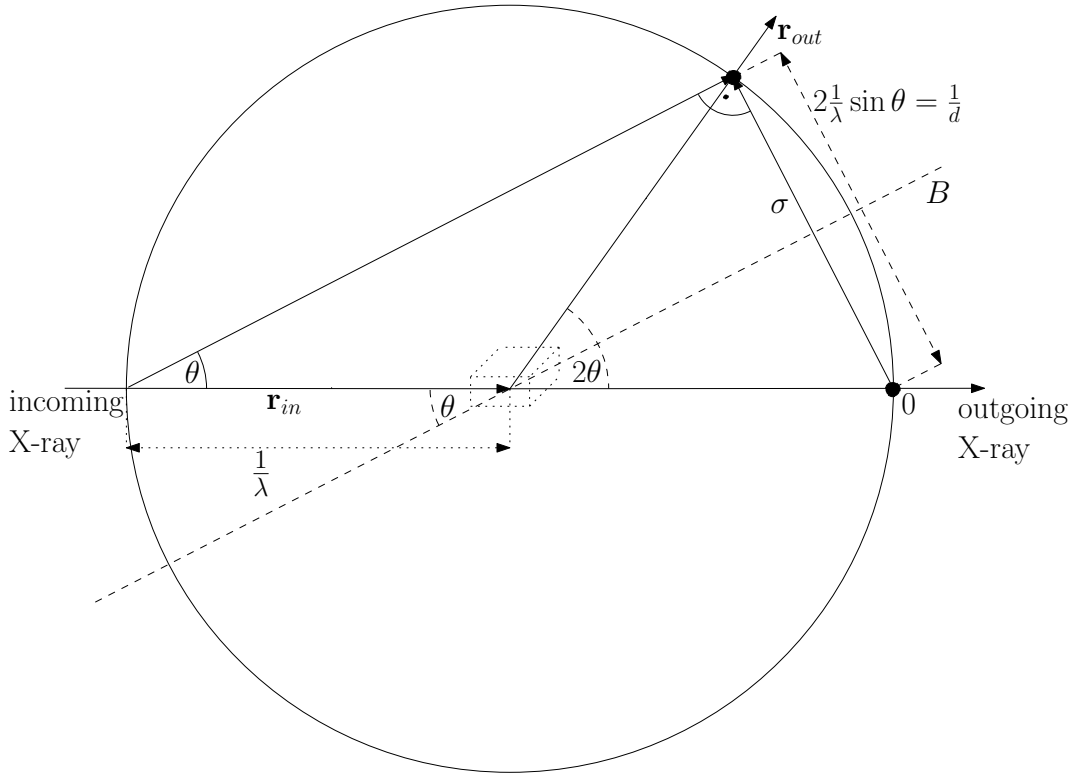


Figure 1.5: Construction of Ewald sphere

Each reflection of the X-ray-beam can be assigned three coordinates in the three-dimensional space of the diffraction pattern, the *reciprocal space*. The reciprocal lattice, consisting of the reflection points, is defined to be the set of vectors $S = (\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*)$ satisfying

$$e^{2\pi i S \cdot X} = 1 \quad \text{for all real lattice vectors } X = (\mathbf{a}, \mathbf{b}, \mathbf{c}). \tag{1.6}$$

The reciprocal basis vectors $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$ spanning the reciprocal space and identifying the *reciprocal lattice* are therefore given by

$$\mathbf{a}^* = \frac{\mathbf{b} \times \mathbf{c}}{\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})}, \tag{1.7}$$

$$\mathbf{b}^* = \frac{\mathbf{a} \times \mathbf{c}}{\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})}, \tag{1.8}$$

$$\mathbf{c}^* = \frac{\mathbf{a} \times \mathbf{b}}{\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})}. \tag{1.9}$$

The reciprocal lattice points are given by $\mathbf{s} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$, $h, k, l \in \mathbb{Z}$. Due to its definition, the vector σ is pointing to the reciprocal lattice point (hkl). Here 0 is the origin of the reciprocal space. The reciprocal lattice is rotated around its origin. If a reciprocal lattice point cuts the sphere, Bragg's law is fulfilled and the diffracted X-ray has maximal intensity. So we get a reflection on the film. If a crystal is put into a X-ray beam, only some of the Bragg planes will be in the correct orientation to show diffraction and produce reflections on the film. If the crystal is rotated, other Bragg planes will produce reflections [Barrow, 1979].

After describing the physical background of X-ray crystallography, now its mathematical basics will be presented.

1.2 Fourier analysis

The phase problem is one of the most important problems in X-ray crystallography. To understand its origin, it is necessary to know some basics about Fourier transforms and especially about 3-dimensional Fourier transforms. These will show a correlation between the intensities measured in the X-ray-experiment and the electron density distribution one tries to calculate.
Fourier series were introduced by Joseph Fourier who made important contributions to the study of trigonometric series and published his results in [Fourier, 1822].
The problem of calculating the electron density distribution in the unit cell will later on be simplified to the problem of calculating it on discrete grid points in the cell. Then Discrete Fourier Transforms will be needed and therefore also be introduced in this chapter.

1.2.1 Fourier series

First, some useful notations necessary for the introduction of Fourier series will be given. Fourier series and coefficients will be introduced and the convergence of Fourier series will be examined. Our introduction to Fourier series is based on [Gasquet and Witomski, 1999; Vretblad, 2003].

Definition 1.1. (Periodic function), [Gasquet and Witomski, 1999]

A function $f : \mathbb{R} \to \mathbb{C}$ *is called* periodic *with period* $P > 0$*, if*

$$f(t + P) = f(t), \quad \forall t \in \mathbb{R}. \tag{1.10}$$

Notation 1.2. $(\mathbf{L^1_p(0, P)})$

Let $L^1_p(0, P)$ be the set of Lebesgue-integrable functions of period P, i.e.

$$L^1_p(0, P) \stackrel{def}{=} \left\{ f : \mathbb{R} \to \mathbb{C} \mid f \text{ has period } P \text{ and } \int_0^P |f(t)|dt < \infty \right\}.$$

Notation 1.3. $(\mathbf{L^2_p(0, P)})$

Let $L^2_p(0, P)$ be the set of periodic and square integrable functions, i.e.

$$L^2_p(0, P) \stackrel{def}{=} \left\{ f : \mathbb{R} \to \mathbb{C} \mid f \text{ has period } P \text{ and } \int_0^P |f(t)|^2 dt < \infty \right\}. \tag{1.11}$$

Every function contained in $L^2_p(0, P)$ is also contained in $L^1_p(0, P)$, i.e.,

$$L^2_p(0, P) \subseteq L^1_p(0, P). \tag{1.12}$$

Definition 1.4. (Fourier series), [Vretblad, 2003]

The Fourier series *of a function* $f \in L_p^1(0,P)$ *in* $t \in \mathbb{R}$ *is defined by*

$$\sum_{n=-\infty}^{\infty} c_n(f) e^{-2i\pi n \frac{t}{P}}, \tag{1.13}$$

where the Fourier coefficients $c_n(f),\ n \in \mathbb{Z}$ *are given by*

$$c_n(f) = \frac{1}{P} \int_0^{P-1} f(t) \exp(2\pi i \cdot n \frac{t}{P}) dt. \tag{1.14}$$

The following classical theorem and corollary investigate the convergence of Fourier series, proofs can be found in [Gasquet and Witomski, 1999].

Theorem 1.5. [Gasquet and Witomski, 1999]

Let $f : \mathbb{R} \to \mathbb{R}$ *be a continuous function with period* P *that is differentiable on* $\mathbb{R}$*, except possibly at a finite number of points. Assume that* f' *is piecewise continuous. Then*

1. *the Fourier series of* f' *is obtained by differentiating the Fourier series of* f *term by term.*
2. *the Fourier coefficients* $c_n(f)$ *of* f *converge absolutely, i.e., they satisfy*

$$\sum_{n=-\infty}^{\infty} |c_n(f)| < \infty, \tag{1.15}$$

3. *the Fourier series of* f *converges uniformly to* f *on* $\mathbb{R}$*.*

Corollary 1.6. [Gasquet and Witomski, 1999]

If $f \in L_p^2(0,a)$ *and if its Fourier coefficients* $c_n(f)$ *satisfy*

$$\sum_{n=-\infty}^{\infty} |c_n(f)| < \infty, \tag{1.16}$$

then f *coincides with a continuous function* $\tilde{f}$ *almost everywhere and the Fourier series of* f *converges uniformly to* $\tilde{f}$ *on* $\mathbb{R}$*.*

1.2.2 Discrete Fourier Transform

As the original problem of finding the entire electron density distribution will be simplified later on to searching the electron density only for a set of given discrete points, we need to introduce the Discrete Fourier Transform, cf. [Briggs and Henson, 1995]. For clarity, first the one-dimensional case will be regarded and later on extended to the three-dimensional case, which is the one needed in crystallography.

Definition 1.7. (Discrete Fourier Transform (DFT))

Let N be a positive integer and let $(f_n),\ n \in \{0,\dots,N-1\}$ be a sequence of N complex numbers. The value $\frac{1}{N}$ is called intersample distance. Then the Discrete Fourier Transform $F_k,\ k \in \{0,\dots,N-1\}$ *of (f_n) is another sequence of N complex numbers given by*

$$F_k \stackrel{def}{=} \frac{1}{N}\sum_{n=0}^{N-1} f_n e^{2\pi i \frac{nk}{N}},\ 0 \le k \le N-1. \tag{1.17}$$

Definition 1.8. (Inverse Discrete Fourier Transform (IDFT))

Let N be a positive integer and let $(F_k),\ k = 0,\dots N-1$ be a sequence of N complex numbers. Then the Inverse Discrete Fourier Transform *of (F_k) is another sequence of N complex numbers given by*

$$f_n \stackrel{def}{=} \sum_{n=0}^{N-1} F_k e^{-2\pi i \frac{nk}{N}},\ 0 \le n \le N-1. \tag{1.18}$$

The Inverse Discrete Fourier Transform of the Discrete Fourier Transform of f_n is again f_n and vice versa, the Discrete Fourier Transform of the Inverse Discrete Fourier Transform of F_k is again F_k.

1.2.3 Multidimensional Fourier analysis

The electron density distribution is a function $\rho : \mathbb{R}^3 \to \mathbb{R}$, so multidimensional Fourier series will be needed. These are introduced in [Tolstov, 1962] as well as in [Vretblad, 2003], a survey on the investigations on the theory of multi-dimensional Fourier series including convergence questions is given in [Golubov, 1984] and [Vretblad, 2003]. Multidimensional Discrete Fourier Transforms are mentioned in [Briggs and Henson, 1995; Vretblad, 2003].

Definition 1.9. (Periodic function in $\mathbb{R}^3$)

A function $f(x,y,z) : \mathbb{R}^3 \to \mathbb{R}$ *is called* periodic *with period* $(M,N,P) \in \mathbb{R}^3$, $M,N,P > 0$, *if*

$$f(x+M, y+N, z+P) = f(x,y,z), \quad \forall (x,y,z) \in \mathbb{R}^3. \tag{1.19}$$

Definition 1.10. (Fourier series)

The Fourier series *of a continuous periodic function* $f(x,y,z) : \mathbb{R}^3 \to \mathbb{R}$ *with period* (M,N,P) *in* $\mathbb{Z}^3$, $M,N,P > 0$, *is defined by*

$$\sum_{m,n,p=-\infty}^{\infty} c_{mnp}(f) e^{-2i\pi(\frac{mx}{M}+\frac{ny}{N}+\frac{pz}{P})}. \tag{1.20}$$

The Fourier coefficients c_{mnp} are given by

$$c_{mnp}(f) = \frac{1}{MNP} \int_0^{M-1} \int_0^{N-1} \int_0^{P-1} f(x,y,z) e^{2i\pi(\frac{mx}{M}+\frac{ny}{N}+\frac{pz}{P})} dx\ dy\ dz, \quad \forall m,n,p \in \mathbb{Z}. \tag{1.21}$$

For better readability the following notations will now be used:

$$\mathbf{x} = (\frac{x}{M}, \frac{y}{N}, \frac{z}{P})^T \in \mathbb{R}^3, \quad \mathbf{h} = (m,n,p)^T \in \mathbb{Z}^3, \tag{1.22}$$

$$\mathbf{U} = [0, M-1] \times [0, N-1] \times [0, P-1] \subseteq \mathbb{R}^3, \quad U = MNP, \tag{1.23}$$

so the Fourier series can now be written in the form

$$\sum_{\mathbf{h} \in \mathbb{Z}^3} c_{\mathbf{h}}(f) \exp(-2i\pi(\mathbf{h}^T \mathbf{x})), \quad \forall \mathbf{x} \in \mathbf{U}. \tag{1.24}$$

and the Fourier coefficients are

$$c_{\mathbf{h}}(f) = \frac{1}{U} \int_{\mathbf{U}} f(\mathbf{x}) \exp(2i\pi(\mathbf{h}^T \mathbf{x}))\ d\mathbf{x}, \quad \forall \mathbf{h} \in \mathbb{Z}^3. \tag{1.25}$$

Definition 1.11. (Three dimensional Discrete Fourier Transform)

Analogous to (1.7)*, let* $(f(\mathbf{h}))$ *be a complex-valued three-dimensional array,*

$$\mathbf{h} \in \Pi, \quad \Pi = [0, M-1] \times [0, N-1] \times [0, P-1] \subseteq \mathbb{Z}^3. \tag{1.26}$$

Let $\mathbf{M}$ *be the diagonal matrix* $\mathbf{M} = diag(M, N, P)$*,* $M, N, P \in \mathbb{N}$*. The* three dimensional Discrete Fourier Transform F *of* $f(\mathbf{h})$ *is given by*

$$F(\mathbf{j}) = \sum_{\mathbf{h} \in \Pi} f(\mathbf{h}) \exp(2\pi i(\mathbf{h}^T \mathbf{M}^{-1} \mathbf{j})), \quad \forall \mathbf{j} \in \Pi. \tag{1.27}$$

For the inverse three dimensional Discrete Fourier Transform analogously to (1.8) one gets:

Definition 1.12. (3D inverse Discrete Fourier Transform)

The three dimensional Inverse Discrete Fourier Transform $f(\mathbf{h})$ *is given by*

$$f(\mathbf{h}) = \frac{1}{U} \sum_{\mathbf{j} \in \Pi} F(\mathbf{j}) \exp(-2\pi i(\mathbf{h}^T \mathbf{M}^{-1} \mathbf{j})), \quad \forall \mathbf{h} \in \Pi. \tag{1.28}$$

1.2.4 Properties of Fourier series and DFT

Some mathematical properties of the Fourier series as well as the Discrete Fourier Transform can directly be used in the reconstruction process in crystallography. The electron density distribution is a real-valued function, while the structure factors are complex values.

Definition 1.13. (Magnitude, phase)

Every complex number $z \in \mathbb{C}$ *can be written in polar form, i.e.,* $z = |z|e^{i\phi}$*,* $\phi \in [0, 2\pi[$*;* $|z|$ *is called* magnitude *and* ϕ *the* phase angle *or simply* phase *of* z*.*

Hermitian symmetry

If $f : \mathbb{R}^3 \to \mathbb{R}$ is a real function with period $\mathbf{P} \in \mathbb{R}^d$, $d \in \mathbb{N}$, then the Fourier coefficients $c_{\mathbf{n}}$ show Hermitian symmetry, [Gasquet and Witomski, 1999], meaning that $c_{\mathbf{n}}$ and $c_{\mathbf{n}}^*$ have the same magnitudes but opposite phase angles, i.e.,

$$c_{-\mathbf{n}} = c_{\mathbf{n}}^*, \ \forall \mathbf{n} \in \mathbb{Z}^3, \tag{1.29}$$

where $c_{\mathbf{n}}^*$ is the complex conjugate of $c_{\mathbf{n}}$.

Analogous, if $f(\mathbf{h})$ is real-valued, the Discrete Fourier Transform shows Hermitian symmetry [Briggs and Henson, 1995], i.e.

$$F^*(\mathbf{k}) = F(-\mathbf{k}), \ \forall \mathbf{k} \in \mathbb{Z}^3. \tag{1.30}$$

Real values

If $f(\mathbf{h})$ is real-valued for all $\mathbf{h} = (m, n, p) \in \Pi$, then $F(0, 0, 0)$ is real, as well as

$$F(\frac{M}{2}, 0, 0),\ F(0, \frac{N}{2}, 0),\ F(0, 0, \frac{P}{2}), ...,\ F(\frac{M}{2}, \frac{N}{2}, \frac{P}{2}). \tag{1.31}$$

This property will be shown exemplarily for $F(\frac{M}{2}, 0, 0)$. Let be $\mathbf{h} = (m, n, p) \in \Pi$, then:

$$F(\frac{M}{2}, 0, 0) = \sum_{\mathbf{h}\in\Pi} f(\mathbf{h}) \exp(2\pi i(\frac{mM}{2M})) = \sum_{\mathbf{h}\in\Pi} f(\mathbf{h}) \cdot (-1)^m. \tag{1.32}$$

Periodicity

The complex sequences defined in Definition (1.11) and (1.12) are periodic sequences with period $\mathbf{P} = (M, N, P) \in \mathbb{Z}^3$ satisfying:

$$F(x + M, y + N, z + P) = F(x, y, z) \quad \forall (x, y, z) \in \mathbb{Z}^3 \tag{1.33}$$

and

$$f(x + M, y + N, z + P) = f(x, y, z) \quad \forall (x, y, z) \in \mathbb{Z}^3. \tag{1.34}$$

This can easily be shown using that $\exp(2\pi i(\mathbf{h}^T\mathbf{M}^{-1}(M, N, P)^T)) = 1$:

$$\begin{aligned}
& F(x + M, y + N, z + P) \\
= & \sum_{\mathbf{h}\in\Pi} f(\mathbf{h}) \exp(2\pi i(\mathbf{h}^T\mathbf{M}^{-1}(x + M, y + N, z + P)^T) \\
= & \sum_{\mathbf{h}\in\Pi} f(\mathbf{h}) \exp(2\pi i(\mathbf{h}^T\mathbf{M}^{-1}(x, y, z)^T) + 2\pi i(\mathbf{h}^T\mathbf{M}^{-1}(M, N, P)^T)) \\
= & \sum_{\mathbf{h}\in\Pi} f(\mathbf{h}) \exp(2\pi i(\mathbf{h}^T\mathbf{M}^{-1}(x, y, z)^T)) \\
= & F(x, y, z), \quad \forall (x, y, z) \in \mathbb{Z}^3,
\end{aligned}$$

and analogous for the IDFT.

Parseval's theorem

A fundamental property of the DFT is described by Paseval's relation [Briggs and Henson, 1995]:

$$\sum_{\mathbf{h}\in\mathbf{\Pi}} |f(\mathbf{h})|^2 = \frac{1}{MNP} \sum_{\mathbf{j}\in\Pi} |F(\mathbf{j})|^2. \tag{1.35}$$

1.2.5 Protein crystallography

Proteins form crystals under certain circumstances. Crystallographers are able to grow crystals of proteins without denaturing the proteins [Rhodes, 1993]. In 1934, Dorothy Hodgkin and John Desmond Bernal recorded the first X-ray diffraction pattern of a protein crystal. In 1962, Max Ferdinand Perutz and John Kendrew were awarded the Nobel Prize for Chemistry for their studies of the structures of globular proteins. They determined the three-dimensional structure of protein crystals from their X-ray diffraction patterns [Kendrew, 1964; Perutz, 1964].

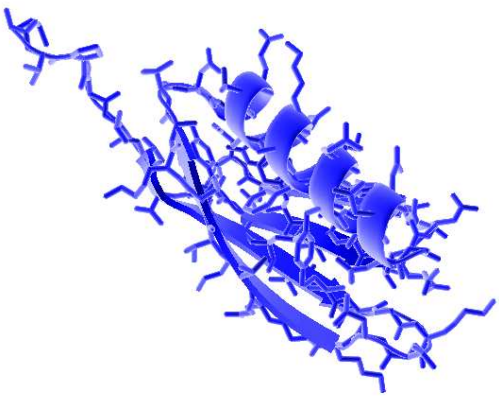


Figure 1.6: Protein G, picture made with PYMOL [DeLano, 2002]

1.2.6 The phase problem

According to the theory of quantum mechanics, the position of an electron can only be described in probabilistic terms, see e.g. [Barrow, 1979]. The *electron density* of an electron is the probability for this electron to be at a specific location. The structure of a crystal calculated from diffraction data is represented by electron density maps.

We are searching for the electron density distribution $\rho(\mathbf{x}) : V \to \mathbb{R}$ in a crystal with V being the unit cell of the crystal. In the X-ray experiment, the X-ray beam is scattered by the electrons in the crystal. Due to the crystal structure, $\rho : \mathbb{R}^3 \to \mathbb{R}$ is a periodic function and therefore can be developed into a *Fourier series*, cf. (1.24), [Drenth, 1994; Sherwood, 1976]

$$\rho(\mathbf{x}) = \frac{1}{V_{cell}} \sum_{\mathbf{h} \in \mathbb{Z}^3} \mathbf{F}(\mathbf{h}) \exp(-2\pi i(\mathbf{h} \cdot \mathbf{x})), \ \forall \mathbf{x} \in V. \tag{1.36}$$

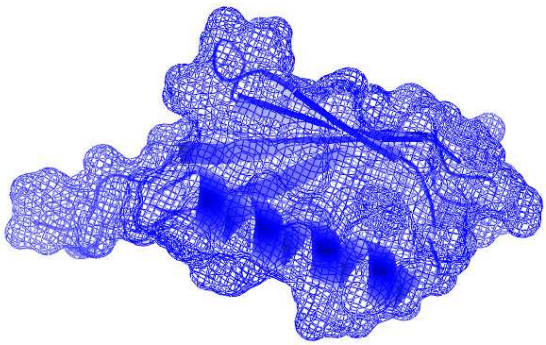


Figure 1.7: Electron density map of Protein G, picture made with PYMOL [DeLano, 2002]

If we knew the structure factors $\mathbf{F}(\mathbf{h})$ for all $\mathbf{h} \in \mathbb{Z}^3$, using formula (1.36) the electron density distribution could be calculated. The Fourier coefficients $\mathbf{F}(\mathbf{h}), \mathbf{h} \in \mathbb{Z}^3$, which are called *structure factors* in crystallography, are given by the formula

$$\mathbf{F}(\mathbf{h}) = \int_V \rho(\mathbf{x}) \exp(2\pi i(\mathbf{h} \cdot \mathbf{x}))d\mathbf{x}, \ \forall \mathbf{h} \in \mathbb{Z}^3, \tag{1.37}$$

cf. (1.25). Since the structure factors are complex numbers, they can be written in the form

$$\mathbf{F}(\mathbf{h}) = F(\mathbf{h}) \exp(i\varphi(\mathbf{h})), \tag{1.38}$$

with magnitude $F(\mathbf{h}) = |\mathbf{F}(\mathbf{h})|$ and phase $\varphi(\mathbf{h}) \in [0, 2\pi[$.

The *electron density map resolution* specifies the size of distinguishable details in the diffraction pattern. For the calculation of the electron density distribution using equation (1.36), an infinite number of structure factors $\mathbf{F}(\mathbf{h})$, $\mathbf{h} \in \mathbb{Z}^3$ would have to be known. The restriction to a finite number of structure factors, taken into account for the calculation of the electron density values, defines the resolution.
Assume, that the lengths of the edges of the considered unit cell are given by $a, b, c \in \mathbb{R}$. For the simple case that the edges of the considered unit cell all are orthogonal, the resolution $d(h, k, l)$, $(h, k, l) \in \mathbb{Z}^3$ is defined as

$$d(h, k, l) = \left(\frac{h^2}{a^2} + \frac{k^2}{b^2} + \frac{l^2}{c^2} \right)^{-\frac{1}{2}}. \tag{1.39}$$

If an electron density distribution is calculated with a set of structure factors $\mathbf{F}(h, k, l)$, such that

$$d_{high} \leq d(h, k, l) \leq d_{low}, \tag{1.40}$$

the resolution of the electron density map is d_{high}. The value d_{low} is the low-resolution cut-off limit. The smaller d_{high}, the higher the resolution. The value d_{low} specifies if the structure factors with small absolute index values (h, k, l) are included and should therefore be large, if they should be included. The range $d_{high} - d_{low}$ is called resolution range.

The intensity $\overline{I}(\mathbf{h})$, $I : \mathbb{R}^3 \to \mathbb{R}^{\geq 0}$ of a diffraction is proportional to the squared structure factors with a known constant of proportionality [Drenth, 1994; Rhodes, 1993; Sayre, 1951]. This means

$$C \cdot \overline{I}(\mathbf{h}) = \mathbf{F}(\mathbf{h}) \cdot \mathbf{F}^*(\mathbf{h}), \ \forall \mathbf{h} \in \mathbb{Z}^3, \tag{1.41}$$

for some constant $C > 0$. and so, knowing the intensities, the magnitudes of the structure factors can be calculated by

$$|\mathbf{F}(\mathbf{h})| = \sqrt{C \cdot \overline{I}(\mathbf{h})}, \ \forall \mathbf{h} \in \mathbb{Z}^3. \tag{1.42}$$

In the following, $I(\mathbf{h}) = C \cdot \overline{I}(\mathbf{h}), \ \forall \mathbf{h} \in \mathbb{Z}^3$ will be used.

The X-ray diffraction experiment provides only the intensities $I(\mathbf{h})$, for some $\mathbf{h} \in \mathbb{Z}^3$ of the X-rays diffracted in different directions. Due to equation (1.42) the magnitudes of the corresponding structure factors $F(\mathbf{h})$ can be calculated, whereas the information about the phase shift is lost. If we knew the intensities $I(\mathbf{h})$, for all $\mathbf{h} \in \mathbb{Z}^3$, all structure factor magnitudes could be calculated. Knowing additionally all structure factor phase shifts, using equation (1.36) we could calculate the electron density distribution.
The *phase problem* is the problem to find the lost phase shift of the structure factors belonging to the measured intensities. Due to the observations made above, it is an important step on the way to determine a structure model. As the information provided by the X-ray diffraction experiment is not sufficient to solve the problem to find the lost phases, additional information about the electron density distribution will be considered.

1.2.7 Symmetries

Friedel's law

As the electron density is always real, one can derive that $I(\mathbf{h}) = I(-\mathbf{h}), \ \forall \mathbf{h} \in \mathbb{Z}^3$. This property is called *Friedel's law* and was first published in [Friedel, 1913].
It is based on the structure factors' *Hermitian symmetry*, already mentioned in (1.29), meaning that $\mathbf{F}(\mathbf{h})$ and $\mathbf{F}(-\mathbf{h})$ have the same magnitudes but opposite phase angles:

$$\mathbf{F}(\mathbf{h}) = |\mathbf{F}(\mathbf{h})| \exp(i\varphi((\mathbf{h}))), \tag{1.43}$$
$$\mathbf{F}(-\mathbf{h}) = |\mathbf{F}(\mathbf{h})| \exp(-i\varphi((\mathbf{h}))) = \mathbf{F}^*(\mathbf{h}),\ \forall \mathbf{h} \in \mathbb{Z}^3. \tag{1.44}$$

$$\begin{aligned} I(\mathbf{h}) &= \mathbf{F}(\mathbf{h}) \cdot \mathbf{F}^*(\mathbf{h}) \\ &= \left(\sum_{j=1}^{N} f_j \exp(2\pi i \mathbf{r}_j \cdot \mathbf{h}) \right) \left(\sum_{k=1}^{N} f_k \exp(-2\pi i \mathbf{r}_k \cdot \mathbf{h}) \right) \\ &= \sum_{j=1}^{N} \sum_{k=1}^{N} f_j f_k \exp(2\pi i (\mathbf{r}_j - \mathbf{r}_k) \cdot \mathbf{h}),\ \forall \mathbf{h} \in \mathbb{Z}^3, \end{aligned} \tag{1.45}$$

$$\begin{aligned} I(-\mathbf{h}) &= \mathbf{F}(-\mathbf{h}) \cdot \mathbf{F}^*(-\mathbf{h}) \\ &= \left(\sum_{j=1}^{N} f_j \exp(-2\pi i \mathbf{r}_j \cdot \mathbf{h}) \right) \left(\sum_{k=1}^{N} f_k \exp(2\pi i \mathbf{r}_k \cdot \mathbf{h}) \right) \\ &= \sum_{k=1}^{N} \sum_{j=1}^{N} f_j f_k \exp(2\pi i (\mathbf{r}_k - \mathbf{r}_j) \cdot \mathbf{h}),\ \forall \mathbf{h} \in \mathbb{Z}^3, \end{aligned} \tag{1.46}$$

and therefore

$$I(\mathbf{h}) = I(-\mathbf{h}),\ \forall \mathbf{h} \in \mathbb{Z}^3. \tag{1.47}$$

This is equivalent to saying that the reciprocal lattice is always centro-symmetric, i.e., in case there is no anomalous dispersion present, there is always a centre of symmetry in the reciprocal lattice, even if there is no such centre of symmetry in the crystal structure [Dove, 2003].

Space group symmetries

The unit cells can contain symmetry elements defining the arrangement of molecules inside the cell. The combination of the 14 Bravais lattices [Stöcker, 1994] with the 32 point groups leads to 230 different space group symmetries [Aroyo et al., 2006; Borchardt-Ott, 2002]. The possible point group symmetry operations are reflection, rotation and rotoinversion, screw axis and glide plane symmetry operations [Borchardt-Ott, 2002]. Due to the existence of asymmetric C-atoms in proteins and nucleic acids, only some rotation, screw axis and glide plane symmetries can occur in those, reducing the number of possible space groups for proteins to 65 ones [Borchardt-Ott, 2002].

These symmetries appear also in the electron density distribution. Let us assume, that the distribution $\rho(\mathbf{x}),\ \mathbf{x} \in \mathbf{V}$ displays the symmetries of a space group

$$\Gamma = \{(\mathbf{R}_\nu, \mathbf{t}_\nu)\}_{\nu=1}^{n_{sym}}. \tag{1.48}$$

Here $n_{sym} \in \mathbb{N}$ is the number of symmetries in a unit cell, $\mathbf{R}_\nu$, $\nu \in \{1, \ldots, n_{sym}\}$ are rotational matrices and $\mathbf{t}_\nu$, $\nu \in \{1, \ldots, n_{sym}\}$ translation vectors [Borchardt-Ott, 2002; Zachariasen, 1945].
This means, that the following equation holds for all $\mathbf{x} \in \mathbf{V}$, $\nu \in \{1, \ldots, n_{sym}\}$:

$$\rho(\mathbf{R}_\nu \mathbf{x} + \mathbf{t}_\nu) = \rho(\mathbf{x}). \tag{1.49}$$

In Figure 1.8 protein G is shown, which shows the symmetries of the space group P 21 21 21. This space group is defined by the rotational matrices

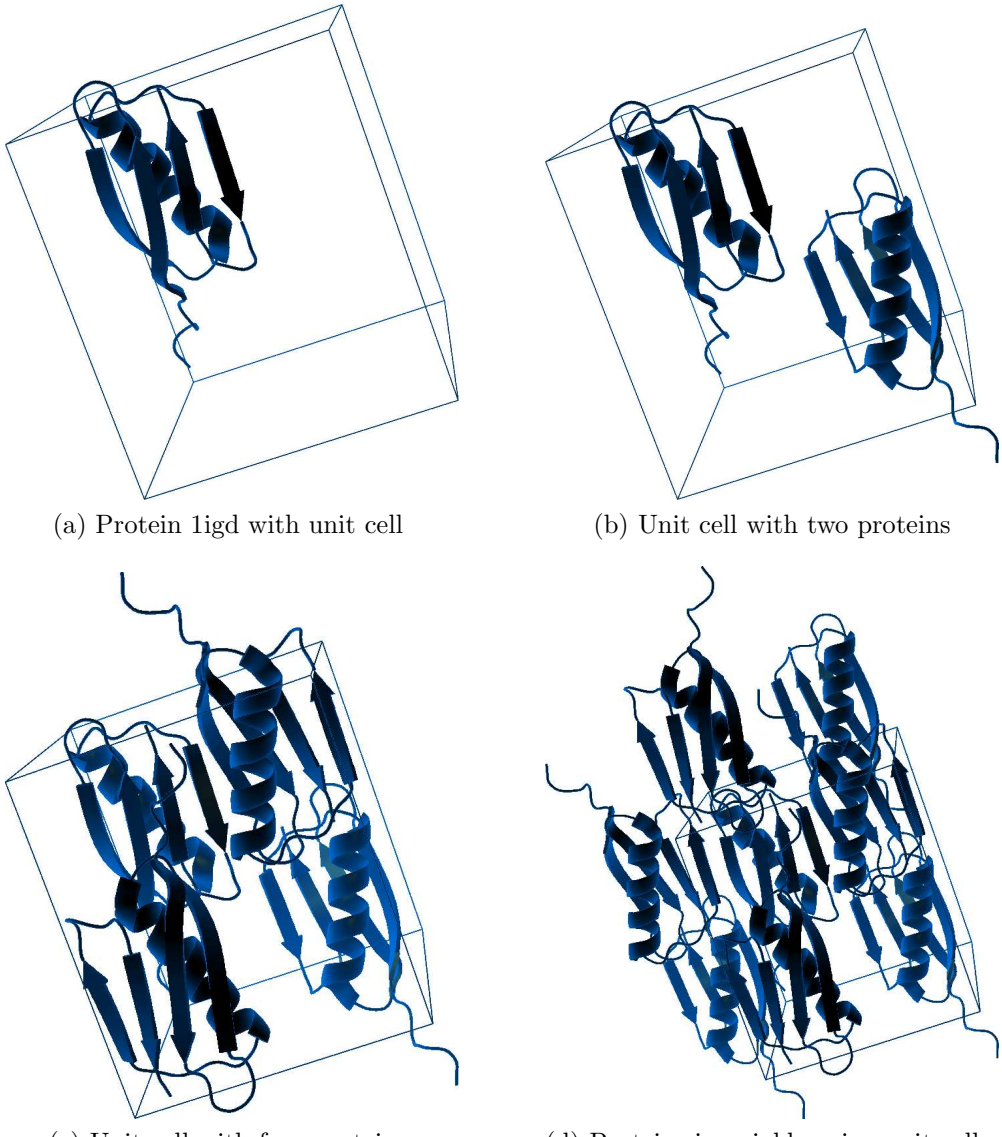
$$\begin{aligned} \mathbf{R}_1 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{R}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \\ \mathbf{R}_3 &= \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad \mathbf{R}_4 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \tag{1.50}$$

and the translation vectors

$$\mathbf{t}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{t}_2 = \begin{pmatrix} 0.5 \\ 0.5 \\ 0 \end{pmatrix}, \quad \mathbf{t}_3 = \begin{pmatrix} 0 \\ 0.5 \\ 0.5 \end{pmatrix}, \quad \mathbf{t}_4 = \begin{pmatrix} 0.5 \\ 0 \\ 0.5 \end{pmatrix}. \tag{1.51}$$

From (1.49) and equations (1.37) and (1.36) one can derive (cf. [Waser, 1955]):

$$\begin{aligned} & \exp(2\pi i(\mathbf{h} \cdot \mathbf{t}_\nu))\mathbf{F}(\mathbf{R}_\nu^T \mathbf{h}) \\ = \ & \exp(2\pi i(\mathbf{h} \cdot \mathbf{t}_\nu)) \int_V \rho(\mathbf{x}) \exp(2\pi i(R_\nu^T \mathbf{h} \cdot \mathbf{x}))d\mathbf{x} \\ = \ & \int_V \rho(\mathbf{x}) \exp(2\pi i(\mathbf{h} \cdot \mathbf{R}_\nu \mathbf{x})) \exp(-2\pi i(\mathbf{h} \cdot \mathbf{t}_\nu))d\mathbf{x} \\ = \ & \int_V \rho(\mathbf{x}) \exp(2\pi i(\mathbf{h} \cdot \mathbf{R}_\nu \mathbf{x} + \mathbf{t}_\nu))d\mathbf{x} \\ = \ & \int_V \rho(\mathbf{R}_\nu \mathbf{x} + \mathbf{t}_\nu) \exp(2\pi i(\mathbf{h} \cdot \mathbf{R}_\nu \mathbf{x} + \mathbf{t}_\nu))d\mathbf{x} \\ = \ & \mathbf{F}(\mathbf{h}),\ \forall \mathbf{h} \in \mathbf{\Pi},\ \nu \in \{1, \ldots, n_{sym}\}. \end{aligned}$$



(a) Protein 1igd with unit cell

(b) Unit cell with two proteins

(c) Unit cell with four proteins

(d) Proteins in neighbouring unit cells

Figure 1.8: Crystallized protein G belonging to space group P 21 21 21

So the structure factors show the symmetry

$$\mathbf{F}(\mathbf{R}_\nu^T\mathbf{h}) = \mathbf{F}(\mathbf{h})\exp(-2\pi i(\mathbf{h}\cdot\mathbf{t}_\nu)),\ \forall\mathbf{h}\in\mathbf{\Pi},\ \nu\in\{1,\ldots,n_{sym}\}. \tag{1.52}$$

Equation (1.52) implies for all $\mathbf{h}\in\mathbf{\Pi}$ and $\nu\in\{1,\ldots,n_{sym}\}$ the conditions

$$\begin{aligned} \text{if } \mathbf{R}_\nu^T\mathbf{h} &= \mathbf{h}, \\ \text{and therefore } \mathbf{F}(\mathbf{h}) &= \mathbf{F}(\mathbf{R}_\nu^T\mathbf{h}) = \mathbf{F}(\mathbf{h})\exp(-2\pi i(\mathbf{h}\cdot\mathbf{t}_\nu)), \\ \text{then } \mathbf{F}(\mathbf{h}) &= 0 \text{ or } \exp(-2\pi i(\mathbf{h}\cdot\mathbf{t}_\nu)) = 1, \\ \text{respectively } \mathbf{F}(\mathbf{h}) &= 0 \text{ or } (\mathbf{h}\cdot\mathbf{t}_\nu)\in\mathbb{Z}. \end{aligned} \tag{1.53}$$

So, if $\mathbf{R}_\nu^T\mathbf{h} = \mathbf{h}$ and $(\mathbf{h},\mathbf{t}_\nu)\notin\mathbb{Z}$ then $\mathbf{F}(\mathbf{h}) = 0$. For the intensities $I(\mathbf{h})$ one gets:

$$\begin{aligned} I(\mathbf{h}) &= \mathbf{F}(\mathbf{h})\cdot\mathbf{F}^*(\mathbf{h}) \\ &= \mathbf{F}(\mathbf{h})\exp(-2\pi i(\mathbf{h}\cdot\mathbf{t}_\nu))\cdot\mathbf{F}(\mathbf{h})\exp(2\pi i(\mathbf{h}\cdot\mathbf{t}_\nu)) \\ &= \mathbf{F}(\mathbf{R}_\nu^T\mathbf{h})\cdot\mathbf{F}^*(\mathbf{R}_\nu^T\mathbf{h}) \\ &= I(\mathbf{R}_\nu^T\mathbf{h}),\ \forall\mathbf{h}\in\mathbf{\Pi},\ \nu\in\{1,\ldots,n_{sym}\}. \end{aligned} \tag{1.54}$$

These symmetry properties of the electron density and the structure factors provide information that can be used in the reconstruction process.

Chapter 2

A Binary Integer-Programming Approach to the Phase Problem

> The lack of real contact between mathematics and biology is either a tragedy, a scandal, or a challenge, it is hard to decide which.
>
> Gian-Carlo Rota, 1986

> Mathematics is biology's next microscope, only better; biology is mathematics' next physics, only better.
>
> J.E. Cohen, 2004

2.1 Solving the phase problem with a binary integer programming approach

In practice, the search for electron-density values is usually restricted to the values calculated in the nodes of a grid in the unit cell of the regarded molecule and to the corresponding subset of structure factors. The resulting numerical errors can be neglected if the grid dimensions are large enough and high-resolution diffraction data is involved [Lunin et al., 2002b]. If one is only interested in the position and the shape of the region with density values above a certain level, a binary function can be used to represent this region. This restriction may reduce the number of solutions of the phase problem significantly [Lunin et al., 2002b]. At a low resolution, this function represents the part of the unit cell occupied by protein molecules, namely a molecular mask or an envelope. When the resolution

is increased this binary function may represent elements of the secondary structure or the trace of the polypeptide chain. The equations connecting the searched binary electron density values with the experimental structure factors will not be strictly valid after the restriction and require some corrections. This chapter is based on the work of Vladimir Lunin, Alexandre Urzhumtsev and Alexander Bockmayr, published in [Lunin et al., 2002b].

2.1.1 The Sayre equation

Restricting the density values $\rho(\mathbf{x}),\ \mathbf{x} \in V$ to $\{0,1\}$ is equivalent to the condition:

$$\rho(\mathbf{x}) = \rho^2(\mathbf{x}),\ \forall \mathbf{x} \in V. \tag{2.1}$$

The *Sayre equation*, introduced in [Sayre, 1952], plays an important role in X-ray-crystallography. It can be derived from the formula (1.36) describing the electron density.
Squaring any function is equivalent to self-convoluting its array of structure factors, so for the squared electron density function we get the Patterson-function: [Patterson, 1931], [Patterson, 1934], [Patterson, 1935]

$$\begin{aligned} \rho^2(\mathbf{x}) &= \left(\frac{1}{V_{cell}} \sum_{\mathbf{h}\in\Pi} \mathbf{F}(\mathbf{h}) \exp(-2\pi i(\mathbf{h}\cdot\mathbf{x})) \right)^2 && (2.2) \\ &= \frac{1}{V_{cell}^2} \sum_{\mathbf{h}\in\Pi} \sum_{\mathbf{h}'\in\Pi} \mathbf{F}(\mathbf{h}') \exp(-2\pi i(\mathbf{h}'\cdot\mathbf{x})) \mathbf{F}(\mathbf{h}-\mathbf{h}') \exp(-2\pi i((\mathbf{h}-\mathbf{h}')\cdot\mathbf{x})) \\ &= \frac{1}{V_{cell}^2} \sum_{\mathbf{h}\in\Pi} \sum_{\mathbf{h}'\in\Pi} \mathbf{F}(\mathbf{h}') \mathbf{F}(\mathbf{h}-\mathbf{h}') \exp(-2\pi i(\mathbf{h}\cdot\mathbf{x})),\ \forall \mathbf{x} \in V. && (2.3) \end{aligned}$$

Since $\rho^2(\mathbf{x})$ like $\rho(\mathbf{x})$ is a periodic function, using the Fourier transform it can be written as

$$\rho^2(\mathbf{x}) = \frac{1}{V_{cell}} \sum_{\mathbf{h}\in\Pi} \mathbf{F^{sq}}(\mathbf{h}) \exp(-2\pi i(\mathbf{h}\cdot\mathbf{x})),\ \forall \mathbf{x} \in V, \tag{2.4}$$

where $\mathbf{F^{sq}}$ is the structure factor of the squared electron density.
So the structure factors of the squared electron density $\mathbf{F^{sq}}$ are given by

$$\mathbf{F^{sq}}(\mathbf{h}) = \frac{1}{V_{cell}} \sum_{\mathbf{h}'\in\Pi} \mathbf{F}(\mathbf{h}') \mathbf{F}(\mathbf{h}-\mathbf{h}'),\ \forall \mathbf{h} \in \Pi. \tag{2.5}$$

Setting $\mathbf{h}-\mathbf{h}' = \mathbf{h}''$ and using equations (2.1) and (1.36) we can derive the Sayre equation for the structure factors:

$$\mathbf{F}(\mathbf{h}) = \mathbf{F^{sq}}(\mathbf{h}) = \frac{1}{V_{cell}} \sum_{\mathbf{h}'} \mathbf{F}(\mathbf{h}') \mathbf{F}(\mathbf{h}''),\ \forall \mathbf{h} \in \Pi. \tag{2.6}$$

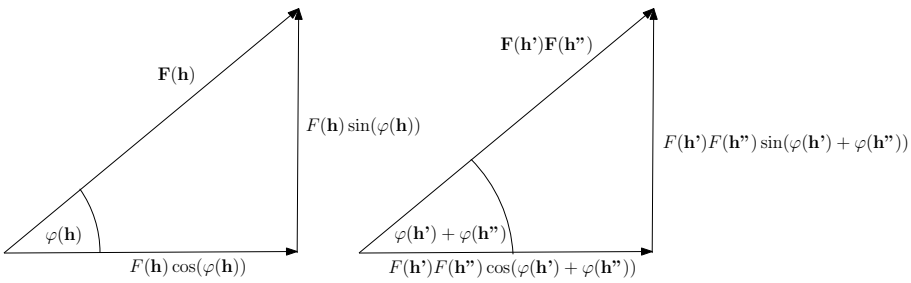


Figure 2.1: Tangent-formula

This equation leads us to the so-called tangent-formula [Karle and Hauptman, 1956]:

$$\tan\varphi(\mathbf{h}) = \frac{\sum\limits_{\mathbf{h}\in\Pi} F(\mathbf{h}')F(\mathbf{h}'')\sin(\varphi(\mathbf{h}')+\varphi(\mathbf{h}''))}{\sum\limits_{\mathbf{h}\in\Pi} F(\mathbf{h}')F(\mathbf{h}'')\cos(\varphi(\mathbf{h}')+\varphi(\mathbf{h}''))},\ \forall\mathbf{h}\in\Pi, \tag{2.7}$$

which is illustrated in figure 2.1.

2.1.2 Grid structure factors

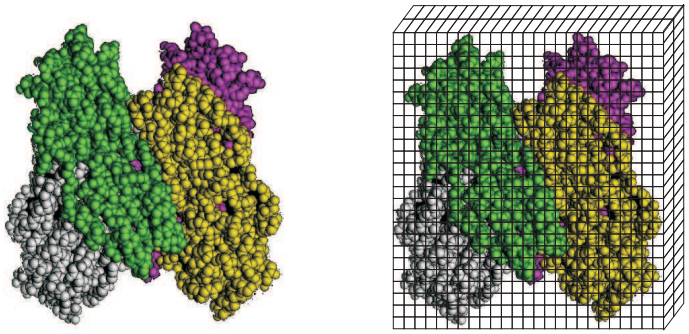
Instead of calculating the electron density distribution in the whole unit cell, a grid covering the unit cell will be introduced (cf. Figure 2.2). Using discrete Fourier transforms only the electron density on grid points will be calculated. The chosen division numbers along the unit-cell axes represent the resolution of the searched electron density map.

Consider a grid $\Pi = [0, M_1 - 1] \times [0, M_2 - 1] \times [0, M_3 - 1] \subseteq \mathbb{Z}^3$, $M_1, M_2, M_3 \in \mathbb{N}$, where $M = M_1M_2M_3$ is the total number of grid points. Let $\mathbf{M}$ be the diagonal matrix $\mathbf{M} = \mathrm{diag}(M_1, M_2, M_3)$. Given the values $\mathbf{y_j}$ of a periodic function f on the grid points $\mathbf{j}$, i.e.,

$$\mathbf{y_j} = f(\frac{j_1}{M_1}, \frac{j_2}{M_2}, \frac{j_3}{M_3}), \quad \mathbf{j} = (j_1, j_2, j_3) \in \Pi \tag{2.8}$$

the *three dimensional discrete Fourier transform* $\mathcal{F}$ calculates the Fourier coefficients of a trigonometric polynomial interpolating f in these grid points [Walker, 1991]:

$$\mathcal{F}(\mathbf{h}) = \frac{1}{M}\sum_{\mathbf{j}\in\Pi} \mathbf{y_j}\exp(2\pi i(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})),\ \forall\mathbf{h}\in\Pi. \tag{2.9}$$



(a) Protein (b) Discretisation

Figure 2.2: Discretisation of a protein crystal structure

The values $\mathbf{y_j}, \mathbf{j} \in \Pi$, can be recovered from the Fourier coefficients $\mathcal{F}(\mathbf{h}), \mathbf{h} \in \Pi$, by the *inverse discrete Fourier transform*:

$$\mathbf{y_j} = \sum_{\mathbf{h} \in \Pi} \mathcal{F}(\mathbf{h}) \exp(-2\pi i(\mathbf{h} \cdot \mathbf{M}^{-1}\mathbf{j})), \ \forall \mathbf{j} \in \Pi. \tag{2.10}$$

The values of the electron density function $\rho(\mathbf{x}), \ \mathbf{x} \in V$ at the grid points are described by the *grid electron density function*

$$\rho_g(\mathbf{j}) = \rho(\mathbf{M}^{-1}\mathbf{j}), \ \forall \mathbf{j} \in \Pi. \tag{2.11}$$

The *grid structure factor* $\mathbf{F}_g(\mathbf{h})$ is defined by the discrete Fourier transform

$$\mathbf{F}_g(\mathbf{h}) = \frac{1}{M} \sum_{\mathbf{j} \in \Pi} \rho_g(\mathbf{j}) \exp(2\pi i(\mathbf{h} \cdot \mathbf{M}^{-1}\mathbf{j})), \ \forall \mathbf{h} \in \Pi. \tag{2.12}$$

Just like the structure factors, the grid structure factors are complex numbers and thus can be written in the form

$$\mathbf{F}_g(\mathbf{h}) = F_g(\mathbf{h}) \exp(i\varphi_g(\mathbf{h})), \tag{2.13}$$

with magnitude $F_g(\mathbf{h}) = |\mathbf{F}_g(\mathbf{h})|$ and phase $\varphi_g(\mathbf{h}) \in [0, 2\pi[$.

If the grid structure factors are known, the grid electron densities $\rho_g(\mathbf{j}) : \Pi \to \mathbb{R}$ can be restored using the inverse discrete Fourier transform.

$$\rho_g(\mathbf{j}) = \sum_{\mathbf{h} \in \Pi} \mathbf{F}_g(\mathbf{h}) \exp(-2\pi i(\mathbf{h} \cdot \mathbf{M}^{-1}\mathbf{j})), \ \forall \mathbf{j} \in \Pi, \tag{2.14}$$

Here, only the values of $\mathbf{F}_g(\mathbf{h}), \ \mathbf{h} \in \Pi$ have to be known to calculate the grid electron density values, while in (1.36) all values of $\mathbf{F}(\mathbf{h}), \ \mathbf{h} \in \mathbb{Z}^3$ would be needed

to calculate the electron density values. Obviously, from the experiments only finitely many intensity values and thus finitely many structure factor values can be obtained.
As the electron density $\rho(\mathbf{j})$ is periodic, by its Definition 2.11 the grid electron density $\rho_g(\mathbf{j})$ is also periodic (cf. equation (1.33)):

$$\rho_g(\mathbf{j}+\mathbf{M}\mathbf{k}) = \rho_g(\mathbf{j}),\ \forall \mathbf{k} \in \mathbb{Z}^3,\ \forall \mathbf{j} \in \Pi. \tag{2.15}$$

As already mentioned in (1.34), the grid structure factors, represented by the inverse discrete Fourier transform of the electron density, are also periodic, satisfying

$$\mathbf{F}_g(\mathbf{h}+\mathbf{M}\mathbf{k}) = \mathbf{F}_g(\mathbf{h}),\ \forall \mathbf{k} \in \mathbb{Z}^3,\ \forall \mathbf{h} \in \Pi. \tag{2.16}$$

So, the definition of $\rho_g(\mathbf{j})$, as well as $\mathbf{F}_g(\mathbf{h})$, at the grid points can be periodically extended to all integer vectors $\mathbf{j} \in \mathbb{Z}^3$ and $\mathbf{h} \in \mathbb{Z}^3$.

2.1.3 Structure factors vs. grid structure factors

In order to clarify the relation between the structure factors and the grid structure factors, we start with equation (2.12) and use (1.36), see also [TenEyck, 1977]:

$$\begin{aligned} V_{cell}\mathbf{F}_g(\mathbf{h}) &= \frac{V_{cell}}{M}\sum_{\mathbf{j}\in\Pi}\rho(\mathbf{M}^{-1}\mathbf{j})\exp(2\pi i(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) \\ &= \frac{1}{M}\sum_{\mathbf{j}\in\Pi}\Big(\sum_{\mathbf{p}\in\mathbb{Z}^3}\mathbf{F}(\mathbf{p})\exp(-2\pi i(\mathbf{p}\cdot(\mathbf{M}^{-1}\mathbf{j})))\Big)\cdot\exp(2\pi i(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) \\ &= \frac{1}{M}\sum_{\mathbf{p}\in\mathbb{Z}^3}\mathbf{F}(\mathbf{p})\sum_{\mathbf{j}\in\Pi}\exp(2\pi i((\mathbf{h}-\mathbf{p})\cdot\mathbf{M}^{-1}\mathbf{j})) \\ &= \sum_{\mathbf{k}\in\mathbb{Z}^3}\mathbf{F}(\mathbf{h}+\mathbf{M}\mathbf{k}). \end{aligned} \tag{2.17}$$

The last equation holds due to

$$\sum_{\mathbf{j}\in\Pi}\exp\left(2\pi i\left((\mathbf{h}-\mathbf{p})\cdot\mathbf{M}^{-1}\mathbf{j}\right)\right) = \begin{cases} \mathbf{M}, & \text{if } \mathbf{h}-\mathbf{p}=\mathbf{M}\mathbf{k},\ \text{for } \mathbf{k}\in\mathbb{Z}^3 \\ 0, & \text{otherwise.} \end{cases} \tag{2.18}$$

Introducing $R(\mathbf{h}) \stackrel{def}{=} \frac{M}{V_{cell}}\sum_{\mathbf{k}\in\mathbb{Z}^3\setminus\{0\}}\mathbf{F}(\mathbf{h}+\mathbf{M}\mathbf{k})$ we can write

$$\mathbf{F}_g(\mathbf{h}) = \frac{\mathbf{F}(\mathbf{h})}{V_{cell}} + \frac{R(\mathbf{h})}{M}. \tag{2.19}$$

The value of $R(\mathbf{h})$ depends on the magnitudes and phases of all structure factors and is generally unknown. But, it may be negligibly small if the grid is fine enough and if the indexes $\mathbf{h}$ are relatively small in comparison with the grid dimensions. Still, it may be significant if one of the indices is close to $\frac{M_1}{2}$, $\frac{M_2}{2}$ or $\frac{M_3}{2}$, cf. [Lunin et al., 2002b].

2.1.4 Inequalities for the grid electron density values

Using (2.12) grid density values $\rho_g(\mathbf{j})$ can be obtained from the grid structure factors $\mathbf{F}_g(\mathbf{h})$ by solving the system of equations

$$\mathbf{F}_g(\mathbf{h}) = \frac{1}{M} \sum_{\mathbf{j} \in \Pi} \rho_g(\mathbf{j}) \, \exp(2\pi i(\mathbf{h} \cdot \mathbf{M}^{-1}\mathbf{j})), \ \forall \mathbf{h} \in \Pi. \tag{2.20}$$

The relation (2.19) between the unknown grid structure factors $\mathbf{F}_g(\mathbf{h})$ and the structure factors $\mathbf{F}(\mathbf{h})$, whose magnitudes can be observed in the X-ray experiment, will now be used to estimate a linear programming formulation to solve the phase problem.
Given an upper bound $\varepsilon_1(\mathbf{h}) \in \mathbb{R}_{\geq 0}$, satisfying

$$|R(\mathbf{h})| \leq \varepsilon_1(\mathbf{h}), \ \forall \mathbf{h} \in \Pi, \tag{2.21}$$

the following system of inequalities for the grid density function can be derived:

$$\Big| \sum_{\mathbf{j} \in \Pi} \rho_g(\mathbf{j}) \exp(2\pi i(\mathbf{h} \cdot \mathbf{M}^{-1}\mathbf{j})) - \frac{M}{V_{cell}} \mathbf{F}(\mathbf{h}) \Big| \leq \varepsilon_1(\mathbf{h}), \ \forall \mathbf{h} \in \Pi \tag{2.22}$$

2.2 Recovering the phases

Now, further constraints restricting the possible phases of the structure factors $\mathbf{F}(\mathbf{h})$ will be deduced.

Assuming first that the magnitudes of the grid structure factors were known (which normally will not be the case, as the experiments provide only the magnitudes of the real structure factors), the following equality system can be derived. The

formula for the grid structure factors (2.12) yields

$$M\mathbf{F}_g(\mathbf{h}) = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \exp(2\pi i(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})), \quad \forall \mathbf{h}\in\Pi \tag{2.23}$$

$$M\cdot F_g(\mathbf{h})\exp(i\varphi_g(\mathbf{h})) = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \exp(2\pi i(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})), \quad \forall \mathbf{h}\in\Pi \tag{2.24}$$

$$M\cdot F_g(\mathbf{h})\cdot(\cos(\varphi_g(\mathbf{h})) + i\cdot\sin(\varphi_g(\mathbf{h}))) = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) + i\cdot\sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j}))), \quad \forall \mathbf{h}\in\Pi \tag{2.25}$$

and thus

$$0 = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j})\cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - M\cdot F_g(\mathbf{h})\cdot\cos(\varphi_g(\mathbf{h})),\ \forall \mathbf{h}\in\Pi \tag{2.26}$$

$$0 = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j})\sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - M\cdot F_g(\mathbf{h})\cdot\sin(\varphi_g(\mathbf{h})),\ \forall \mathbf{h}\in\Pi. \tag{2.27}$$

Taking into account, that only the real structure factors are known, equality (2.19) has to be considered.

$$M\cdot\mathbf{F}_g(\mathbf{h}) = \frac{M}{V_{cell}}\mathbf{F}(\mathbf{h}) + R(\mathbf{h}),\ \forall \mathbf{h}\in\Pi. \tag{2.28}$$

Replacing $M\cdot F_g(\mathbf{h})$ in the latter equation (2.23) and using (2.26) and (2.27) yields

$$\frac{M}{V_{cell}}\mathbf{F}(\mathbf{h}) + R(\mathbf{h}) = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j})\exp(2\pi i(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})),\ \forall \mathbf{h}\in\Pi \tag{2.29}$$

$$R(\mathbf{h}) = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j})(\cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) + i\cdot\sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j}))) - \frac{M}{V_{cell}}F(\mathbf{h})\cdot(\cos(\varphi(\mathbf{h})) + i\cdot\sin(\varphi(\mathbf{h}))),\ \forall \mathbf{h}\in\Pi \tag{2.30}$$

and thus

$$-|R(\mathbf{h})| \le \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j})\cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \frac{M}{V_{cell}}F(\mathbf{h})\cos(\varphi(\mathbf{h})) \le |R(\mathbf{h})|,\ \forall \mathbf{h}\in\Pi, \tag{2.31}$$

$$-|R(\mathbf{h})| \le \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j})\sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \frac{M}{V_{cell}}F(\mathbf{h})\sin(\varphi(\mathbf{h})) \le |R(\mathbf{h})|,\ \forall \mathbf{h}\in\Pi. \tag{2.32}$$

Estimating $R(\mathbf{h})$ by $|R(\mathbf{h})| \leq \varepsilon_1(\mathbf{h})$ according to (2.21) leads to

$$\begin{aligned} -\varepsilon_1(\mathbf{h}) &\leq \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) \\ &\quad -\frac{M}{V_{cell}} F(\mathbf{h}) \cos(\varphi(\mathbf{h})) \leq \varepsilon_1(\mathbf{h}),\ \forall \mathbf{h} \in \Pi, \qquad (2.33) \\ -\varepsilon_1(\mathbf{h}) &\leq \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) \\ &\quad -\frac{M}{V_{cell}} F(\mathbf{h}) \sin(\varphi(\mathbf{h})) \leq \varepsilon_1(\mathbf{h}),\ \forall \mathbf{h} \in \Pi. \qquad (2.34) \end{aligned}$$

2.2.1 Symmetries

Like already mentioned in section 1.2.7, different symmetries can appear in the electron density distribution. Possible symmetry groups of a crystal can be estimated from the diffraction pattern and are thus known in advance. This additional information can be used in the phasing process. Every crystal can be assigned to exactly one of the 230 crystallographic space groups. However, due to measuring uncertainties a unique assignment only based on experimental data is not always possible.
The density distribution $\rho(\mathbf{x})$ of a crystal displays the symmetries of a space group,

$$\Gamma = \{(\mathbf{R}_\nu, \mathbf{t}_\nu)\}_{\nu=1}^{n_{sym}},\ n_{sym} \in \mathbb{N}, \qquad (2.35)$$

with $\mathbf{R}_\nu$ being a rotation matrix and $\mathbf{t}_\nu$ a translation vector if and only if the following holds, cf. also section 1.2.7:

$$\rho(\mathbf{R}_\nu \mathbf{x} + \mathbf{t}_\nu) = \rho(\mathbf{x}),\ \forall \mathbf{x} \in \mathbb{R}^3,\ \forall \nu \in \{1, \ldots, n_{sym}\}. \qquad (2.36)$$

From (2.36) and (1.37) the following symmetries for the structure factors can be derived [Waser, 1955]:

$$\mathbf{F}(\mathbf{h}) = \exp(2\pi i(\mathbf{h}\cdot\mathbf{t}_\nu))\ \mathbf{F}(\mathbf{R}_\nu^T \mathbf{h}),\ \forall \mathbf{h} \in \Pi,\ \forall \nu \in \{1, \ldots, n_{sym}\}. \qquad (2.37)$$

If $\mathbf{R}_\nu^T \mathbf{h} = -\mathbf{h}$ for some ν, $\mathbf{h}$ is called *centric reflection*, otherwise it is called *acentric*.

2.2.2 Properties of grid structure factors

The problem with introducing binary variables is, that the X-ray experiment provides magnitudes corresponding to a real electron density and not to a binary

function approximating it. But in [Lunin et al., 2002b] it was shown that there is a high correlation between initial structure factors and those calculated from binary envelopes.

Using (2.37) and the Hermitian symmetry $\mathbf{F}(-\mathbf{h}) = \mathbf{F}^*(\mathbf{h})$ of the structure factors, cf. (1.29), the following phase restrictions for centric reflections can be obtained:

$$\text{if } \mathbf{R}_\nu^T\mathbf{h} = -\mathbf{h}, \text{ then } \varphi(\mathbf{h}) = \psi(\mathbf{h}) \text{ or } \varphi(\mathbf{h}) = \psi(\mathbf{h}) + \pi, \ \forall \mathbf{h} \in \Pi, \tag{2.38}$$

whereby $\psi(\mathbf{h}) \stackrel{def}{=} \pi(\mathbf{h} \cdot \mathbf{t}_\nu), \ \forall \mathbf{h} \in \Pi$. This can be derived from equation (2.37):

$$\begin{aligned} \mathbf{F}(\mathbf{R}_\nu^T\mathbf{h}) &= \mathbf{F}(\mathbf{h}) \exp(-2\pi i(\mathbf{h} \cdot \mathbf{t}_\nu)), \\ &\quad \forall \mathbf{h} \in \Pi \qquad (2.39) \\ \Rightarrow F(\mathbf{R}_\nu^T\mathbf{h}) \cdot \exp(i\varphi(\mathbf{R}_\nu^T\mathbf{h})) &= F(\mathbf{h}) \cdot \exp(-2\pi i(\mathbf{h} \cdot \mathbf{t}_\nu)) \cdot \exp(i\varphi(\mathbf{h})), \\ &\quad \forall \mathbf{h} \in \Pi \qquad (2.40) \\ \Rightarrow F(-\mathbf{h}) \cdot \exp(i\varphi(-\mathbf{h})) &= F(\mathbf{h}) \cdot \exp(-2\pi i(\mathbf{h} \cdot \mathbf{t}_\nu)) \cdot \exp(i\varphi(\mathbf{h})), \\ &\quad \forall \mathbf{h} \in \Pi. \qquad (2.41) \end{aligned}$$

With the help of the Hermitian symmetry of the structure factors and the phases (1.43) one sees

$$\begin{aligned} F(\mathbf{h}) \cdot \exp(-i\varphi(\mathbf{h})) &= F(\mathbf{h}) \cdot \exp(-2\pi i(\mathbf{h} \cdot \mathbf{t}_\nu)) \cdot \exp(i\varphi(\mathbf{h})), \\ &\quad \forall \mathbf{h} \in \Pi \qquad (2.42) \\ \Rightarrow \exp(-i\varphi(\mathbf{h})) &= \exp(-2\pi i(\mathbf{h} \cdot \mathbf{t}_\nu)) \cdot \exp(i\varphi(\mathbf{h})), \ \forall \mathbf{h} \in \Pi \\ \Rightarrow \exp(-2i\varphi(\mathbf{h})) &= \exp(-2\pi i(\mathbf{h} \cdot \mathbf{t}_\nu)), \ \forall \mathbf{h} \in \Pi \\ \Rightarrow \exp(-2i\varphi(\mathbf{h})) &= \exp(-2i\psi(\mathbf{h})), \ \forall \mathbf{h} \in \Pi \\ \Rightarrow \varphi(\mathbf{h}) = \psi(\mathbf{h}) &\ \vee \ \varphi(\mathbf{h}) = \psi(\mathbf{h}) + \pi, \ \forall \mathbf{h} \in \Pi. \end{aligned}$$

So, if the reflection is centric, only two values of the phase, $\psi(\mathbf{h})$ or $\psi(\mathbf{h}) + \pi$, with $\psi(\mathbf{h})$ being known, are possible. Thus, a new variable $\alpha(\mathbf{h}) \in \{0, 1\}$ can be introduced, representing the phase ambiguity, which yields

$$\mathbf{F}(\mathbf{h}) = F(\mathbf{h}) \exp(i\varphi(\mathbf{h})) = F(\mathbf{h})(2\alpha(\mathbf{h}) - 1) \exp(i\psi(\mathbf{h})), \ \forall \mathbf{h} \in \Pi. \tag{2.43}$$

In the inequality system (2.22), $\mathbf{F}(\mathbf{h})$ can be replaced:

$$\begin{aligned} \Big| \sum_{\mathbf{j} \in \Pi} \rho_g(\mathbf{j}) \exp(2\pi i(\mathbf{h} \cdot \mathbf{M}^{-1}\mathbf{j})) &- \frac{M}{V_{cell}} F(\mathbf{h})(2\alpha(\mathbf{h}) - 1) \exp(i\psi(\mathbf{h})) \Big| \\ &\leq \varepsilon_1(\mathbf{h}), \ \forall \mathbf{h} \in \Pi. \qquad (2.44) \end{aligned}$$

Taking real and imaginary parts, this results in the following inequalities for centric reflections $h \in \Pi$:

$$\begin{aligned} \Big| \sum_{\mathbf{j}\in\Pi} \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j}))\rho_g(\mathbf{j}) \;&-\; (2\alpha(\mathbf{h})-1)\frac{M}{V_{cell}}F(\mathbf{h})\cos\psi(\mathbf{h})\Big| \\ &\leq\; \varepsilon_1(\mathbf{h}),\ \forall\mathbf{h}\in\Pi, \qquad (2.45) \\ \Big| \sum_{\mathbf{j}\in\Pi} \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j}))\rho_g(\mathbf{j}) \;&-\; (2\alpha(\mathbf{h})-1)\frac{M}{V_{cell}}F(\mathbf{h})\sin\psi(\mathbf{h})\Big| \\ &\leq\; \varepsilon_1(\mathbf{h}),\ \forall\mathbf{h}\in\Pi. \qquad (2.46) \end{aligned}$$

As the structure factor magnitudes $F(\mathbf{h})$ are known from experiment, these inequalities are linear in $\rho_g(\mathbf{j})$ and $\alpha(\mathbf{h})$.

2.2.3 Acentric reflections

For the acentric reflections, the phase can take any value from 0 to 2π. [Lunin et al., 2002b] suggests for this case to restrict the phase of the structure factor to one of four possible values $\overline{\varphi}(\mathbf{h}) \in \left\{\pm\frac{\pi}{4},\ \pm\frac{3\pi}{4}\right\},\ \forall\mathbf{h}\in\Pi$. So, $\cos(\overline{\varphi}(\mathbf{h}))$ and $\sin(\overline{\varphi}(\mathbf{h}))$ take the values $\pm\frac{1}{\sqrt{2}}$, for all $\mathbf{h}\in\Pi$. This simplification results in a phase sampling error $\varepsilon_2(\mathbf{h}) \in \mathbb{R}_{\geq 0}$, which can be estimated by

$$\varepsilon_2(\mathbf{h}) \leq \frac{1}{\sqrt{2}},\ \forall\mathbf{h}\in\Pi. \qquad (2.47)$$

We introduce two new variables $\alpha(\mathbf{h}), \beta(\mathbf{h}) \in \{-1, 1\}$ for every grid point $\mathbf{h}\in\Pi$. Taking the phase error $\varepsilon_2(\mathbf{h})$ for the phase values $\varphi(\mathbf{h}),\ \forall\mathbf{h}\in\Pi$ into account yields

$$\begin{aligned} \cos\varphi(\mathbf{h}) &= \alpha(\mathbf{h})\frac{1}{\sqrt{2}} \pm \varepsilon_2(\mathbf{h}),\ \forall\mathbf{h}\in\Pi, \qquad (2.48) \\ \sin\varphi(\mathbf{h}) &= \beta(\mathbf{h})\frac{1}{\sqrt{2}} \pm \varepsilon_2(\mathbf{h}),\ \forall\mathbf{h}\in\Pi. \qquad (2.49) \end{aligned}$$

Assuming that the magnitudes of the grid structure factors are known and starting with equations (2.26) and (2.27) leads to the following equalities for acentric reflections $\mathbf{h}\in\Pi$:

$$\begin{aligned} 0 = \sum_{\mathbf{j}\in\Pi} \quad & \rho_g(\mathbf{j})\cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) \\ & -M\cdot F_g(\mathbf{h})\cdot\left(\alpha(\mathbf{h})\frac{1}{\sqrt{2}} \pm \varepsilon_2(\mathbf{h})\right), \quad \forall\mathbf{h}\in\Pi \qquad (2.50) \end{aligned}$$

$$0 = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - M\cdot F_g(\mathbf{h})\cdot\left(\beta(\mathbf{h})\frac{1}{\sqrt{2}} \pm \varepsilon_2(\mathbf{h})\right), \quad \forall \mathbf{h}\in\Pi. \tag{2.51}$$

This equation system would be linear in $\rho_g(\mathbf{j})$ if the grid structure factors magnitudes were known.
The system can be changed into a binary integer program by replacing the variables $\alpha(\mathbf{h})$ and $\beta(\mathbf{h})$ by binary variables $y_R(\mathbf{h}) \in \{0,1\}$ and $y_I(\mathbf{h}) \in \{0,1\}$, satisfying

$$\begin{aligned}&\alpha(\mathbf{h}) = 2y_R(\mathbf{h}) - 1\\ &\Rightarrow\ (\alpha(\mathbf{h}) = -1 \Leftrightarrow y_R(\mathbf{h}) = 0,\ \alpha(\mathbf{h}) = 1 \Leftrightarrow y_R(\mathbf{h}) = 1), \quad \forall \mathbf{h}\in\Pi\end{aligned} \tag{2.52}$$

$$\begin{aligned}&\beta(\mathbf{h}) = 2y_I(\mathbf{h}) - 1\\ &\Rightarrow\ (\beta(\mathbf{h}) = -1 \Leftrightarrow y_I(\mathbf{h}) = 0,\ \beta(\mathbf{h}) = 1 \Leftrightarrow y_I(\mathbf{h}) = 1), \quad \forall \mathbf{h}\in\Pi.\end{aligned} \tag{2.53}$$

Thus,

$$0 = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - M\cdot F_g(\mathbf{h})\cdot\left((2y_R(\mathbf{h}) - 1)\frac{1}{\sqrt{2}} \pm \varepsilon_2(\mathbf{h})\right), \quad \forall \mathbf{h}\in\Pi \tag{2.54}$$

$$0 = \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - M\cdot F_g(\mathbf{h})\cdot\left((2y_I(\mathbf{h}) - 1)\frac{1}{\sqrt{2}} \pm \varepsilon_2(\mathbf{h})\right), \quad \forall \mathbf{h}\in\Pi. \tag{2.55}$$

This results in

$$\begin{aligned}&-M\cdot F_g(\mathbf{h})\varepsilon_2(\mathbf{h})\\ \leq\ &\sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}M\cdot F_g(\mathbf{h})y_R(\mathbf{h}) + M\cdot F_g(\mathbf{h})\frac{1}{\sqrt{2}}\\ \leq\ &M\cdot F_g(\mathbf{h})\varepsilon_2(\mathbf{h}),\ \forall \mathbf{h}\in\Pi,\end{aligned} \tag{2.56}$$

$$\begin{aligned}&-M\cdot F_g(\mathbf{h})\varepsilon_2(\mathbf{h})\\ \leq\ &\sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}M\cdot F_g(\mathbf{h})y_I(\mathbf{h}) + M\cdot F_g(\mathbf{h})\frac{1}{\sqrt{2}}\\ \leq\ &M\cdot F_g(\mathbf{h})\varepsilon_2(\mathbf{h}),\ \forall \mathbf{h}\in\Pi\end{aligned} \tag{2.57}$$

and considering $\varepsilon_2(\mathbf{h}) \leq \frac{1}{\sqrt{2}}$ yields

$$\begin{aligned} -M \cdot F_g(\mathbf{h})\sqrt{2} \leq & \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) \\ & -\sqrt{2}M \cdot F_g(\mathbf{h}) y_R(\mathbf{h}) \leq 0, \quad \forall \mathbf{h} \in \Pi, \end{aligned} \tag{2.58}$$

$$\begin{aligned} -M \cdot F_g(\mathbf{h})\sqrt{2} \leq & \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) \\ & -\sqrt{2}M \cdot F_g(\mathbf{h}) y_I(\mathbf{h}) \leq 0, \quad \forall \mathbf{h} \in \Pi. \end{aligned} \tag{2.59}$$

In the usual case that only the magnitudes of the real, but not the binary structure factors are known, the calculation, here starting with inequalities (2.33) and (2.34) is quite similar. The binary variables $y_R(\mathbf{h}) \in \{0,1\}$ and $y_I(\mathbf{h}) \in \{0,1\}$ are introduced for the phase sampling, the phase sampling error $\varepsilon_2(\mathbf{h})$ is estimated by

$$\varepsilon_2(\mathbf{h}) \leq \frac{1}{\sqrt{2}}, \ \forall \mathbf{h} \in \Pi. \tag{2.60}$$

$$\begin{aligned} -\varepsilon_1(\mathbf{h}) \ \leq & \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) \\ & -\frac{M}{V_{cell}} F(\mathbf{h}) \left((2y_R(\mathbf{h}) - 1) \frac{1}{\sqrt{2}} \pm \varepsilon_2(\mathbf{h}) \right) \\ \leq & \ \varepsilon_1(\mathbf{h}), \ \forall \mathbf{h} \in \Pi, \end{aligned} \tag{2.61}$$

$$\begin{aligned} -\varepsilon_1(\mathbf{h}) \ \leq & \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) \\ & -\frac{M}{V_{cell}} F(\mathbf{h}) \left((2y_I(\mathbf{h}) - 1) \frac{1}{\sqrt{2}} \pm \varepsilon_2(\mathbf{h}) \right) \\ \leq & \ \varepsilon_1(\mathbf{h}), \ \forall \mathbf{h} \in \Pi. \end{aligned} \tag{2.62}$$

Thus,

$$\begin{aligned} & -\varepsilon_1(\mathbf{h}) - \frac{M}{V_{cell}} F(\mathbf{h}) \varepsilon_2(\mathbf{h}) \\ \leq & \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}\frac{M}{V_{cell}} F(\mathbf{h}) y_R(\mathbf{h}) \\ & + \frac{1}{\sqrt{2}} \frac{M}{V_{cell}} F(\mathbf{h}) \leq \varepsilon_1(\mathbf{h}) + \frac{M}{V_{cell}} F(\mathbf{h}) \varepsilon_2(\mathbf{h}), \ \forall \mathbf{h} \in \Pi, \end{aligned} \tag{2.63}$$

$$\begin{aligned} & -\varepsilon_1(\mathbf{h}) - \frac{M}{V_{cell}} F(\mathbf{h})\varepsilon_2(\mathbf{h}) \\ \leq & \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}\frac{M}{V_{cell}} F(\mathbf{h}) y_I(\mathbf{h}) \\ & + \frac{1}{\sqrt{2}} \frac{M}{V_{cell}} F(\mathbf{h}) \leq \varepsilon_1(\mathbf{h}) + \frac{M}{V_{cell}} F(\mathbf{h})\varepsilon_2(\mathbf{h}),\ \forall \mathbf{h} \in \Pi \end{aligned} \tag{2.64}$$

Estimating the phase sampling error, like in (2.60) leads to

$$\begin{aligned} & -\varepsilon_1(\mathbf{h}) - \sqrt{2}\frac{M}{V_{cell}} F(\mathbf{h}) \\ \leq & \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}\frac{M}{V_{cell}} F(\mathbf{h}) y_R(\mathbf{h}) \\ \leq & \varepsilon_1(\mathbf{h}),\ \forall \mathbf{h} \in \Pi, \end{aligned} \tag{2.65}$$

$$\begin{aligned} & -\varepsilon_1(\mathbf{h}) - \sqrt{2}\frac{M}{V_{cell}} F(\mathbf{h}) \\ \leq & \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}\frac{M}{V_{cell}} F(\mathbf{h}) y_I(\mathbf{h}) \\ \leq & \varepsilon_1(\mathbf{h}),\ \forall \mathbf{h} \in \Pi. \end{aligned} \tag{2.66}$$

As the error $\varepsilon_1(\mathbf{h})$, resulting from the scaling of the observed magnitudes to the magnitudes of the binary functions, can not be estimated easily, [Lunin et al., 2002b] suggest a different way to take this error into account. Here, it is assumed, that the number of non-zero values K in the binary electron density map is already known. K can e.g. be specified by claiming a fixed number of non-zero grid electron density values.
Given binary electron density values on the regarded grid points $B(\mathbf{j}) \in \{0,1\}$, $\forall \mathbf{j} \in \Pi$, the corresponding structure factors $\hat{B}(\mathbf{h}),\ \forall \mathbf{h} \in \Pi$ are given by

$$\hat{B}(\mathbf{h}) = \frac{1}{M} \sum_{\mathbf{j}\in\Pi} B(\mathbf{j}) \exp(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})),\ \forall \mathbf{h} \in \Pi. \tag{2.67}$$

Parseval's relation (1.35), shows

$$M \sum_{\mathbf{h}\in\Pi} \hat{B}^2(\mathbf{h}) = \sum_{\mathbf{j}\in\Pi} B^2(\mathbf{j}). \tag{2.68}$$

Considering that $B(\mathbf{j}) \in \{0,1\}, \forall \mathbf{j} \in \Pi$ and therefore $B(\mathbf{j}) = B^2(\mathbf{j}), \forall \mathbf{j} \in \Pi$ leads to:

$$M \sum_{\mathbf{h}\in\Pi} \hat{B}^2(\mathbf{h}) = \sum_{\mathbf{j}\in\Pi} B^2(\mathbf{j}) = \sum_{\mathbf{j}\in\Pi} B(\mathbf{j}) = K. \tag{2.69}$$

The value of $\hat{B}(\mathbf{0})$ can easily be estimated:

$$\hat{B}(\mathbf{0}) = \frac{1}{M} \sum_{\mathbf{j} \in \Pi} B(\mathbf{j}) = \frac{K}{M}. \tag{2.70}$$

Assuming that the binary structure factors $\hat{B}(\mathbf{h})$ are approximately proportional to the observed structure factor values $F(\mathbf{h})$, a proportionality factor κ can be introduced

$$\hat{B}(\mathbf{h}) \approx \kappa F(\mathbf{h}), \ \forall \mathbf{h} \in \Pi. \tag{2.71}$$

So,

$$\begin{aligned} \kappa^2 \sum_{\substack{\mathbf{h} \in \Pi, \\ h \neq 0}} F^2(\mathbf{h}) &\overset{(2.71)}{\approx} \sum_{\substack{\mathbf{h} \in \Pi, \\ h \neq 0}} \hat{B}^2(\mathbf{h}) = \sum_{\mathbf{h} \in \Pi} \hat{B}^2(\mathbf{h}) - \hat{B}^2(\mathbf{0}) \\ &\overset{(2.69)}{=} \frac{K}{M} - \hat{B}^2(\mathbf{0}) \overset{(2.70)}{=} \frac{K}{M} - \left(\frac{K}{M}\right)^2. \end{aligned} \tag{2.72}$$

The proportionality factor κ can thus be estimated by

$$\kappa = \left(\frac{\frac{K}{|M|} - \left(\frac{K}{|M|}\right)^2}{\sum\limits_{\substack{\mathbf{h} \in \Pi \\ \mathbf{h} \neq 0}} F(\mathbf{h})^2} \right)^{\frac{1}{2}}. \tag{2.73}$$

So, $\kappa \geq 0$ is a scaling factor reflecting that the magnitudes $F^{obs}(\mathbf{h})$ obtained from the analysis of the diffraction pattern correspond to a real electron density distribution, and not to a binary one [Lunin et al., 2002b]. Equations (2.65) and (2.66) can now be reformulated using the scaling factor κ

$$\begin{aligned} -\varepsilon_1(\mathbf{h}) - \sqrt{2}\kappa F(\mathbf{h}) &\leq \sum_{\mathbf{j} \in \Pi} \rho_g(\mathbf{j}) \cos(2\pi(\mathbf{h} \cdot \mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}\kappa F(\mathbf{h}) y_R(\mathbf{h}) \\ &\leq \varepsilon_1(\mathbf{h}), \ \forall \mathbf{h} \in \Pi, \end{aligned} \tag{2.74}$$

$$\begin{aligned} -\varepsilon_1(\mathbf{h}) - \sqrt{2}\kappa F(\mathbf{h}) &\leq \sum_{\mathbf{j} \in \Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h} \cdot \mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}\kappa F(\mathbf{h}) y_I(\mathbf{h}) \\ &\leq \varepsilon_1(\mathbf{h}), \ \forall \mathbf{h} \in \Pi. \end{aligned} \tag{2.75}$$

[Lunin et al., 2002b] suggest also the introduction of a gap

$$0 \leq G \leq 1 \tag{2.76}$$

to take into account a smaller range of solutions. For $G = 1$ this formulation equals the above equations. If additionally we set $\kappa = M$ it equals the formulation for given binary structure factors (2.58) and (2.59).

$$\begin{aligned} -(1+G)\frac{1}{\sqrt{2}}\kappa F(\mathbf{h}) &\leq \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}\kappa F(\mathbf{h}) y_R(\mathbf{h}) \\ &\leq (G-1)\frac{1}{\sqrt{2}}\kappa F(\mathbf{h}),\ \forall \mathbf{h} \in \Pi, \end{aligned} \tag{2.77}$$

$$\begin{aligned} -(1+G)\frac{1}{\sqrt{2}}\kappa &\leq \sum_{\mathbf{j}\in\Pi} \rho_g(\mathbf{j}) \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})) - \sqrt{2}\kappa F(\mathbf{h}) y_I(\mathbf{h}) \\ &\leq (G-1)\frac{1}{\sqrt{2}}\kappa F(\mathbf{h}),\ \forall \mathbf{h} \in \Pi. \end{aligned} \tag{2.78}$$

In [Lunin et al., 2002b], it is shown, that the restriction to only four possible values already yields useful results. In the bachelor thesis [Bode, 2010], the improvement in the solution quality by using the eight different phase values, $\overline{\varphi}(\mathbf{h}) \in \left\{0,\ \pm\frac{\pi}{4},\ \pm\frac{\pi}{2},\ \pm\frac{3\pi}{4}, \pi\right\},\ \forall \mathbf{h} \in \Pi$ has been investigated. But, increasing the number of variables also significantly increases the running time of the implementation described in Chapter 6, so there the phase values will be restricted to four possible ones.

2.3 Constraint-based modelling of the phasing problem

2.3.1 Constraint system

In the context of direct phasing, it may be sufficient to find a binary *envelope* of the regarded molecules, i.e., a binary function indicating areas where the electron density is above a certain level [Lunin et al., 2002b]. Using this idea, the unknowns $\rho_g(\mathbf{j})$ may be replaced by binary variables $z_\mathbf{j} \in \{0, 1\}$, for each grid point $\mathbf{j} \in \Pi$. The value of $z_\mathbf{j}$ should be 1 if the electron density $\rho_g(\mathbf{j})$ is above a certain level and 0 otherwise, so the solution of the problem provides a binary envelope of the regarded molecules.

This leads to a system of linear inequalities in 0-1 variables for representing the electron density values at grid points. The following notations are used for simplification (the superscripts R and I stand for the real and imaginary part resp.)

$$a^R_\mathbf{j}(\mathbf{h}) \stackrel{def}{=} \cos(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})),\ a^I_\mathbf{j}(\mathbf{h}) \stackrel{def}{=} \sin(2\pi(\mathbf{h}\cdot\mathbf{M}^{-1}\mathbf{j})), \tag{2.79}$$

For centric reflections, set

$$y_{\mathbf{h}}^R = y_{\mathbf{h}}^I \stackrel{def}{=} \alpha(\mathbf{h}), \tag{2.80}$$

$$b_{\mathbf{h}}^R \stackrel{def}{=} 2\kappa F(\mathbf{h})\cos\psi(\mathbf{h}),\ \ b_{\mathbf{h}}^I \stackrel{def}{=} 2\kappa F(\mathbf{h})\sin\psi(\mathbf{h}), \tag{2.81}$$

$$c_{\mathbf{h}}^R \stackrel{def}{=} \kappa F(\mathbf{h})\cos\psi(\mathbf{h}),\ \ c_{\mathbf{h}}^I \stackrel{def}{=} \kappa F(\mathbf{h})\sin\psi(\mathbf{h}), \tag{2.82}$$

and for acentric reflections

$$y_{\mathbf{h}}^R = \alpha(\mathbf{h}),\ \ y_{\mathbf{h}}^I \stackrel{def}{=} \beta(\mathbf{h}), \tag{2.83}$$

$$b_{\mathbf{h}}^R \stackrel{def}{=} 2\kappa F(\mathbf{h})2^{-1/2},\ \ b_{\mathbf{h}}^I \stackrel{def}{=} 2\kappa F(\mathbf{h})2^{-1/2}, \tag{2.84}$$

$$c_{\mathbf{h}}^R \stackrel{def}{=} \kappa F(\mathbf{h})2^{-1/2},\ \ c_{\mathbf{h}}^I \stackrel{def}{=} \kappa F(\mathbf{h})2^{-1/2}. \tag{2.85}$$

For further simplification, we introduce

$$A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \stackrel{def}{=} \sum_{\mathbf{j}\in\Pi} a_{\mathbf{j}}^R(\mathbf{h})z_{\mathbf{j}} - \left(b_{\mathbf{h}}^R y_{\mathbf{h}}^R - c_{\mathbf{h}}^R\right), \tag{2.86}$$

$$A^I(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \stackrel{def}{=} \sum_{\mathbf{j}\in\Pi} a_{\mathbf{j}}^I(\mathbf{h})z_{\mathbf{j}} - \left(b_{\mathbf{h}}^I y_{\mathbf{h}}^I - c_{\mathbf{h}}^I\right). \tag{2.87}$$

Then the binary variables $z_{\mathbf{j}}, y_{\mathbf{h}}^R,\ y_{\mathbf{h}}^I$, with $\mathbf{j},\mathbf{h}\in\Pi$ have to satisfy (cf. (2.74) and (2.75))

$$|A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h}))| \le \varepsilon_{\mathbf{h}} \text{ and } |A^I(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h}))| \le \varepsilon_{\mathbf{h}},\ \forall\mathbf{h}\in\Pi, \tag{2.88}$$

where $\varepsilon_{\mathbf{h}} = \varepsilon_1(\mathbf{h})$ for centric and $\varepsilon_{\mathbf{h}} = \varepsilon_1(\mathbf{h}) + \frac{1}{\sqrt{2}}\kappa F(\mathbf{h})$ for acentric reflections. The phase problem then can be stated by the following system of linear inequalities in 0-1 variables:

$$-\varepsilon_{\mathbf{h}} \le A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \le \varepsilon_{\mathbf{h}}, \quad \forall\mathbf{h}\in\Pi, \tag{2.89}$$

$$-\varepsilon_{\mathbf{h}} \le A^I(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \le \varepsilon_{\mathbf{h}}, \quad \forall\mathbf{h}\in\Pi, \tag{2.90}$$

$$z_{\mathbf{j}}, y_{\mathbf{h}}^R,\ y_{\mathbf{h}}^I \in \{0,1\}, \quad \forall\mathbf{h},\mathbf{j}\in\Pi. \tag{2.91}$$

2.4 The objective function

One possibility to work with the inequality system (2.88) is to apply a penalty method, [Brinkmann and Bockmayr, 2008]. Whenever $|A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h}))| > \varepsilon_\mathbf{h}$, we include $|A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h}))|$ as a penalty term, similarly for $|A^I(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h}))|$. If $A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h}))$ respectively $A^I(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h}))$ satisfy $-\varepsilon_\mathbf{h} \leq A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \leq \varepsilon_\mathbf{h}$ respectively $-\varepsilon_\mathbf{h} \leq A^I(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \leq \varepsilon_\mathbf{h}$ the penalties are equal to zero.

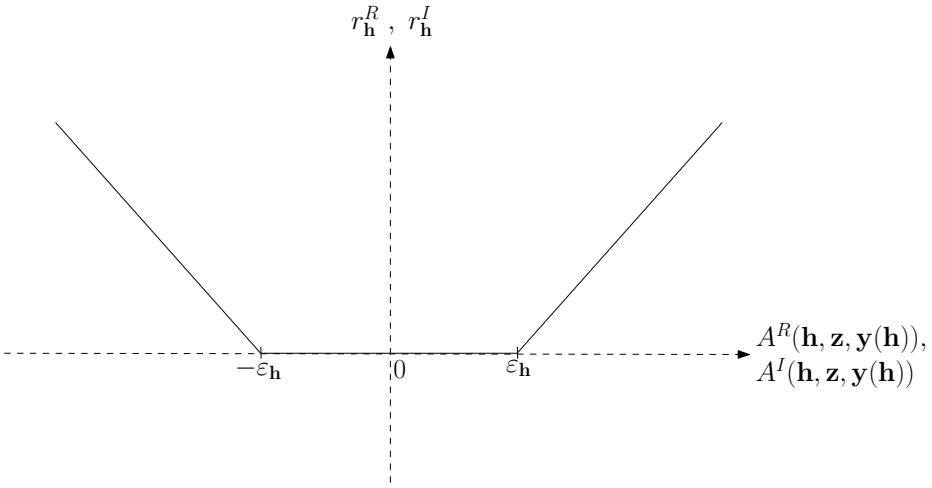


Figure 2.3: Penalty function

This can be modelled as a mixed-integer optimisation problem with the help of additional positive real variables $r^R_\mathbf{h}, r^I_\mathbf{h} \in \mathbb{R}^{\geq 0}, \forall \mathbf{h} \in \Pi$ representing the penalties:

$$
\begin{array}{lll}
\min & \displaystyle\sum_{\mathbf{h}\in\Pi}(r^R_\mathbf{h} + r^I_\mathbf{h}) & (2.92)\\
\text{subject to} & 0 \leq r^R_\mathbf{h},\ 0 \leq r^I_\mathbf{h}, \quad \forall \mathbf{h} \in \Pi, & (2.93)\\
& -\varepsilon_\mathbf{h} - r^R_\mathbf{h} \leq A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \leq \varepsilon_\mathbf{h} + r^R_\mathbf{h}, \quad \forall \mathbf{h} \in \Pi, & (2.94)\\
& -\varepsilon_\mathbf{h} - r^I_\mathbf{h} \leq A^I(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \leq \varepsilon_\mathbf{h} + r^I_\mathbf{h}, \quad \forall \mathbf{h} \in \Pi, & (2.95)\\
& z_\mathbf{j}, y^R_\mathbf{h},\ y^I_\mathbf{h} \in \{0,1\}, \quad \forall \mathbf{h},\mathbf{j} \in \Pi. & (2.96)
\end{array}
$$

Whereas in this approach for every grid point $\mathbf{h} \in \Pi$ penalties $r^R_\mathbf{h}$ and $r^I_\mathbf{h}$ are added and therefore the introduction of $2M$ (M is the number of grid points) penalty variables is required, the following model requires only the introduction of 2 new variables. Here, not the sum of the deviations in the grid points is minimized, but the maximal deviation.

$$
\begin{aligned}
\min \quad & r^R + r^I && (2.97)\\
\text{subject to} \quad & 0 \le r^R,\ 0 \le r^I, \quad \forall \mathbf{h} \in \Pi, && (2.98)\\
& -\varepsilon_{\mathbf{h}} - r^R \le A^R(\mathbf{h}, \mathbf{z}, \mathbf{y}(\mathbf{h})) \le \varepsilon_{\mathbf{h}} + r^R, \quad \forall \mathbf{h} \in \Pi, && (2.99)\\
& -\varepsilon_{\mathbf{h}} - r^I \le A^I(\mathbf{h}, \mathbf{z}, \mathbf{y}(\mathbf{h})) \le \varepsilon_{\mathbf{h}} + r^I, \quad \forall \mathbf{h} \in \Pi, && (2.100)\\
& z_{\mathbf{j}}, y^R_{\mathbf{h}},\ y^I_{\mathbf{h}} \in \{0,1\}, \quad \forall \mathbf{h}, \mathbf{j} \in \Pi. && (2.101)
\end{aligned}
$$

In this chapter a mixed integer programming approach to the phase problem has been developed. Due to the lack of information about the structure factors' phases and the different simplifications made, this approach does not yield an unique solution, but a set of solutions. In order to increase the quality of those solutions, in the following chapters programming formulations to model additional properties of the crystallized proteins are derived.

Chapter 3

3-D Polyominoes

> The purpose of models is not to fit the data, but to sharpen the questions.
>
> Samuel Karlin

In order to increase the quality of the solutions of the optimization problem formulated in section 2.4, we will add information about the geometric properties of proteins by adding appropriate constraints. The problem how to describe topological properties of binary pictures arises also in discrete tomography, so that we can revert to the methods developed in that context. In this chapter some of these methods will be described and extended for three-dimensional pictures.

3.1 Two-dimensional reconstruction problem

The two-dimensional reconstruction problem from orthogonal projections can be described in the following way:

Definition 3.1. (Two-dimensional reconstruction problem)

Given two vectors $\mathbf{h} = (\mathbf{h}_1, \ldots, \mathbf{h}_m) \in \mathbb{N}^m$, $m \in \mathbb{N}$ *and* $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_n) \in \mathbb{N}^n$, $n \in \mathbb{N}$, *find a binary matrix* $\mathbf{X} \in \{0,1\}^{m \times n}$, *also called binary picture, satisfying*

$$\sum_{j=1}^{n} \mathbf{X}_{i,j} = \mathbf{h}_i, \quad \forall i \in \{1, \ldots m\} \tag{3.1}$$

$$\sum_{i=1}^{m} \mathbf{X}_{i,j} = \mathbf{v}_j, \quad \forall j \in \{1, \ldots n\} \tag{3.2}$$

The vectors $\mathbf{h}$ *and* $\mathbf{v}$ *are called projections of* $\mathbf{X}$*.*

Some basic properties of binary pictures are, for example, convexity and connectivity [Bockmayr and Hooker, 2005].

Definition 3.2. (Horizontally convex)

A binary picture $\mathbf{X} \in \{0,1\}^{m\times n}$ *is horizontally convex, if and only if for all* $i \in \{1,\ldots,m\}$ *and for all* $j_1, j_2 \in \{1,\ldots,n\}$*:*

$$\mathbf{X}_{i,j_1} = \mathbf{X}_{i,j_2} = 1 \Rightarrow \mathbf{X}_{i,j} = 1, \quad \forall j \in \{j_1,\ldots,j_2\} : \ j_1 < j_2. \tag{3.3}$$

So, the reconstruction problem is to reconstruct the original binary picture from the number of ones in every row and every column of the corresponding matrix.

Definition 3.3. (Vertically convex)

A binary picture $\mathbf{X} \in \{0,1\}^{m\times n}$ *is vertically convex, if and only if for all* $j \in \{1,\ldots,n\}$ *and for all* $i_1, i_2 \in \{1,\ldots,m\}$*:*

$$\mathbf{X}_{i_1,j} = \mathbf{X}_{i_2,j} = 1 \Rightarrow \mathbf{X}_{i,j} = 1, \quad \forall i \in \{i_1,\ldots,i_2\} : \ i_1 < i_2. \tag{3.4}$$

Definition 3.4. (Convex)

A binary picture is called convex, if and only if it is horizontally and vertically convex.

For the description of connectivity, we need to define a neighbour relation.

Definition 3.5. (Adjacency relation)

Two positions $\mathbf{j} = (i,j)$ *and* $\tilde{\mathbf{j}} = (\tilde{i},\tilde{j})$*,* $\mathbf{j},\ \tilde{\mathbf{j}} \in \{1,\ldots,m\}\times\{1,\ldots,n\}$ *in the matrix* $\mathbf{X} \in \{0,1\}^{m\times n\times p}$ *are called adjacent or neighbouring, if and only if*

$$\| \mathbf{j} - \tilde{\mathbf{j}} \|_2 = 1. \tag{3.5}$$

Definition 3.6. (Connected / Polyomino)

A binary picture $\mathbf{X} \in \{0,1\}^{m \times n}$ *is connected, if and only if for all* $\mathbf{i}_s, \mathbf{i}_t \in \{1,\ldots,m\} \times \{1,\ldots,n\}$ *with* $\mathbf{X}_{\mathbf{i}_s} = \mathbf{X}_{\mathbf{i}_t} = 1$ *there exists a sequence* $\mathbf{i}_s, \mathbf{i}_1, \ldots, \mathbf{i}_k, \mathbf{i}_t$, $k \in \mathbb{N}$ *with* $\mathbf{X}_{\mathbf{i}_1} = \ldots = \mathbf{X}_{\mathbf{i}_k} = 1$*, such that every two consecutive points of the sequence are adjacent.*
A connected binary picture is called a polyomino.

The complexity of the reconstruction problem depending on the additional properties of the picture introduced above has been examined by Woeginger [Woeginger, 2001]. Tabular 3.1 shows the complexity of pattern reconstruction under different constraints. These are the polyomino-constraint as well as vertical (v) and horizontal (h) convexity.

	v & h convex	v convex	h convex	no restriction
Polyomino	P	NP-complete	NP-complete	NP-complete
No restriction	NP-complete	NP-complete	NP-complete	P

Table 3.1: Computational complexity of pattern reconstruction [Woeginger, 2001]

Bockmayr and Hooker in [Bockmayr and Hooker, 2005] introduced linear inequalities for the modeling of convexity and connectivity properties as well as constraint programming formulations. They suggested the following integer linear programming formulations.
A binary picture $\mathbf{X} \in \{0,1\}^{m \times n}$ with projections $\mathbf{h} \in \mathbb{N}^m$ and $\mathbf{v} \in \mathbb{N}^n$ is horizontally convex if and only if

$$\mathbf{h}_i \mathbf{X}_{i,k} + \sum_{j=k+\mathbf{h}_i}^{n} \mathbf{X}_{i,j} \leq \mathbf{h}_i, \quad \forall i \in \{1,\ldots,m\},\ \forall k \in \{1,\ldots,n-\mathbf{h}_i\}. \tag{3.6}$$

This inequalities describe that if there is a 1 in row i, the subsequent $\mathbf{h}_i$ entries in the row can be either 0 or 1, but the entries following afterwards all have to be 0. Vertically convexity can be described similarly. A binary picture $\mathbf{X} \in \{0,1\}^{m \times n}$ with projections $\mathbf{h} \in \mathbb{N}^m$ and $\mathbf{v} \in \mathbb{N}^n$ is vertically convex if and only if

$$\mathbf{v}_j \mathbf{X}_{k,j} + \sum_{i=k+\mathbf{v}_j}^{m} \mathbf{X}_{i,j} \leq \mathbf{v}_j, \quad \forall j \in \{1,\ldots,n\},\ k \in \{1,\ldots,m-\mathbf{v}_j\}. \tag{3.7}$$

For the modeling of connectivity it is assumed, that $\mathbf{h}_i, \mathbf{v}_j \geq 1$, $\forall i \in \{1,\ldots,m\}$, $\forall j \in \{1,\ldots,n\}$. In this chapter only models for connectivity of pictures that are

convex in at least one direction will be regarded.
The following formulation of connectivity presented in [Bockmayr and Hooker, 2005] requires the picture to be either horizontally or vertically convex. It is based on the idea, that in the case of horizontal convexity, if there is a convex set of ones in a row, the set of ones in the subsequent one has to overlap with those. If the picture is vertically convex, a convex set of ones in a column has to overlap with the set of ones in the subsequent column.
A horizontally convex binary picture is connected if and only if

$$\sum_{j=k}^{k+\mathbf{h}_i-1} \mathbf{X}_{i,j} - \sum_{j=k}^{k+\mathbf{h}_i-1} \mathbf{X}_{i+1,j} \leq \mathbf{h}_i - 1,$$
$$\forall i \in \{1,\ldots,m-1\},\ \forall k \in \{1,\ldots,n-\mathbf{h}_i+1\}. \tag{3.8}$$

Analogous, a vertically convex binary picture is connected if and only if

$$\sum_{i=k}^{k+\mathbf{v}_j-1} \mathbf{X}_{i,j} - \sum_{i=k}^{k+\mathbf{v}_j-1} \mathbf{X}_{i,j+1} \leq \mathbf{v}_j - 1, \forall j \in \{1,\ldots,n-1\},\ \forall k \in \{1,\ldots,m-\mathbf{v}_j+1\}.$$

A different, quite natural approach for the modeling of connectivity of horizontally convex pictures is the following one. It is based on the idea, that if there is a convex set of subsequent ones in a row, in the next row, not all ones are allowed to be on positions not overlapping with them. A horizontally convex picture is connected if and only if

$$\sum_{j=k}^{k+\mathbf{h}_i-1} \mathbf{X}_{i,j} + \sum_{j=1}^{k-1} \mathbf{X}_{i+1,j} + \sum_{j=k+\mathbf{h}_i}^{n} \mathbf{X}_{i+1,j} < \mathbf{h}_i + \mathbf{h}_{i+1},$$
$$\forall i \in \{1,\ldots,m-1\},\ \forall k \in \{1,\ldots,n-\mathbf{h}_i+1\}. \tag{3.9}$$

Equality of the left and right hand side in the inequality above will only be reached if all entries of row $i+1$ are placed either left or right of all the entries of row i, which would contradict the connectivity property. In every other case the connectivity is fulfilled.
The vertically convex case can be modeled equivalently. A vertically convex picture is connected if and only if

$$\sum_{i=k}^{k+\mathbf{v}_i-1} \mathbf{X}_{i,j} + \sum_{i=1}^{k-1} \mathbf{X}_{i,j+1} + \sum_{i=k+\mathbf{v}_i}^{m} \mathbf{X}_{i,j+1} < \mathbf{v}_i + \mathbf{v}_{i+1},$$
$$\forall j \in \{1,\ldots,n-1\},\ \forall k \in \{1,\ldots,m-\mathbf{v}_j+1\}. \tag{3.10}$$

3.2 Three-dimensional reconstruction problem

Now, the formulations for the two-dimensional reconstruction problem will be extended for three-dimensional reconstruction problems and supplemented by additional constraints.
The three-dimensional reconstruction problem from orthogonal projections is defined analogous to the two-dimensional one:

Definition 3.7. (Three-dimensional reconstruction problem)

Given three matrices $\mathbf{h} \in \mathbb{N}^{n\times p}$, $n,p \in \mathbb{N}$, $\mathbf{v} \in \mathbb{N}^{m\times p}$, $m,p \in \mathbb{N}$ *and* $\mathbf{w} \in \mathbb{N}^{m\times n}$, $m,n \in \mathbb{N}$ *find a three-dimensional binary array* $\mathbf{X} \in \{0,1\}^{m\times n\times p}$, *also called three-dimensional binary picture, satisfying*

$$\sum_{i=1}^{m} \mathbf{X}_{i,j,k} = \mathbf{h}_{j,k},\ \forall j \in \{1,\dots,n\},\ \forall k \in \{1,\dots,p\}, \tag{3.11}$$

$$\sum_{j=1}^{n} \mathbf{X}_{i,j,k} = \mathbf{v}_{i,k},\ \forall i \in \{1,\dots,m\},\ \forall k \in \{1,\dots,p\}, \tag{3.12}$$

$$\sum_{k=1}^{p} \mathbf{X}_{i,j,k} = \mathbf{w}_{i,j},\ \forall i \in \{1,\dots,m\},\ \forall j \in \{1,\dots,n\}. \tag{3.13}$$

The matrices $\mathbf{h}$, $\mathbf{v}$ *and* $\mathbf{w}$ *are called projections of* $\mathbf{X}$.

Once again, convexity will be regarded in each of the main directions:

Definition 3.8. (Convexity of three-dimensional binary pictures)

A binary picture $\mathbf{X} \in \{0,1\}^{m\times n\times p}$ *is convex if it is directionally convex in every direction, i.e., if and only if for all*

$$\mathbf{X}_{i_1,j,k} = \mathbf{X}_{i_2,j,k} = 1 \Rightarrow \mathbf{X}_{i,j,k} = 1,\ \forall i_1,i_2 \in \{1,\dots,m\} :\ i_1 < i_2,$$
$$\forall i \in \{i_1,\dots,i_2\},\ \forall j \in \{1,\dots,n\},\ \forall k \in \{1,\dots,p\}\ \textit{and} \tag{3.14}$$
$$\mathbf{X}_{i,j_1,k} = \mathbf{X}_{i,j_2,k} = 1 \Rightarrow \mathbf{X}_{i,j,k} = 1,\ \forall j_1,j_2 \in \{1,\dots,n\} :\ j_1 < j_2,$$
$$\forall j \in \{j_1,\dots,j_2\},\ \forall i \in \{1,\dots,m\},\ \forall k \in \{1,\dots,p\}\ \textit{and} \tag{3.15}$$
$$\mathbf{X}_{i,j,k_1} = \mathbf{X}_{i,j,k_2} = 1 \Rightarrow \mathbf{X}_{i,j,k} = 1,\ \forall k_1,k_2 \in \{1,\dots,p\} :\ k_1 < k_2,$$
$$\forall k \in \{k_1,\dots,k_2\},\ \forall i \in \{1,\dots,m\},\ \forall j \in \{1,\dots,n\}. \tag{3.16}$$

If only one of the properties (3.14), (3.15) *or* (3.16) *holds, the binary picture is called convex in* $x-$, *respectively* $y-$ *or* $z-$ *direction.*

An integer linear programming formulation for the convexity constraints can be formulated analogous to the two-dimensional case:

- *in x-direction:*

$$\mathbf{h}_{j,k}\mathbf{X}_{l,j,k} + \sum_{i=l+\mathbf{h}_{j,k}}^{m} \mathbf{X}_{i,j,k} \leq \mathbf{h}_{j,k},\ \forall l \in \{1,\ldots,m-\mathbf{h}_{jk}\},\ \forall j \in \{1,\ldots,n\},\ \forall k \in \{1,\ldots,p\}, \tag{3.17}$$

- *in y-direction:*

$$\mathbf{v}_{i,k}\mathbf{X}_{i,l,k} + \sum_{j=l+\mathbf{v}_{i,k}}^{n} \mathbf{X}_{i,j,k} \leq \mathbf{v}_{i,k},\ \forall l \in \{1,\ldots,n-\mathbf{v}_{ik}\},\ \forall i \in \{1,\ldots,m\},\ \forall k \in \{1,\ldots,p\}, \tag{3.18}$$

- *in z-direction:*

$$\mathbf{w}_{i,j}\mathbf{X}_{i,j,l} + \sum_{k=l+\mathbf{w}_{i,j}}^{p} \mathbf{X}_{i,j,k} \leq \mathbf{w}_{i,j},\ \forall l \in \{1,\ldots,p-\mathbf{w}_{i,j}\},\ \forall i \in \{1,\ldots,m\},\ \forall j \in \{1,\ldots,n\}. \tag{3.19}$$

Now, the connectivity-relation will also be introduced for three-dimensional binary pictures. Therefore, a suitable adjacency relation is needed. Each point in the three-dimensional picture is adjacent to its vertical and horizontal as well as to ist upper and lower neighbours. A three dimensional picture is called connected, if and only if the set of ones in the binary picture is connected with respect to the adjacency relation.

Definition 3.9. (Adjacency relation in 3d binary pictures)

Two positions $\mathbf{j}$ *and* $\tilde{\mathbf{j}}$ *in* $\mathbb{N}^{m\times n\times p}$ *in the three-dimensional array* $\mathbf{X} \in \{0,1\}^{m\times n\times p}$ *are called adjacent or neighbouring if and only if*

$$\| \mathbf{j} - \tilde{\mathbf{j}} \|_2 = 1. \tag{3.20}$$

The notation $\mathbf{j}\eth\tilde{\mathbf{j}}$ *will be used for adjacent points* $\mathbf{j}$ *and* $\tilde{\mathbf{j}}$ *in* $\mathbb{N}^{m\times n\times p}$.

Definition 3.10. (Connected 3d binary picture)

A three-dimensional binary picture is connected, if and only if for all $\mathbf{i}_s$ *and for all* $\mathbf{i}_t$*, with* $\mathbf{i}_s,\ \mathbf{i}_t \in \{1,\ldots,m\}\times\{1,\ldots,n\}\times\{1,\ldots,p\}$ *with* $\mathbf{X}_{\mathbf{i}_s} = \mathbf{X}_{\mathbf{i}_t} = 1$*, there exists a sequence* $\mathbf{i}_s, \mathbf{i}_1, \ldots, \mathbf{i}_l, \mathbf{i}_t$ *with* $l \in \mathbb{N}$*, such that* $\mathbf{X}_{\mathbf{i}_1} = \ldots = \mathbf{X}_{\mathbf{i}_l} = 1$ *where every two consecutive elements of the sequence are adjacent.*

In order to model connectivity for $x-$convex pictures, two different approaches have been made. These approaches can directly be reformulated for $y-$ and $z-$convex pictures.
One is an extension of the 2d-model (3.8) and (3.9) of Bockmayr and Hooker, the other is an extension of the new 2d-model introduced in (3.9) and (3.10). In contrast to the 2-dimensional approach, in the 3-dimensional one, an additional direction has to be considered for the modelling of connectivity.
An extension of the 2d-modeling approach in [Bockmayr and Hooker, 2005] leads to constraints to model connectivity in the three-dimensional case.
A three-dimensional x-convex binary picture $\mathbf{X} \in \{0,1\}^{m\times n\times p}$ is connected if and only if for every $j \in \{1,\ldots,n-1\}$ and for every $k \in \{1,\ldots,p-1\}$ with $\mathbf{h}_{j,k} > 0$ and $\mathbf{h}_{j+1,k} + \mathbf{h}_{j,k+1} > 0$ at least one of the following constraints holds:

- *for k fixed:*

$$\sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j,k} - \sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j+1,k} \leq \mathbf{h}_{j,k} - 1,\ \forall l \in \{1,\ldots,m-\mathbf{h}_{j,k}+1\}, \quad (3.21)$$

- *for j fixed:*

$$\sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j,k} - \sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j,k+1} \leq \mathbf{h}_{j,k} - 1,\ \forall l \in \{1,\ldots,m-\mathbf{h}_{j,k}+1\}. \quad (3.22)$$

As the compliance of one of the above inequalities is sufficient, they can be combined in the following way: A three-dimensional x-convex binary picture $\mathbf{X} \in \{0,1\}^{m\times n\times p}$ is connected if and only if for every $j = 1,\ldots,n-1$ and for every $k = 1,\ldots,p-1$ with $\mathbf{h}_{j,k} > 0$ and $\mathbf{h}_{j+1,k} + \mathbf{h}_{j,k+1} > 0$ at least one of the following constraints holds:

$$\sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j,k} - \sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j+1,k} - \sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j,k+1} \leq \mathbf{h}_{j,k} - 1,$$
$$\forall l \in \{1,\ldots,m-\mathbf{h}_{j,k}+1\}. \quad (3.23)$$

These constraints can be reformulated analogous regarding the projections $\mathbf{v}_{i,k}$ or $\mathbf{w}_{i,j}$, if the picture is convex in y- or z-direction.

The second approaches to model connectivity (3.9) and (3.10) in the two-dimensional case can also be extended to the three-dimensional case in order to get a system of constraints to model connectivity for pictures being convex in x-direction.
A three-dimensional x-convex binary picture $\mathbf{X} \in \{0,1\}^{m \times n \times p}$ is connected if and only if for every $j \in \{1, \ldots, n-1\}$ and for every $k \in \{1, \ldots, p-1\}$ with $\mathbf{h}_{j,k} > 0$ and $\mathbf{h}_{j+1,k} + \mathbf{h}_{j,k+1} > 0$ at least one of the following constraints is fulfilled:

$$\sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j,k} + \sum_{i=1}^{l-1} \mathbf{X}_{i,j+1,k} + \sum_{i=l+\mathbf{h}_{j,k}}^{m} \mathbf{X}_{i,j+1,k} \leq \mathbf{h}_{j,k} + \mathbf{h}_{j+1,k} - 1,$$
$$\forall l \in \{1, \ \ldots, m - \mathbf{h}_{j,k} + 1\}, \tag{3.24}$$

$$\sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j,k} + \sum_{i=1}^{l-1} \mathbf{X}_{i,j,k+1} + \sum_{i=l+\mathbf{h}_{j,k}}^{m} \mathbf{X}_{i,j,k+1} \leq \mathbf{h}_{j,k} + \mathbf{h}_{j,k+1} - 1,$$
$$\forall l \in \{1, \ \ldots, m - \mathbf{h}_{j,k} + 1\}. \tag{3.25}$$

These two constraints can also be combined into one ensuring, the a three-dimensional x-convex binary picture $\mathbf{X} \in \{0,1\}^{m \times n \times p}$ is connected. This is fulfilled if and only if for every $j = 1, \ldots, n-1$ and for every $k = 1, \ldots, p-1$ with $\mathbf{h}_{j,k} > 0$ and $\mathbf{h}_{j+1,k} + \mathbf{h}_{j,k+1} > 0$ at least one of the following constraints holds:

$$\sum_{i=l}^{l+\mathbf{h}_{j,k}-1} \mathbf{X}_{i,j,k} + \sum_{i=1}^{l-1} \mathbf{X}_{i,j+1,k} + \sum_{i=l+\mathbf{h}_{j,k}}^{m} \mathbf{X}_{i,j+1,k} + \sum_{i=1}^{l-1} \mathbf{X}_{i,j,k+1} + \sum_{i=l+\mathbf{h}_{j,k}}^{m} \mathbf{X}_{i,j,k+1}$$
$$\leq \mathbf{h}_{j,k} + \mathbf{h}_{j+1,k} + \mathbf{h}_{j,k+1} - 1,$$
$$\forall l \in \{1, \ldots, m - \mathbf{h}_{j,k} + 1\}. \tag{3.26}$$

In this chapter some geometric constraints used in discrete tomography like convexity and connectivity for two- and three-dimensional pictures have been formulated as integer linear programs. The connectivity constraints developed here are only valid for directionally-convex pictures. As proteins in general do not necessarily satisfy this convexity-constraint, in the next chapter geometric constraints not needing this additional restriction are introduced. Also, there can be more than one connected component in a unit cell, so the connectivity constraint will be extended appropriately.

Chapter 4

Additional Geometric Constraints

> The bigger the real-life problems, the greater the tendency for the discipline to retreat into a reassuring fantasy-land of abstract theory and technical manipulation.
>
> Tom Naylor

In general, the linear program for solving the phase problem, introduced in Chapter 2.1, does not have a unique optimal solution but a set of different optimal solutions. In order to reduce the number of those and at the same time increase the quality of the remaining ones, additional constraints can be added. These constraints take into account different geometric properties of crystals. In this chapter two of them will be introduced: one to exclude isolated points in the binary electron density pattern and a connectivity constraint, ensuring that the number of connected components in the unit cell does not exceed a given bound.

4.1 Additional constraints

The electron density distribution is binarised as described in the previous chapters, such that the value of $z_{\mathbf{j}}$ should be 1 if the electron density $\rho(\mathbf{j})$ is above a certain level and 0 otherwise for all grid points $\mathbf{j} \in \Pi$, $\Pi \stackrel{def}{=} [0, M_1 - 1] \times [0, M_2 - 1] \times [0, M_3 - 1] \subseteq \mathbb{Z}^3, \quad M_1 \in \mathbb{N},\ M_2 \in \mathbb{N},\ M_3 \in \mathbb{N}$. So, introducing the cut off-level $\boldsymbol{\kappa}$ the binary electron density values $z_{\mathbf{j}}$ satisfy

$$z_{\mathbf{j}} = \begin{cases} 0, & \rho(\mathbf{j}) \leq \boldsymbol{\kappa} \\ 1, & \rho(\mathbf{j}) > \boldsymbol{\kappa} \end{cases}, \ \forall \mathbf{j} \in \Pi. \tag{4.1}$$

4.1.1 Exclusion of Isolated Points

In the electron density distribution of a protein no peaks of very high or very low electron density occur, if an appropriate resolution is used. This means, in the grid electron density distribution no isolated points of high electron density surrounded by low electron density values as well as no isolated points of low electron density surrounded by high electron density values are expected to occur.
Regarding a set of grid points Π

$$\Pi \stackrel{def}{=} [0, M_1-1]\times[0, M_2-1]\times[0, M_3-1] \subseteq \mathbb{Z}^3, \quad M_1 \in \mathbb{N},\ M_2 \in \mathbb{N},\ M_3 \in \mathbb{N}, \tag{4.2}$$

inside a unit cell, it can be distinguished between border points and interior points of the unit cell (here we assume $M_1,\ M_2,\ M_3 \geq 3$) :

$$\Pi^0 \stackrel{def}{=} [1, M_1 - 2] \times [1, M_2 - 2] \times [1, M_3 - 2] \subseteq \Pi \tag{4.3}$$

is the set of interior points,

$$\delta\Pi \stackrel{def}{=} \Pi \setminus \Pi^0 \subseteq \mathbb{Z}^3 \tag{4.4}$$

is the set of border points. Unit cells are just a part of the whole crystal which is generated by them. So, the grid inside the unit cell can be extended to a grid underlying the whole crystal. In this grid, the border points of one unit cell are also border points of an adjacent one.

Therefore we introduce an extended version of the neighbour relation (3.20):

Definition 4.1. (Neighbour relation)

Two grid points $\mathbf{j} = (j_1, j_2, j_3) \in \Pi$ *and* $\tilde{\mathbf{j}} = (\tilde{j_1}, \tilde{j_2}, \tilde{j_3}) \in \Pi$ *are* neighbours in the extended sense *of the neighbour relation, if and only if:*

$$\left\| \begin{pmatrix} (j_1 + 1 \mod M_1) - (\tilde{j_1} + 1 \mod M_1) \\ (j_2 + 1 \mod M_2) - (\tilde{j_2} + 1 \mod M_2) \\ (j_3 + 1 \mod M_3) - (\tilde{j_3} + 1 \mod M_3) \end{pmatrix} \right\|_2 = 1. \tag{4.5}$$

We will use the notation $\mathbf{j} \mathfrak{n}_{ext} \tilde{\mathbf{j}}$ *for two grid points* $\mathbf{j} \in \Pi$ *and* $\tilde{\mathbf{j}} \in \Pi$ *that are neighbours in the extended sense.*

In figure 4.1 the extended neighbour relation in the whole grid an for one unit cell in two dimensions is illustrated. Neighbours are connected by edges. In the extended neighbour relation also the border points $\mathbf{j} = (0, j_2, j_3)$ and $\tilde{\mathbf{j}} = (M_1 - 1, j_2, j_3)$ are neighboured for all $j_2 \in \{0, \ldots, M_2 - 1\}$ and all $j_3 \in \{0, \ldots, M_3 - 1\}$ as well as $\mathbf{j} = (j_1, 0, j_3)$ and $\tilde{\mathbf{j}} = (j_1, M_2-1, j_3)$ are neighboured for all $j_1 \in \{0, \ldots, M_1-1\}$ and all $j_3 \in \{0, \ldots, M_3 - 1\}$ and $\mathbf{j} = (j_1, j_2, 0)$ and $\tilde{\mathbf{j}} = (j_1, j_2, M_3 - 1)$ are neighboured for all $j_1 \in \{0, \ldots, M_1 - 1\}$ and all $j_2 \in \{0, \ldots, M_2 - 1\}$.

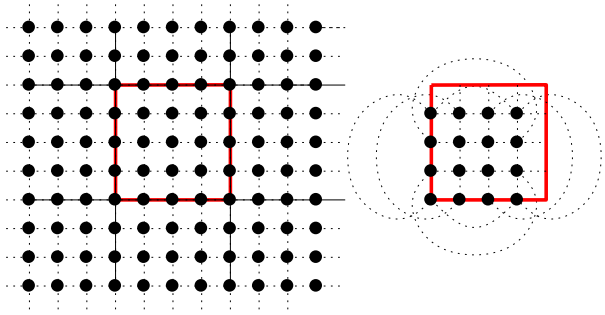


Figure 4.1: Extended neighbour relation

Definition 4.2. (Isolated point)

A binary grid point $z_{\mathbf{j}} \in \Pi$ *is called* isolated point*, if and only if*

$$z_{\mathbf{j}} = 0 \quad \Rightarrow \quad z_{\mathbf{i}} = 1, \ \forall \ \mathbf{i} \mathbf{n}_{ext} \mathbf{j}, \tag{4.6}$$

$$z_{\mathbf{j}} = 1 \quad \Rightarrow \quad z_{\mathbf{i}} = 0, \ \forall \ \mathbf{i} \mathbf{n}_{ext} \mathbf{j}. \tag{4.7}$$

Using appropriate resolutions it can be assumed, there are no peaks in the electron density distribution. This means for the binary case, there are no isolated points $z_{\mathbf{j}}, \ \mathbf{j} \in \Pi$ of value 1 or 0.

Every interior grid point has 6 neighbours and every grid point has 6 neighbours in the extended sense. A grid point $\mathbf{j}$ with value $z_{\mathbf{j}}$ is isolated if and only if all neighbours of $\mathbf{j}$ take a different value. In consequence, there are no isolated points if for $z_{\mathbf{j}} = 1$, it is assured that $\sum_{\tilde{\mathbf{j}} \mathbf{n}_{ext} \mathbf{j}} z_{\tilde{\mathbf{j}}} \geq 1$ and if $z_{\mathbf{j}} = 0$, then $\sum_{\tilde{\mathbf{j}} \mathbf{n}_{ext} \mathbf{j}} z_{\tilde{\mathbf{j}}} \leq 5$.

This leads to the following condition for the exclusion of isolated points. Due to the use of the extended neighbour relation, the condition also excludes isolated border points.

$$-5 \leq z_{\mathbf{j}} - \sum_{\tilde{\mathbf{j}} \mathbf{n}_{ext} \mathbf{j}} z_{\tilde{\mathbf{j}}} \leq 0, \ \forall \mathbf{j} \in \Pi. \tag{4.8}$$

4.1.2 Minimum covering

Another possibility to exclude unwanted solutions is to specify a minimum covering, i.e., a minimum percentage of points of value 1 in the resulting binary picture.

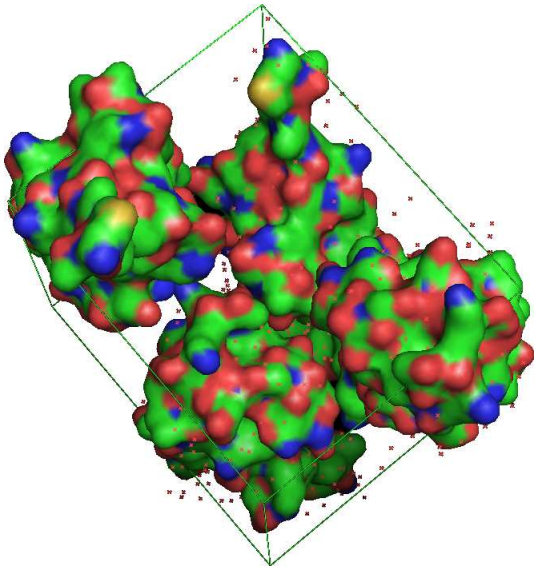


Figure 4.2: Unit cell of Protein G

Given a percentage p specifying the minimum number of grid points with a binary electron density of value 1 in the grid, this constraint can be formulated as follows:

$$\frac{p}{100} \cdot M_1 M_2 M_3 \leq \sum_{\mathbf{j} \in \Pi} z_{\mathbf{j}}, \ \forall \mathbf{j} \in \Pi. \tag{4.9}$$

4.2 Connectivity

4.2.1 Introduction

At low resolution and a high enough high cut-off level $\boldsymbol{\kappa}$, the high-level region $\Omega_{\boldsymbol{\kappa}} \stackrel{def}{=} \{\mathbf{j} : \rho(\mathbf{j}) > \boldsymbol{\kappa}\}$ is expected to consist of a small number of connected components, which should be equal to the number of molecules inside the unit cell. At lower cut-off level these components merge into fewer regions. In some cases the number of components could be smaller than the number of molecules in the unit cell. This happens if neighbouring molecules are close to each other [Lunin et al., 2002a]. So, it is possible to estimate an upper bound for the number of molecules in advance. This property can be used to choose appropriate solutions among the possible solutions and thus increase the quality of the set of solutions. At low resolution this connectivity approach is one of the most efficient methods [Lunin et al., 2000; Lunina et al., 2003; Müller et al., 2006].

4.2.2 Constraint programming

For modeling the connectivity property, constraint programming methods are used. A general constraint programming problem consists of given variables, constraints and, in case an optimization and not only a satisfiability problem is regarded, an objective function. A feasible solution is an assignment of values to the variables satisfying all the given constraints. An optimal solution is a feasible solution of a constraint programming problem for which the objective function takes a maximum respectively minimum value.
Constraint Programming in general can handle arbitrary constraints. One can distinguish between arithmetic constraints, like linear and also non-linear constraints, and symbolic constraints. A symbolic constraint can be every subset of the set of all assignments of values to variables. One symbolic constraint can also include large families of linear constraints. Such symbolic constraints can not only simplify the declaration of constraints, but also improve the efficiency of the solving process, [Apt, 2003; Bockmayr and Hooker, 2005]. Figure 4.3 shows the dependencies of different types of arithmetic constraint problems.

The special types of Constraint Programming Problems shown here, each leading to special solution strategies, are

- **Finite Domain Programming (FD)**
  In Finite Domain Programming, the possible values the given variables can take are restricted to a finite set.

- **Linear Programming (LP)**
  In linear programming, all constraints are linear inequalities and the objective function is also linear, i.e., a linear program can be written in the form

$$\max\{c^T x \mid Ax \leq b\} \quad \text{with } A \in \mathbb{R}^{m \times n},\ b \in \mathbb{R}^m,\ c \in \mathbb{R}^n. \tag{4.10}$$

- **Integer Programming (IP)**
  In integer programming, linear programs are regarded with the additional restriction that all variables have to be integral:

$$\max\{c^T x \mid Ax \leq b\} \quad \text{with } A \in \mathbb{R}^{m \times n},\ b \in \mathbb{R}^m,\ c \in \mathbb{R}^n,\ x \in \mathbb{Z}^n. \tag{4.11}$$

- **Mixed Integer Programming (MIP)**
  In mixed integer programming only some of the variables are restricted to be integral.

- **Binary Integer Programming (BIP)**
  In binary integer programming (BIP) the variables can only take the values 0 or 1.

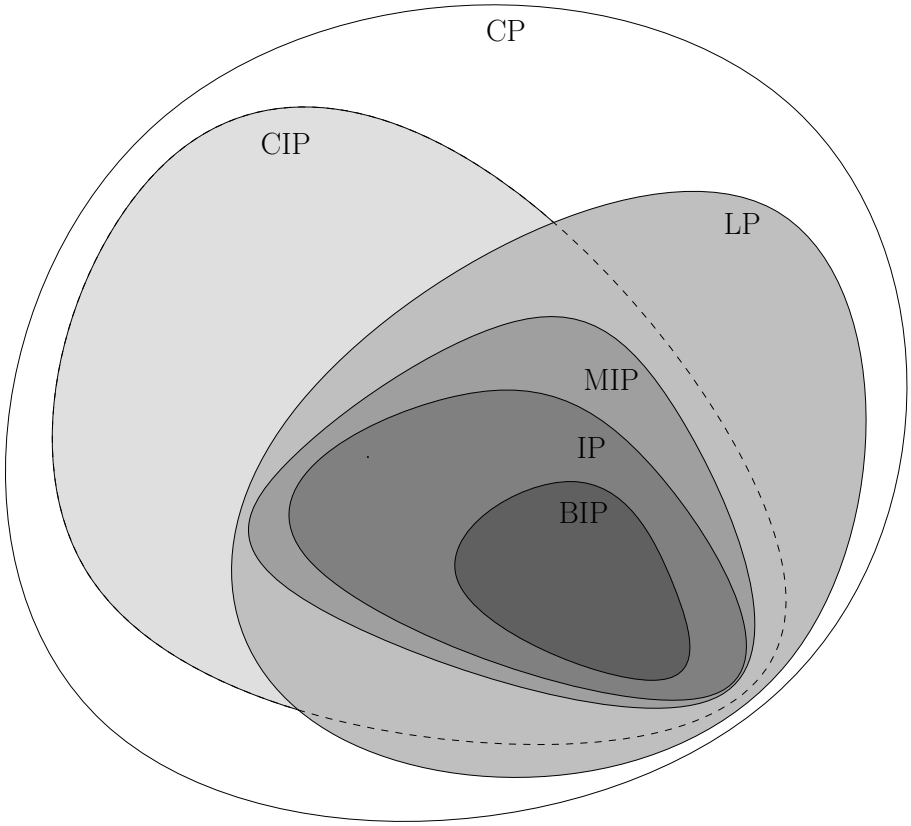


Figure 4.3: CP: Constraint Programming, CIP: Constraint Integer Programming, LP: Linear Programming, MIP: Mixed Integer Programming, IP: Integer Programming, BIP: Binary Integer Programming

4.2.3 Graph-theoretical approach

In this section, integer linear programming formulations to restrict the number of connected components in a unit cell based on graph theory will be deduced. A binary grid electron density distribution satisfies the 'K-component-constraint' if and only if it contains at most $K \in \mathbb{N}$ components. 0-1-integer linear programming formulations for this problem are deduced and different results concerning these different models and the polytopes described by the appropriate inequalities are presented.
First, the necessary graph-theoretical terms will be defined [Grötschel et al., 1988; Korte and Vygen, 2000]. We will only regard undirected graphs, thus an edge (v_1, v_2) for two distinct vertices v_1 and v_2 is equal to the edge (v_2, v_1).

Definition 4.3. (Induced subgraph)

A subgraph $G^* = (V^*, E^*)$ of the graph $G = (V, E)$ is called induced *by the vertices $\overline{V} \subseteq V$, if*

$$V^* = \overline{V} \text{ and } E^* = \{(v_1, v_2) \in E | v_1, v_2 \in \overline{V}\}. \tag{4.12}$$

The subgraph G^* of $G = (V, E)$, induced by the vertices $\overline{V} \subseteq V$ will be denoted by $G[V^*]$, the set of edges E^* in the induced subgraph will be denoted by $E(G[V^*])$.

Definition 4.4. (Walk, Path, Cycle)

Let $G = (V, E)$ be a graph.
A sequence $v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_{k+1}$, such that $k \geq 0$, $v_i \in V$, $\forall i \in \{1, \ldots, k\}$ and $e_i = (v_i, v_{i+1}) \in E$, $\forall i \in \{1, \ldots, k\}$ where all edges are distinct, i.e., $e_i \neq e_j$, $\forall i, j \in \{1, \ldots, k\}$, is called a walk *in G.*
If additionally all vertices are distinct, i.e., $v_i \neq v_j$, $\forall i, j \in \{1, \ldots, k+1\}$, the sequence is called a path.
A walk with identical endpoints $v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_1$ with $v_i \neq v_j$, $\forall i, j \in \{1, \ldots, k\}$ is called cycle *or* subtour.

Definition 4.5. (Cut, cutset)

Let $V^* \subseteq V$ be a subset of the vertices of a graph $G = (V, E)$. Then $\delta(V^*)$ is defined as

$$\delta(V^*) \stackrel{def}{=} \{e = (v_1, v_2) \in E | v_1 \in V^*, v_2 \in V \setminus V^*)\}. \tag{4.13}$$

A subset $E^* \subseteq E$ of the edges of G is called a cut *or* cutset*, if $E^* = \delta(X)$ for some $X \subseteq V$.*
A minimum cut *$\delta_{min}(G)$ in an unweighted graph $G = (V, E)$ is a cut $\delta(V^*)$, $V^* \subseteq V$, such that the number of edges in the cut $|\delta(V^*)|$ is minimal amongst all cuts in G.*

Obviously, $\delta(V^*) = \delta(V \setminus V^*)$.

Definition 4.6. (Connectivity)

A graph $G = (V, E)$ is connected*, if and only if there is a $v_1 - v_2$–path for all $v_1, v_2 \in V$, otherwise G is disconnected.*
The maximal connected subgraphs of G are its (connected) components.
A subset of the vertices in a graph will be called connected, if and only if the subgraph induced by them is connected.

A graph G consists of exactly K components G_i, $i \in \{1, \ldots, K\}$, if $G = \bigcup_{i=1}^{K} G_i$, i.e., G is the disjoint union of the subgraphs G_i of G, and each subgraph is connected.

Notation 4.7.

Let be $V^* \subseteq V$, then $G[V^*]$ is the subgraph of G that is induced by the set of vertices V^*. $E(G[V^*])$ is the set of edges in $G[V^*]$. The vector $y^{E^*} \in \{0,1\}^{|E|}$ is the incidence vector of the set $E^* \subseteq E(G[V^*])$.

$$y_e^{E^*} = \begin{cases} 1, \text{ for } e \in E^* \\ 0, \text{ for } e \notin E^*. \end{cases} \tag{4.14}$$

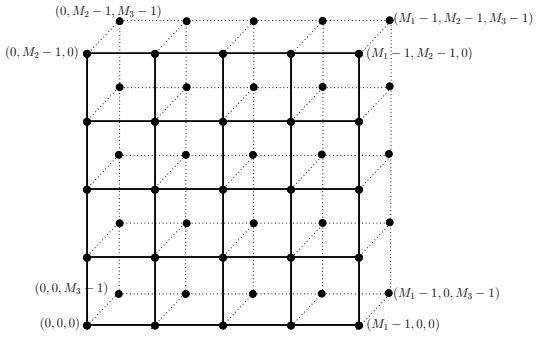


Figure 4.4: The graph $G_\Pi = (V_\Pi, E_\Pi)$

In the crystallographic context, a graph representing properties of the binary grid electron density maps has to be defined.
Let $G_\Pi = (V_\Pi, E_\Pi)$ be an undirected graph with $M = M_1 \cdot M_2 \cdot M_3$ vertices denoted by $v_{\mathbf{j}}$, $\mathbf{j} \in \Pi$. Vertices $v_{\mathbf{j}} \in V_\Pi$ and $v_{\tilde{\mathbf{j}}} \in V_\Pi$ with $\mathbf{j}$ and $\tilde{\mathbf{j}}$ being neighbours are connected by edges, i.e.,

$$E_\Pi = \left\{ e = (v_{\mathbf{j}}, v_{\tilde{\mathbf{j}}}) \mid \mathbf{j} \text{n} \tilde{\mathbf{j}} \right\}. \tag{4.15}$$

Let $V_\Pi^* \subseteq V_\Pi$ be the set of vertices with a corresponding electron density above the cut-off level, i.e., the set of vertices satisfying

$$V_\Pi^* = \{ v_{\mathbf{j}} \mid z_{\mathbf{j}} = 1, \ \mathbf{j} \in \Pi \}. \tag{4.16}$$

We will use the formulation, the three-dimensional binary picture representing the binary grid electron density distribution contains of $K \in \mathbb{N}$ components, if and only if the corresponding graph $G_\Pi^* = (V_\Pi^*, E_\Pi^*)$ contains K components.

Figure 4.5 shows the graph representing the binary grid electron density distribution. Black filled vertices represent grid electron density values above the cut-off level, neighboured black vertices are connected by solid edges.

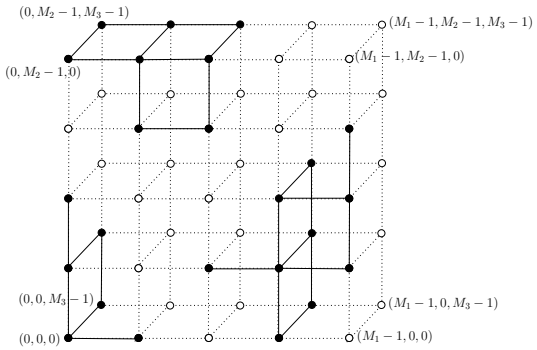


Figure 4.5: The graph $G^*_\Pi = (V^*_\Pi, E^*_\Pi)$

4.2.4 Linear programming

In this section, some basics about linear programming will be presented, which will be needed in the following sections [Korte and Vygen, 2000; Schrijver, 2003]. We start with the definition of a linear programming problem.

Definition 4.8. (Linear programming)

Given a matrix $A \in \mathbb{R}^{m \times n}$ and vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, find a vector $x \in \mathbb{R}^n$

- *such that $Ax \leq b$ and $c^T x$ is maximum,*
- *decide that $\{x \in \mathbb{R}^n : Ax \leq b\}$ is empty,*
- *or decide that for all $\alpha \in \mathbb{R}$ there is an $x \in \mathbb{R}^n$ with $Ax \leq b$ and $c^T x > \alpha$.*

Definition 4.9. (Polyhedra, polytopes)

A subset $P \subseteq \mathbb{R}^n$, $n \in \mathbb{N}$ is called a polyhedron *if there exists a matrix $A \in \mathbb{R}^{m \times n}$, $m \in \mathbb{Z}_{\geq 0}$ and a vector $b \in \mathbb{R}^m$ such that*

$$P = \{x \mid Ax \leq b\}. \tag{4.17}$$

So, P is a polyhedron if and only if it is the intersection of finitely many affine halfspaces.
P is a polytope *if and only if P is a bounded polyhedron.*

Definition 4.10. (Halfspace)

A subset H of $\mathbb{R}^n$ is called an affine halfspace *if $H = \{x \mid c^T x \leq \delta\}$, for some $c \in \mathbb{R}^n$ with $c \neq 0$ and some $\delta \in \mathbb{R}$.*

Definition 4.11. (Hyperplanes, face, vertex)

A hyperplane *in $\mathbb{R}^n$ is a set H with $H = \{x \in \mathbb{R}^n \mid c^T x = \delta\}$ for some $c \in \mathbb{R}^n$ with $c \neq 0$ and some $\delta \in \mathbb{R}$.*
Let $P = \{x \mid Ax \leq b\}$ be a polyhedron in $\mathbb{R}^n$. If c is a non-zero vector and $\delta = \max\{c^T x \mid Ax \leq b\}$, the hyperplane $\{x \mid c^T x = \delta\}$ is called supporting hyperplane *of P.*
A subset F of P is called a face *if $F = P$ or if $F = P \cap H$ for some supporting hyperplane H of P.*
For $z \in P$, let $A_z x \leq b_z$ be the system consisting of those inequalities from $Ax \leq b$ that are satisfied by z with equality. Then z is a vertex *of P if and only if $rank(A_z) = n$.*

Definition 4.12. (Integral polyhedra and polytopes)

A polyhedron P is called integral polyhedron *if each vertex of P is integer.*
A polytope P is called integral polytope *if each vertex of P is integer.*

4.2.5 Spanning tree polytopes

As mentioned above, it is possible to estimate an upper bound for the number of molecules in a unit cell. Thus, it seems to be useful to derive conditions implying that there are at most $K \in \mathbb{N}$ components in the graph representing the molecules in the unit cell. Starting with the presentation of some results concerning the well-known travelling salesman problem, see e.g. [Magnanti and Wolsey, 1995; Nemhauser and Wolsey, 1988] and spanning tree polytopes [Bertsimas and Tsitsiklis, 1997; Korte and Vygen, 2000], such conditions are deduced in the following. Some new results concerning the description of this component number restriction by inequalities and the properties of the resulting polyhedra will be presented.

We start with the basic definitions of forests, trees and spanning trees [Bertsimas and Tsitsiklis, 1997; Korte and Vygen, 2000].

Definition 4.13. (Forest, tree, spanning tree)

A forest *is a graph, which does not contain a cycle. A connected forest is called a* tree. *A* spanning tree *of a graph is a tree containing all nodes of the graph.*

Theorem 4.14. (Spanning tree)

The following statements are equivalent for a graph $G = (V, E)$:

- *$G_T = (V, E_T)$ is a spanning tree of the graph G*
- *$G_T = (V, E_T)$ is a subgraph of G containing $|V| - 1$ edges and no cycles.*
- *$G_T = (V, E_T)$ is a connected subgraph of G containing $|V| - 1$ edges.*

Theorem 4.15. Subtour elimination formulation for spanning trees, Spanning tree polytope

The vector $y \in \{0, 1\}^{|E|}$ is the incidence vector of the edges of a spanning tree $G_T = (V, E_T)$ of the graph $G = (V, E)$, if and only if the following conditions hold:

$$\sum_{e \in E_T} y_e = |V| - 1, \tag{4.18}$$

$$\sum_{e \in E_T(G[X])} y_e \leq |X| - 1, \quad \forall\ X :\ \emptyset \neq X \subset V. \tag{4.19}$$

Given a connected undirected graph $G = (V, E)$, the polytope

$$P_{span} \stackrel{def}{=} \{\ y \in [0, 1]^{|E|} \subseteq \mathbb{R}^{|E|} :\ \sum_{e \in E} y_e = |V| - 1,\ \sum_{e \in E(G[X])} y_e \leq |X| - 1,\ \forall\ X :\ \emptyset \neq X \subset V\ \} \tag{4.20}$$

is the spanning tree polytope as its vertices are exactly the incidence vectors of spanning trees of G. The spanning tree polytope is integral.

The conditions (4.19) are called subtour elimination constraints as they assure the absence of subtours. Every graph satisfying these conditions therefore particularly is a forest.

Theorem 4.16. (Forest polytope), [Korte and Vygen, 2000]

The convex hull of the incidence vectors of all forests in an undirected graph $G = (V, E)$, i.e., the set of convex combinations of these incidence vectors, is the polytope

$$P_{forest} \stackrel{def}{=} \{\, y \in [0,1]^{|E|} \subseteq \mathbb{R}^{|E|} : \sum_{e \in E(G[X])} y_e \leq |X| - 1, \quad \forall\, X :\ \emptyset \neq X \subseteq V \,\}. \tag{4.21}$$

With the help of cuts, a different description of spanning trees can be formulated. Calculating one minimal cut will then directly show if a given vector is an incidence vector of a spanning tree of G.

Proposition 4.17. (Connectivity), [Korte and Vygen, 2000]

A graph $G = (V, E)$ is connected, if and only if $\delta(X) \neq \emptyset,\ \forall \emptyset \neq X \subset V$.

The description of spanning trees using cutsets is equivalent to the characterizations given in Theorem 4.14. It is based on the formulation that a spanning tree is a connected subgraph of an undirected graph $G = (V, E)$ containing $|V| - 1$ edges.

Theorem 4.18. Cutset formulation for spanning trees [Bertsimas and Tsitsiklis, 1997]

The vector $y \in \{0,1\}^{|E|}$ is the incidence vector of the edges of a spanning tree $G_T = (V, E_T)$ of the graph $G = (V, E)$, if and only if the following conditions hold:

$$\sum_{e \in E} y_e = |V| - 1, \tag{4.22}$$

$$\sum_{e \in \delta(X)} y_e \geq 1, \quad \forall\, X :\ \emptyset \neq X \subset V. \tag{4.23}$$

Theorem 4.19. (Cutset polytope), [Korte and Vygen, 2000]

The polytope P_{cut} defined by

$$P_{cut} \stackrel{def}{=} \{\, y \in [0,1]^{|E|} \subseteq \mathbb{R}^{|E|} : \quad \sum_{e \in E} y_e = |V| - 1, \quad \sum_{e \in \delta(X)} y_e \geq 1, \quad \forall\, X :\ \emptyset \neq X \subseteq V \}\,. \tag{4.24}$$

is the cutset polytope.
Then, $P_{cut} \supseteq P_{span}$. In general, the vertices of P_{cut} are fractional.

Both formulations, the subtour elimination (4.18) and (4.19) as well as the cutset formulation (4.22) and (4.23) for spanning trees have a number of constraints which is exponential in the number of nodes.
Theorem 4.19 conveys, that in general, the subtour elimination formulation is stronger than the cutset formulation, since the feasible region of the LP-relaxation of the cutset formulation includes the feasible region of the LP-relaxation of the subtour elimination formulation.
Now, we will use minimal cuts to derive a different description of spanning trees.

Theorem 4.20. [Bertsimas and Tsitsiklis, 1997]

The vector $y \in \{0,1\}^{|E|}$ is the incidence vector of the edges of a spanning tree $G_T = (V, E_T)$ of the graph $G = (V, E)$, if and only if the following conditions hold:

$$\sum_{e \in E} y_e = |V| - 1, \tag{4.25}$$

$$\sum_{e \in \delta(X_0)} y_e \geq 1, \tag{4.26}$$

with some X_0, $\emptyset \subset X_0 \subset E$, such that $\sum_{e \in \delta(X_0)} y_e \stackrel{def}{=} \min_{\emptyset \neq X \subset V} \sum_{e \in \delta(X)} y_e$.

Then $\delta(X_0)$ is a minimum cut in G.

The definition of cuts will now be generalised by the introduction of K-cuts respectively multicuts [Magnanti and Wolsey, 1995]. Based on those, first a polytope equal to P_{span} and later on a description of a polytope describing forests containing at most K trees can be derived.

Definition 4.21. (K-cut, multicut)

Given a partition of the nodes V of an undirected graph $G = (V, E)$ into $K \in \{2, \dots, |V|\}$ disjoint sets V_i, $i \in \{1, \dots, K\}$ with $\bigcup_{i=1}^{K} V_i = V$, the K-cut *(respectively* multicut*) $\delta(V_1, \dots, V_K)$ is the set of all edges connecting two different sets V_i, V_j with $i \neq j \in \{1, \dots, K\}$.*

Definition 4.22. (Minimum K-cut)

A minimum K-cut $\delta_{min}(G)$ *in an unweighted graph $G = (V, E)$ is a K-cut*

$$\delta(V_1, V_2, \dots V_K), V_1, V_2, \dots V_K \subset V, \tag{4.27}$$

such that the number of edges in the cut $|\delta(V_1, V_2, \dots V_K)|$ is minimal amongst all K-cuts in G.
If the edges of the graph G are weighted with weights $w_e \in \mathbb{R} \forall e \in E(G)$, a minimum K-cut $\delta_{min}(G)$ *is a K-cut $\delta(V_1, V_2, \dots V_K)$, $V_1, V_2, \dots V_K \subset V$, such that the sum of the weights of the edges in the cut $\sum_{e \in |\delta(V_1, V_2, \dots V_K)|} w_e$ is minimal amongst all K-cuts in G.*

Let $G_\Pi = (V_\Pi, E_\Pi)$ be an undirected graph with $M = M_1 \cdot M_2 \cdot M_3$ vertices. In Figure 4.6, this graph is shown. The black vertices represent points with an electron density above a certain threshold. The set of black vertices $V_\Pi^* \subseteq V$ induces a subgraph $G_\Pi^* = (V_\Pi^*, E_\Pi^*)$. In this induced subgraph a 3-cut is shown, the cut is empty and therefore also a minimum 3-cut.

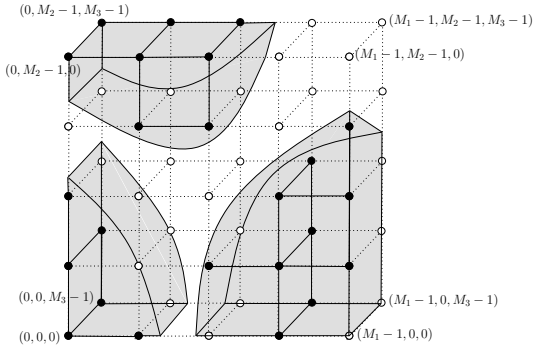


Figure 4.6: A minimum 3-cut

Using K-cuts, another description of the spanning tree polytope can be given.

Definition 4.23. (Multicut polytope)

The multicut polytope P_{mcut}^K *of an undirected graph* $G = (V, E)$ *for a given* $K \in \{1, \ldots, |V|\}$ *is defined by*

$$P_{mcut}^K \stackrel{def}{=} \{\, y \in [0,1]^{|E|} \subseteq \mathbb{R}^{|E|} : \quad \sum_{e \in E} y_e = |V| - 1, \quad \sum_{e \in \delta(V_1, \ldots, V_K)} y_e \geq K - 1, \quad \forall\, \emptyset \neq V_1, \ldots, V_K \subset V \,\}\,. \tag{4.28}$$

4.2.6 Connected components in a graph

Therefore, we regard the problem to partition the node set of a given graph into at most $K \in \mathbb{N}$ subsets, such that the number of edges with end points in two different elements of partitions is minimized.

Based on the presented fundamentals about spanning trees, some new results about spanning forests consisting of $K \in \mathbb{N}$ trees will be proved.

Theorem 4.24. (Subtour elimination formulation)

An undirected graph $G = (V, E)$ *consists of at most* $K \in \mathbb{N}$ *components, if and only if there exists* $y^* \in \{0,1\}^{|E|}$*:*

$$\sum_{e \in E} y_e^* \quad \geq |V| - K, \tag{4.29}$$

$$\sum_{e \in E(G[X])} y_e^* \quad \leq |X| - 1, \quad \forall\, X :\ \emptyset \neq X \subseteq V. \tag{4.30}$$

Corollary 4.25.

The graph $G = (V, E)$ *consists of exactly* $K \in \mathbb{N}$ *components, if and only if there exists* $y^* \in \{0,1\}^{|E|}$ *for which the inequalities* (4.30) *are valid and the first inequality* (4.29) *holds with equality, i.e.*

$$\sum_{e \in E} y_e^* \quad = |V| - K, \tag{4.31}$$

$$\sum_{e \in E(G[X])} y_e^* \quad \leq |X| - 1, \quad \forall\, X :\ \emptyset \neq X \subset V. \tag{4.32}$$

Every $y^* \in \{0,1\}^{|E|}$ *satisfying these constraints is the incidence vector of a spanning forest of* G*.*

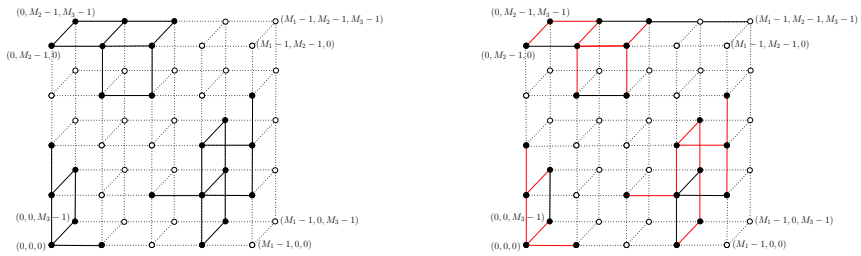


Figure 4.7: a) The graph b) A spanning forest

Figure 4.7 shows a graph with three connected components and a spanning forest in this graph.
The number of the subtour elimination constraints (4.30), i.e.

$$\sum_{e \in E(G[X])} y_e \leq |X| - 1, \quad \forall\, X :\ \emptyset \neq X \subseteq V \tag{4.33}$$

grows exponentially in the number of nodes $|V|$ in the underlying graph G.

Proof: (of Theorem 4.24 and Corollary 4.25)
For every undirected graph $G = (V, E)$ consisting of exactly $\tilde{K}$ components a spanning forest $G_F = (V_F, E_F)$, $V_F \subseteq V$, $E_F \subseteq E$ with $\tilde{K}$ spanning trees $T_I = (V_I, E_I)$, $I \in \{1, \ldots, \tilde{K}\}$, $V_I \subseteq V_F$, $\bigcup_{I=1}^{\tilde{K}} T_I = G_F$ can be found and for y^* being the incidence vector of this spanning forest we get

$$\sum_{e \in E} y_e^* = \sum_{I=1}^{\tilde{K}} \sum_{e \in E_I} y_e^* = \sum_{I=1}^{\tilde{K}} (|V_I| - 1) = |V| - \tilde{K}. \tag{4.34}$$

So, (4.31) is only fulfilled for $K = \tilde{K}$, (4.29) is fulfilled for $\tilde{K} \leq K$.

Let $\tilde{K}$ be the exact number of components in the graph G. Let $C_I = (V_I, E_I)$, $I \in \{1, \ldots \tilde{K}\}$ be the components of G.
An arbitrary y^* satisfying (4.29) and (4.30) respectively (4.32) fulfills

$$|V| - K \leq \sum_{e \in E} y_e^* = \sum_{I=1}^{\tilde{K}} \sum_{e \in C_I} y_e^* \leq \sum_{I=1}^{\tilde{K}} (|V_I| - 1) = |V| - \tilde{K} \tag{4.35}$$

For $K = \tilde{K}$, $\sum_{e \in E} y_e^* = \sum_{I=1}^{\tilde{K}} (|V_I| - 1)$ directly follows, thus y_e^* is the incidence vector of a forest. As $|V| - K \leq |V| - \tilde{K}$, the exact number of components in the graph G is less or equal to K.

Definition 4.26.

$$P_{sub}^{\leq K} \stackrel{def}{=} \{y \in [0,1]^{|E|} \subseteq \mathbb{R}^{|E|} \mid \begin{array}{ll} \sum_{e \in E} & y_e \geq |V| - K, \\ \sum_{e \in E(G[X])} & y_e \leq |X| - 1, \\ & \forall\ X :\ \emptyset \neq X \subseteq V\} \end{array} \quad (4.36)$$

is the polytope corresponding to the subtour elimination formulation for a graph with at most K components.

$$P_{sub}^{=K} \stackrel{def}{=} \{y \in [0,1]^{|E|} \subseteq \mathbb{R}^{|E|} \mid \begin{array}{ll} \sum_{e \in E} & y_e = |V| - K, \\ \sum_{e \in E(G[X])} & y_e \leq |X| - 1, \\ & \forall\ X :\ \emptyset \neq X \subset V\}. \end{array} \quad (4.37)$$

is the polytope corresponding to the subtour elimination formulation for a graph with exactly K components.

Theorem 4.27.

The polytope $P_{sub}^{=K}$ is integral. Its vertices are exactly the incidence vectors of forests with exactly K trees in the underlying graph G.

For the proof of Theorem 4.27, the following results will be used.

Theorem 4.28. [Schrijver, 2003]

Let P be a rational polyhedron in $\mathbb{Q}^n$. Then P is integral if and only if for each $c \in \mathbb{R}^n$, the linear programming problem $\max\{c^T x \mid Ax \leq b\}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ has an integer optimum solution if it is finite.

Lemma 4.29. (Complementary slackness), [Korte and Vygen, 2000]

Let $\min\{c^T y \mid Ay \leq b,\ y \geq 0\}$ *be a primal linear program and* $\max\{z^T b \mid z^T A \geq c, z \geq 0\}$ *its dual. For feasible solutions* y *and* z *the following statements are equivalent:*

- y *and* z *are both optimal solutions of the primal respectively the dual linear program,*
- $y^T(c - z^T A) = 0$ *and* $z^T(b - Ay) = 0$.

In the proof of Theorem 4.27 the basic ideas of the proof of Theorem 4.15 presented in [Korte and Vygen, 2000] will be used.

Proof: (of Theorem 4.27)

The proof will be presented in three steps. First it is shown, that every incidence vector of a forest with exactly K trees in the graph G is a vertex of $P_{sub}^{=K}$. Then, every integral vertex of $P_{sub}^{=K}$ is proven to be the incidence vector of a forest with exactly K trees. Finally, the integrality of the vertices of $P_{sub}^{=K}$ is shown.

1. Let $F = (V, E_F)$ be a forest in $G = (V, E)$ consisting of exactly K trees and y^F the incidence vector of its edge set E_F. Let T_i, $i \in \{1, \ldots, K\}$ be the trees in F, $T_i = (V_i, E_i) \subseteq F$, $i \in \{1, \ldots, K\}$, $\bigcup_{i=1}^{K} T_i = F$. Thus,

$$\sum_{e \in E_F} y_e = \sum_{i=1}^{K} \sum_{e \in E_i} y_e = \sum_{i=1}^{K} (|V_i| - 1) = |V| - K, \tag{4.38}$$

   i.e., equation (4.31) is fulfilled.
   F is a forest, so it does not contain any circles and the subtour inequalities (4.32) hold and therefore $y^F \in P_{sub}^{=K}$.
   As $y^F \in \{0,1\}^{|E|}$, it is a vertex of $P_{sub}^{=K}$.

2. Let $y \in \{0,1\}^{|E|}$ be an integral vertex of $P_{sub}^{=K}$. Then, $y \in \{0,1\}^{|E|}$ is the incidence vector of an edge set of a subgraph of G containing exactly $|V| - K$ edges and no circuit. Thus, y is the incidence vector of a forest with exactly K trees.

3. Let $w : E \to \mathbb{R}$ be an arbitrary edge-weight function on G. Applying Kruskal's algorithm [Kruskal, 1956] on the weighted graph G provides an edge-weight-minimum spanning forest $F = (V, E_F) \subseteq G$ containing as many

trees as G has components. $K \in \mathbb{N}$ is the number of components of G. Let $E_F = \{f_1, \ldots, f_{|V|-K}\}$ be the edges chosen by Kruskal's algorithm, i.e., $w(f_1) \leq w(f_2) \leq \ldots \leq w(f_{|V|-K})$.
Let X_k be the connected component of $(V, \{f_1, \ldots, f_{|V|-K}\})$ containing f_k, $k \in \{1, \ldots, |V|-K\}$. As f_k is the edge chosen in the $k-th$ step of Kruskal's algorithm, the edges in X_k are the edges in G already chosen by the algorithm and connected with f_k by already chosen edges.
Let $y^* \in \{0,1\}^{|E|}$ be the incidence vector of E_F. Consider the linear program

$$\begin{aligned}
\min \quad & \sum_{e \in E} w(e) y_e && (4.39)\\
\text{s.t.} \quad & \sum_{e \in E} y_e = |V| - K, && (4.40)\\
& \sum_{e \in E(G[X])} y_e \leq |X| - 1, \quad \forall X : \emptyset \neq X \subset V, && (4.41)\\
& y_e \geq 0. && (4.42)
\end{aligned}$$

Using Lemma 4.29 it will be shown that y^* is an optimal solution to this LP and with Theorem 4.28 the integrality of $P_{sub}^{=K}$ can be concluded.
For an edge $e \in E$, connecting the vertices v_1 and v_2, $v_1, v_2 \in V$ the notation $e \subseteq V^*$, $V^* \subseteq V$ will be used if and only if $v_1, v_2 \in V^*$.
Now, by introducing dual variables z_X for each $X : \emptyset \neq X \subset V$ and z_K for the component constraint $\sum_{e \in E} y_e \geq |V| - K$, the dual problem can be formulated:

$$\begin{aligned}
\max \quad & (|V| - K) z_K - \sum_{X : \emptyset \neq X \subset V} (|X| - 1) z_X && (4.43)\\
\text{s.t.} \quad & z_K - \sum_{X : e \subseteq X \subset V} z_X \leq w(e), \quad \forall\, e \in E && (4.44)\\
& z_X \geq 0. && (4.45)
\end{aligned}$$

For $k = 1, \ldots, |V| - K - 1$, define $z^*_{X_k} \stackrel{def}{=} w(f_l) - w(f_k)$, where l is the first index greater than k for which $f_l \cap X_k \neq \emptyset$. If no such index exists, set $z^*_{X_k} \stackrel{def}{=} -w(f_k) + |w(f_{|V|-K})|$. Note that by this choice $z^*_{X_k} \geq 0$ is satisfied for every k. Furthermore define $z^*_K \stackrel{def}{=} -|w(f_{|V|-K})|$ and

$$z^*_X \stackrel{def}{=} \begin{cases} z^*_{X_k}, & \text{if } X = X_k,\ X_k \in \{X_1, \ldots, X_{|V|-K}\}, \\ 0, & \text{if } X \subset V,\ X \notin \{X_1, \ldots, X_{|V|-K}\}. \end{cases} \qquad (4.46)$$

The variable z_K can take negative values as it is not restricted in the dual problem.

Then, the equalities

$$z_K^* - \sum_{X: e \subseteq X \subset V} z_X^* = w(f_i), \quad \forall\, e \in E, \tag{4.47}$$

hold, where i is the smallest index satisfying $e \in X_i$. Moreover, $w(f_i) \leq w(e)$, as otherwise in Kruskal's algorithm e had been chosen instead of f_i. So, $z^* = (z_K^*, z_{X_1}^*, \ldots, z_{X_{2^{|V|}}}^*)$ is a feasible solution of the dual problem.
For every $y_e^* > 0$, i.e., for every e belonging to the spanning forest created by Kruskal's algorithm the equality

$$z_K^* - \sum_{X: e \subseteq X \subset V} z_X^* = w(e), \quad \forall\, e \in E, \tag{4.48}$$

is fulfilled and thus

$$y_e^* \left(w(e) - \left(z_K^* - \sum_{X: e \subseteq X \subset V} z_X^* \right) \right) = 0, \quad \forall\, e \in E. \tag{4.49}$$

The variables z_X^* take values greater than zero only if $F[X]$, i.e., the subgraph of F induced by X, is connected. Then, for X the corresponding subtour constraint is fulfilled with equality.
This implies, that

$$z_X^* \left((|X| - 1) - \sum_{e \in E(G[X])} y_e^* \right) = 0, \quad \forall X :\ \emptyset \neq X \subset V. \tag{4.50}$$

As

$$z_K^* \left((|V| - K) - \sum_{e \in E} y_e^* \right) = 0 \tag{4.51}$$

for arbitrary z_K^*, Lemma 4.29 can be applied, thus y^* is an optimal solution for the primal and z^* for the dual linear program. With Theorem 4.28 the integrality of $P_{sub}^{=K}$ can be concluded.

Theorem 4.30. ($P_{sub}^{\leq K}$ is integral)

The polytope $P_{sub}^{\leq K}$ is integral. Its vertices are exactly the incidence vectors of forests with at most K trees in the underlying graph G.

Proof: The forest polytope (4.21) is integral [Schrijver, 2003]. The polytope $P_{sub}^{\leq K}$ is an intersection of the halfspace defined by

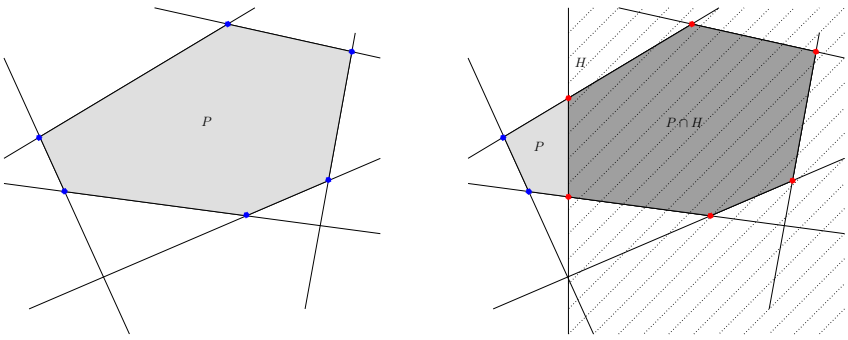
$$H = \left\{ y \in \mathbb{R}^{|E|} \mid \sum_{e \in E} y_e \geq |V| - K \right\} \tag{4.52}$$

and the forest polytope.
In the two special cases that this intersection is empty or consists of the whole polytope P_{forest}, the polytope $P_{sub}^{\leq K}$ is empty or is obviously integral as P_{forest} is integral.
In the general case, where $P_{forest} \cap H$ is a proper subset of P_{forest}, the hyperplane defined by

$$H^K = \left\{ y \in \mathbb{R}^{|E|} \mid \sum_{e \in E} y_e = |V| - K \right\} \tag{4.53}$$

is a supporting hyperplane of $P_{sub}^{\leq K}$.
Like illustrated in Figure 4.8, the vertices of $P_{sub}^{\leq K}$ are the vertices of P_{forest} contained in H united with the vertices of the face of $P_{sub}^{\leq K}$, that is defined by

$$\begin{aligned} F \stackrel{def}{=} \{ y^* \in [0,1]^{|E|} \subseteq \mathbb{R}^{|E|} \mid & \sum_{e \in E} y_e^* = |V| - K, \\ & \sum_{e \in E(G[X])} y_e^* \leq |X| - 1, \\ & \forall\, X :\ \emptyset \neq X \subset V \}. \end{aligned} \tag{4.54}$$



(a) Polytope (b) Polytope ∩ Halfspace

Figure 4.8: Vertices of a) polytope, b) polytope ∩ halfspace

This face is exactly the polytope $P_{sub}^{=K}$, of which is known, that it is integral 4.27. So, the vertices of $P_{sub}^{\leq K}$ are the union of a subset of the vertices of the two integral

polytopes P_{forest} and $P_{sub}^{=K}$ and therefore integral.
These integral vertices fulfil the subtour inequalities (4.30) as well as the inequality (4.29), assuring that the induced subgraph consists of at most K trees. Thus, the vertices are the incidence vectors of forests with at most K trees in the underlying graph G.
Since all such incidence vectors of forests with at most K trees in G fulfil (4.29) and (4.30) they are all contained in $P_{sub}^{\leq K}$ and as $y \in \{0,1\}^{|E|}$, they are vertices of $P_{sub}^{\leq K}$.

4.2.7 Cutset formulation

Now another description of a graph with at most K components will be derived, which will help to find an integer linear programming formulation for restricting the number of components in a graph and to design a separation algorithm (see section 4.2.9).

Theorem 4.31. (Cutset formulation)

An undirected graph $G = (V, E)$ consists of at most $K \in \mathbb{N}$ components, if and only if every $(K+1)$-cut is not empty.

Proof: The spanning forest of G consists of as many trees as G has components.
" $\Rightarrow$ ":
If G consists of exactly $\tilde{K} \leq K$ components, the spanning forest consists of $\tilde{K}$ trees, thus every $(\tilde{K}+1)$-cut contains at least one edge of the spanning forest. As $\tilde{K} \leq K$, every $(K+1)$-cut also contains at least one edge of the spanning forest.
" $\Leftarrow$ ":
If every $(K+1)$-cut is not empty, the graph can not be partitioned into $(K+1)$ or more unconnected components.

Definition 4.32.

The polytope corresponding to the cutset formulation of a graph with at most K components is defined by

$$P_{cut}^{\leq K} \stackrel{def}{=} \{ \quad y \in [0,1]^{|E|} \subseteq \mathbb{R}^{|E|} \mid \sum_{e \in \delta(V_1, V_2, \ldots, V_{K+1})} y_e \geq 1, \quad \forall\, \emptyset \neq V_1, V_2, \ldots, V_{K+1} \subset V \} \,. \tag{4.55}$$

Corollary 4.33.

The graph $G = (V, E)$ consists of at most K components, if and only if the number of edges in a minimum $(K+1)$-cut $\delta(G)$ is greater than zero, i.e.

$$|\delta(G)| > 0 \tag{4.56}$$

for a minimum $(K+1)$-cut $\delta(G)$.

Proof: $|\delta(G)| > 0$ for a minimum $(K+1)$-cut $\delta(G)$ holds if and only if there is at least one edge in every $(K+1)$-cut. (A cut $\delta(G)$ containing no edge satisfies $|\delta(G)| = 0$.)
Then, Theorem 4.31 can be applied.

Theorem 4.34.

An undirected graph $G = (V, E)$ consists of exactly $K \in \mathbb{N}$ components, if and only if there exists $y^* \in \{0,1\}^E$:

$$\sum_{e \in E} y_e^* = |V| - K, \tag{4.57}$$

$$\sum_{e \in \delta(V_1, V_2, \ldots, V_{K+1})} y_e^* \geq 1, \quad \forall\, \emptyset \neq V_1, V_2, \ldots, V_{K+1} \subset V. \tag{4.58}$$

Proof: From Theorem 4.31 it is known that the second condition is equivalent to G consisting of at most K components.
" $\Rightarrow$ ":
Assume, that G consists of exactly K components, then the incidence vector of every spanning forest of G fulfils (4.57) (see proof of Theorem 4.24) and obviously also (4.58).
" $\Leftarrow$ ":
Equation (4.57) ensures, that G does not consist of less than K components.

Definition 4.35.

The polyhedral description of the polytope corresponding to the cutset formulation of a graph with exactly K components is

$$P_{cut}^{=K} = \Big\{ y \in [0,1]^{|E|} \subseteq \mathbb{R}^{|E|} \mid \sum_{e \in E} y_e = |V| - K, \sum_{e \in \delta(V_1, V_2, \ldots, V_{K+1})} y_e \geq 1, \forall\, \emptyset \neq V_1, V_2, \ldots, V_{K+1} \subset V \Big\}. \tag{4.59}$$

Corollary 4.36.

The graph $G = (V, E)$ consists of exactly K components, if and only if there exists $y^* \in \{0,1\}^E$, being the incidence vector of the edges E^* of a subgraph $G^* = (V, E^*) \subseteq (V, E)$ of G, $E^* = \{e \in E \mid y_e^* = 1\}$ and satisfying

$$\sum_{e \in E} y_e^* = |V| - K, \tag{4.60}$$

$$\sum_{e \in \delta(G^*)} y_e^* \geq 1, \quad \text{for a minimum } (K+1)\text{-cut } \delta(G^*). \tag{4.61}$$

Proof: Condition (4.61) assures, that there is no empty $(K+1)$-cut with and Theorem 4.34 can be applied.

With the observations made above, the following theorem can be derived:

Theorem 4.37.

Let $G = (V, E)$ be an undirected graph. The following properties are equivalent:

1. *G is a forest with at most K components,*
2. *$|E| \geq |V| - K$ and there are no cycles in the graph,*
3. *There is no empty $(K+1)$-cut and there are no cycles in the graph.*

Proof: 1. (1) $\Leftrightarrow$ (2) follows directly from Theorem 4.24, as the inequalities (4.30) are exactly the subtour elimination inequalities, ensuring that there are no cycles in the graph. Every acyclic graph is a forest and vice versa.

2. The second equivalence (2) $\Leftrightarrow$ (3) can be derived from theorems 4.31 and 4.24.
   " $\Leftarrow$ " : If there is no empty $(K+1)$-cut, Theorem 4.31 shows that there are at most K components in the graph and thus, there have to be at least $|V| - K$ edges.
   " $\Rightarrow$ " : Theorem 4.24 shows, if (2) holds, G consists of at most K components. With (4.31) it can be concluded that the cutset of all $(K+1)$-cuts is not empty.

The subsequent theorem describes the relationship between the different polytopes introduced in this chapter.

Theorem 4.38.

The polytopes satisfy the relationship $P_{cut}^{\leq K} \supseteq P_{sub}^{\leq K} \supseteq P_{sub}^{=K}$ *and* $P_{cut}^{=K} \supseteq P_{sub}^{=K}$.

Proof: 1. Obviously, $P_{sub}^{\leq K} \supseteq P_{sub}^{=K}$.

2. To show $P_{cut}^{\leq K} \supseteq P_{sub}^{\leq K}$, it is assumed, that $y \in P_{sub}^{\leq K}$, but $y \notin P_{cut}^{\leq K}$, i.e., there exists a $(K+1)$-cut $\delta(V_1, \ldots, V_{K+1})$, that partitions the vertices V of a graph $G = (V, E)$ into $K+1$ disjoint subsets, $V = \bigcup_{i=1}^{K+1} V_i$ with $\sum_{e \in \delta(S)} y_e < 1$. Then

$$\begin{aligned}
|V| - K &\leq \sum_{e \in E} y_e \\
&= \sum_{e \notin \delta(S)} y_e + \sum_{e \in \delta(S)} y_e \\
&= \sum_{i=1}^{K+1} \sum_{e \subseteq V_i} y_e + \sum_{e \in \delta(S)} y_e \\
&< \sum_{i=1}^{K+1} \sum_{e \subseteq V_i} y_e + 1 \\
&\leq \sum_{i=1}^{K+1} (|V_i| - 1) + 1 \\
&= |V| - (K+1) + 1 \\
&= |V| - K.
\end{aligned}$$

This is a contradiction, so $P_{cut}^{\leq K} \supseteq P_{sub}^{\leq K}$.

3. Now it remains to show, that $P_{cut}^{=K} \supseteq P_{sub}^{=K}$.
With the hyperplane defined by

$$H^K \stackrel{def}{=} \left\{ y \in \mathbb{R}^{|E|} \mid \sum_{e \in E} y_e = |V| - K \right\}, \tag{4.62}$$

the regarded polytopes can be written in the following way:

$$P_{sub}^{=K} = P_{sub}^{\leq K} \cap H \text{ and } P_{cut}^{=K} = P_{cut}^{\leq K} \cap H. \tag{4.63}$$

$P_{sub}^{\leq K} \subseteq P_{cut}^{\leq K}$ (shown in (2)) implies, that $P_{sub}^{\leq K} \cap H \subseteq P_{cut}^{\leq K} \cap H$.

In general, $P_{sub}^{\leq K}$ is a proper subset of $P_{cut}^{\leq K}$. This is quite obvious, as $P_{cut}^{\leq K}$ can contain incidence vectors of graphs including subtours while $P_{sub}^{\leq K}$ only contains incidence vectors of forests.

4.2.8 Cutting plane algorithms

Cutting plane algorithms turned out to be very useful for solving integer linear programs, especially when combined with a branch and bound algorithm in a branch and cut framework, cf. section 4.2.10.
These methods work by solving iteratively a sequence of linear programming relaxations of the integer programming problem. The linear relaxations are non-integer linear programs. By improving those relaxations in every iteration, better approximations of the integer linear problem can be found. If an optimal solution of an relaxed problem turns out to be integer, a solution of the integer linear programming problem has been found. If it is not, there exists a linear inequality separating the current optimal solution of the relaxed problem from the convex hull of the feasible set of the integer linear problem. So, this new inequality is satisfied by the still unknown integer solution but not by the non-integer solution already attained. Such an inequality is a cut.
The cut is added to the relaxed linear program to cut off the current non-integer solution. By repeating this process iteratively, either an optimal integer solution or an approximation of the optimal integer solution is found [Gomory, 1958].
The problem of finding such a cut is the separation problem.
Cutting plane algorithms are also used for solving linear programming problems with exponentially many constraints. Here, first a subproblem of the original linear problem is solved, ignoring most of the constraints. Then, iteratively some of the firstly ignored constraints are added as cutting planes until a solution satisfying all constraints of the original problem is found. The separation problem to find these cutting planes is considered in the following section.

4.2.9 Separation

The restriction to a fixed maximum number of components in the regarded graph is a subproblem in the more general model to find a binary electron density distribution defined in chapter 2.
A quite similar situation has to be considered in the Spanning Tree Problem to find a spanning tree in a given graph. Here, given a graph $G = (V, E)$, a vector $y \in \{0, 1\}^{|E|}$ representing the edge weights and satisfying (4.18) and (4.19) has to

be found.
In both problems the number of constraints grows exponentially in the number of nodes of the underlying graph, so that an efficient way to solve these problems has to be found. For the Spanning Tree Problem usually a cutting plane procedure is used, which will be described subsequently. The problem of finding these cutting planes is called *separation problem* [Korte and Vygen, 2000].

Definition 4.39. (Separation Problem)

Given a polytope $P \subseteq \mathbb{R}^n$ and a vector $x^* \in \mathbb{R}^n$, either

- *decide that $x^* \in P$*
- *or find a vector $d \in \mathbb{R}^n$ and $\delta \in \mathbb{R}$, such that $d^T x \leq \delta$, $\forall x \in P$ and $d^T x^* > \delta$.*

The hyperplane $\{x \in \mathbb{R}^n \mid d^T x = \delta\}$ is called cutting plane.

Using a cutting plane procedure, the general Spanning Tree Problem is solved ignoring most of the subtour elimination constraints (4.30). So, for a given graph $G = (V, E)$

$$\sum_{e \in E(G[X])} y_e \leq |X| - 1, \quad \forall\ X :\ \emptyset \neq X \subseteq V, \tag{4.64}$$

$$y_e \in \{0, 1\}\ \forall e \in E, \tag{4.65}$$

only the constraints for some $X \subseteq V$ are considered first, not for all $X \subseteq V$.
If the solution of this reduced problem additionally satisfies all constraints that had been ignored, a solution for the whole problem has been found. Otherwise a violated constraint has to be found (separation problem) and is added to the reduced problem. So, the procedure works as follows:
Given a solution of the reduced problem $y^* \in [0, 1]^{|E|}$, try to find a set $X \subseteq V$, for which $\sum_{e \in E(G[X])} y_e^* \leq |X| - 1$ is not fulfilled. If there is no such set, $y^* \in [0, 1]^{|E|}$ is feasible for the whole problem.

Now, we consider the separation problem to find such a set X violating a subtour inequality.
A *network* $N = (V, A, s, t)$ is a directed graph with two outstanding nodes s and t, called the source and the sink of N and a capacity function $c : A \to \mathbb{R}^{\geq 0}$ on the

arcs, $a \mapsto c(a)$.
A *flow* in this network is a mapping $f : A \to \mathbb{R}^{\geq 0}$ satisfying

$$f(a) \leq c(a) \; \forall a \in A \tag{4.66}$$

$$\sum_{a \in \delta^-(v)} f(a) = \sum_{a \in \delta^+(v)} f(a) \; \forall v \in V \setminus \{s,t\}, \tag{4.67}$$

with $\delta^-(v) \stackrel{def}{=} \{a = (u,v) \in A \mid u \in V\}$ and $\delta^+(v) \stackrel{def}{=} \{a = (v,u) \in A \mid u \in V\}$.
The value of a flow θ is defined by $\theta = \sum_{a \in \delta^+(s)} f(a) - \sum_{a \in \delta^-(s)} f(a)$. The flow for which the value of flow is maximal amongst all other flows is called *maximum flow*.

Regard the capacitated network we get from the given graph $G = (V,E)$ by replacing every edge e, $e \in E$ by two arcs pointing in opposite directions and setting the capacity of each of both arcs to y_e^*. Then define a root node and calculate the maximum flow from the root node to every other node in this capacitated network. If a maximum flow has value $\theta < 1$, by the max-flow min-cut theorem, there exists a cut of capacity less than 1 and this cut directly provides a set $X_0 \subseteq V$ with $\sum_{e \in E(G[X_0])} y_e^* > |X_0| - 1$. The constraint $\sum_{e \in E(G[X_0])} y_e^* \leq |X_0| - 1$ then is added to the reduced problem and the procedure will be started again with this new reduced problem [Magnanti and Wolsey, 1995].

The number of constraints in the subtour elimination formulation (4.24) grows exponentially in the number of nodes in the underlying graph, so a cutting plane approach seems to be a suitable way to solve this problem. The advantage of the subtour elimination constraints in contrast to the cutset-constraint is, that the corresponding polytope is integral, therefore it is a stronger description than the cutset-formulation and the optimal solutions all can be found in the vertices of the polytope. The cutset-formulation can be related to a flow-problem by the max-flow min-cut theorem, which can be solved efficiently [Magnanti and Wolsey, 1995].

In the context of the phasing problem, the constraints restricting the maximum number of components in the regarded graph to a fixed value should be separated. Given a solution $y^* \in [0,1]^{|E|}$ of one of the phasing problem formulations presented in 2.4, consider the corresponding graph $G = (V,E)$ like described in 4.2.3 and add edge weights $y_e^* \; \forall e \in E$. Using Corollary 4.33, separation constraints can be found by evaluating the minimal $(K+1)$-cut $\delta(V_1, \ldots, V_{K+1})$, $V_1, \ldots, V_{K+1} \in V$. There exists a polynomial algorithm for evaluating the minimal K-cut in a graph with only positive edge weights for fixed K, which has been presented in [Goldschmidt

and Hochbaum, 1994].
If (4.56) is fulfilled, the actual solution is valid and the graph consist of at most K components. Otherwise, the cutset constraint

$$\sum_{e \in \delta(V_1,\ldots,V_{K+1})} y_e^* \geq 1, \tag{4.68}$$

for the actual minimal $(K+1)$-cut $\delta(V_1, \ldots, V_{K+1})$ can be added as a cutting plane. Using a Breadth-First Search algorithm, the number of components $\tilde{K}$ of a given graph and an assignment of the graph nodes to the components can be calculated. This yields a $\tilde{K}$-cut, such that the sum of the edge values in the cut is smaller than 1. If $\tilde{K} > K$ the constraint

$$\sum_{e \in \delta(V_1,\ldots,V_{\tilde{K}})} y_e^* \geq 1, \tag{4.69}$$

can be added, which is even stronger than (4.68) in case $\tilde{K} > K + 1$.
If the calculated solution $y^* \in [0,1]^{|E|}$ fulfills (4.29) and the sum of the edge values of a minimal $(K+1)$-cut $\delta(V_1, \ldots, V_{K+1})$ is equal to zero, this cut provides also violated subtour constraints. Assuming, the subtour constraints (4.30) were fulfilled, we get

$$|V| - K \overset{(4.29)}{\leq} \sum_{e \in E} y_e^* = \sum_{i=1}^{K+1} \sum_{e \in V_i} y_e^* \tag{4.70}$$

$$\overset{(4.30)}{\leq} \sum_{i=1}^{K+1} |V_i| - 1 = |V| - K - 1, \tag{4.71}$$

which is a contradiction. So, for at least one of the subsets $V_i,\ i \in \{1, \ldots, K+1\}$ the subtour constraint (4.30) is not fulfilled and the violated one can be added as a cutting plane.

In the cutting plane approach described here, the graph for which the cutting planes are generated, is assumed to be known. As in our approach to solve the phasing problem, the regarded graph depends on the binary electron density distribution, which is not known in advance, only local cuts can be generated using this approach. In section 5.1, a cutting plane procedure generating global cuts is described.

4.2.10 Branch and Bound

For solving integer linear programs efficiently, a branch and bound algorithm is useful as it preserves us from enumerating all possible solutions. We will now

describe the general branch and bound algorithm and refer to the special case of solving binary integer linear programs using branch and bound [Hillier and Lieberman, 2001; Land and Doig, 1960].
The branch and bound procedure can be split into two parts - a branching and a bounding step. We regard a linear programming problem with an objective function that should be minimized. The set of all possible solutions of the integer linear program is denoted by S.

In the branching step, S is split into smaller subsets S_1 to S_k with $\bigcup_{i=1}^{k} S_i = S$. The optimal solution value of the minimization problem over S is equal to the minimum value of the optimal solutions of the minimization problems solved over the smaller sets S_1 to S_k.
By recursively splitting these smaller sets into more subsets, a kind of tree structure is created, where the nodes are the subsets of S.
Regarding binary variables, the branching procedure is applied by solving the LP-relaxation of the considered integer linear program. This means, all variables are regarded as being real variables with a lower bound of 0 and an upper bound of 1 and the optimization problem is solved over these real variables. If the solution variables of the relaxed problem all take binary values, the integer linear program is solved. Otherwise, in a branching step, one variable not taking a binary value is chosen and the set of solutions is split into a set, where this variable takes the value 0 and another set, where this variable takes the value 1.
In the bounding step, upper and lower bounds for the optimal solutions over some subsets of S are calculated. If a lower bound of some subset S_i is greater than the upper bound of a subset S_j, S_i and its subsets can be erased from the set of possible optimal solutions. The algorithm stops, when the set of possible solutions contains only one single solution or the upper bound of the optimal solution values over S equals the lower bound.
The whole branch and bound procedure for a solving a binary linear program is illustrated in the flow chart on the next page.

Branch and cut methods are a combination of cutting plane algorithms with branch and bound algorithms. Starting with a cutting plane algorithm, relaxed linear programs are solved and the solution is improved iteratively. If no more cutting planes can be found or it becomes too expensive to search for them, a branch and bound procedure is started.

4.2.11 Graph partitioning problems

The 'K-component-constraint' is closely related to the partition problem in graph theory, which occurs in different variations and has already been investigated in

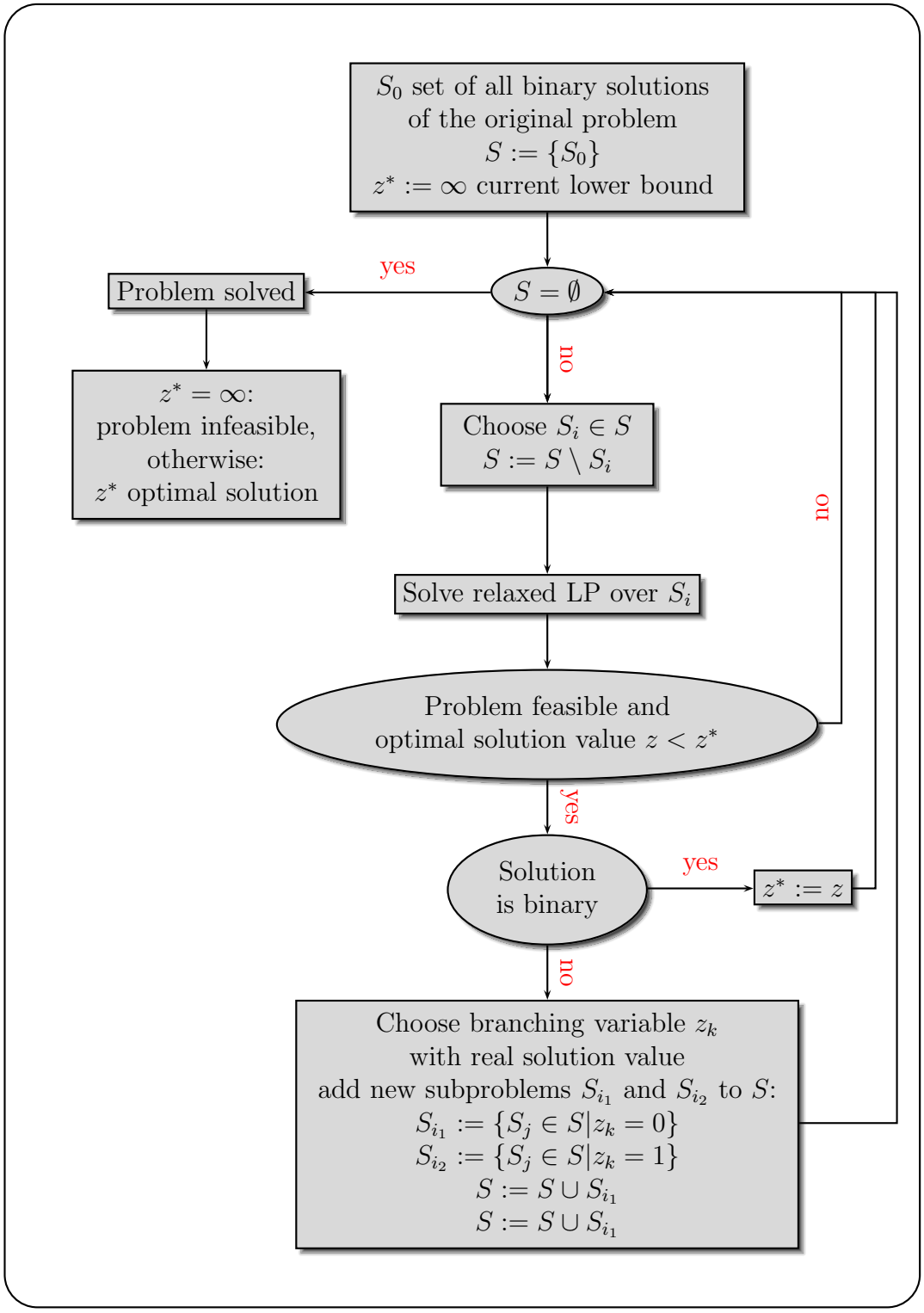


Figure 4.9: Branch and Bound algorithm

different publications. In some of them, inequalities are described, that could be used in branch and cut procedures for solving the phasing problem with the K-component-constraint.
The problem to partition the node set of a given graph into $K \in \mathbb{N}$ subsets, such that the number of edges with end points in two different subsets is minimized, is known to be NP-hard in general [Garey and Johnson, 1979].
For $K = 2$ this problem has been studied e.g. in [Barahona and Mahjoub, 1986].
In [Goldschmidt and Hochbaum, 1994] it is shown that it is possible to find in polynomial time a set of edges, that by its removal partitions the graph into exactly $K \in \mathbb{N}$ components provided K is fixed and, in case the graph edges are weighted, all edge weights are non-negative.
In [Sørensen, 2007] as well as [Ferreira et al., 1998], the problem to partition a graph with edge weights as well as node weights into at most $K \in \mathbb{N}$ subsets, while minimizing the weights of the edges between the subsets and restricting the node weights in each subset, is regarded.
The problem to partition the node set of a given graph into at most $K \in \mathbb{N}$ subsets, such that the number of edges with end points in two different subsets is minimized has been considered in [Chopra and Rao, 1993]. The approach presented there needs variables for edge weights as well as for node weights.
In [Chopra and Rao, 1995, 1993; Deza et al., 1992] several classes of valid inequalities for the polytopes that are the convex hulls of the incidence vectors of all multicuts partitioning the complete graph with $N \in \mathbb{N}$ nodes K_n into at least or at most $K \in \mathbb{N}$ subsets, are presented. If the regarded graph $G = (V, E)$ is not complete, [Chopra and Rao, 1993] suggest to add all missing edges $\overline{E}$ to this graph, such that $E \cup \overline{E} = E(K_n)$ and assign edge weights $w_e = 0,\ \forall e \in \overline{E}$. Then the 'K-component-constraint' can be formulated using only edge variables.
Some of those inequality-classes are also profed to be facet-defining and polynomial-time separable, such as the cycle inequalities for chordless cycles [Chopra and Rao, 1995]. Here, binary edge variables $x_e,\ e \in E(K_n)$ are introduced, taking the value 1 if an edge is cut by the partition and 0 otherwise. Then the cycle inequalities tell, that if all nodes belonging to a cycle $C = (V(C), E(C))$ in the graph $G = (V, E)$ are in the same subset of a partition, then none of the edges of the cycle $E(C)$ can be in the cut:

$$x_{\overline{e}} \leq \sum_{e \in E(C) \setminus \{\overline{e}\}} x_e, \tag{4.72}$$

for every cycle C of G and all edges $\overline{e} \in C$.
Other classes of inequalities that are facet-defining and polynomial-time separable are the q-wheel inequalities, if q is odd and $K \geq 4$ as well as the q-bicycle-wheel inequalities for odd q and $K \geq 3$ [Chopra and Rao, 1993; Deza et al., 1992].
In further work, adding such inequalities could possibly decrease the running time

needed for the solution of the phasing problem.

Chapter 5

Modelling the Phase Problem

All models are wrong but some are useful.

George E.P.Box

Based on the previous chapter a modelling approach for the phase problem taking into account the mentioned supplementary geometric constraints will be presented here.
In section 2.4 the mixed integer program for the calculation of the phases has been presented, without taking any topological properties of the regarded protein into account. For this approach the following variables are necessary.
The electron density has to be calculated on a grid, so for every grid point $\mathbf{j} \in \Pi$, $\Pi \stackrel{def}{=} [0, M_1 - 1] \times [0, M_2 - 1] \times [0, M_3 - 1] \subseteq \mathbb{Z}^3$ a binary variable $z_{\mathbf{j}} \in \{0, 1\}$ representing the binary grid electron density in this point is introduced. For the representation of the phase values, two binary variables $y^R_{\mathbf{h}}$ and $y^I_{\mathbf{h}}$, $\mathbf{h} \in \Pi$ are needed.
Additionally for every $\mathbf{h} \in \Pi$ two penalty variables $r^R_{\mathbf{h}}, r^I_{\mathbf{h}} \in \mathbb{R}^{\geq 0}$ are introduced. These are the only variables occurring in the objective function (2.92).
Now, the topological properties of proteins discussed in chapter 4 will be incorporated and added to the optimization problem formulated in (2.92) to (2.96). So, we are searching for an integer linear programming formulation for the following

constraint program. The constants $\mathfrak{p} \in \mathbb{R}$ and $K \in \mathbb{N}$ are supposed to be known.

$$\begin{aligned}
\min \quad & \sum_{\mathbf{h}\in\Pi}(r_{\mathbf{h}}^R + r_{\mathbf{h}}^I) && (5.1)\\
\text{subject to} \quad & -\varepsilon_{\mathbf{h}} - r_{\mathbf{h}}^R \leq A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \leq \varepsilon_{\mathbf{h}} + r_{\mathbf{h}}^R, \quad \forall \mathbf{h}\in\Pi, && (5.2)\\
& -\varepsilon_{\mathbf{h}} - r_{\mathbf{h}}^I \leq A^I(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \leq \varepsilon_{\mathbf{h}} + r_{\mathbf{h}}^I, \quad \forall \mathbf{h}\in\Pi, && (5.3)\\
& z_{\mathbf{j}},\ y_{\mathbf{h}}^R,\ y_{\mathbf{h}}^I \in \{0,1\},\ r_{\mathbf{h}}^R \leq 0,\ r_{\mathbf{h}}^I \leq 0, \quad \forall \mathbf{j},\ \mathbf{h}\in\Pi, && (5.4)\\
& z_{\mathbf{j}} = 1 \text{ for at least } \mathfrak{p}\% \text{ of the } |\Pi| \text{ variables}, && (5.5)\\
& \text{the binary picture represented by the solution} && \\
& \bullet \text{ should not contain any isolated points}, && (5.6)\\
& \bullet \text{ contains at most } K \text{ components}. && (5.7)
\end{aligned}$$

Using the linear programming formulation for the exclusion of isolated points (4.8) as well as the specification of a minimum covering (4.9), without adding any additional variables, we get the following problem description:

$$\begin{aligned}
\min \quad & \sum_{\mathbf{h}\in\Pi}(r_{\mathbf{h}}^R + r_{\mathbf{h}}^I) && (5.8)\\
\text{subject to} \quad & -\varepsilon_{\mathbf{h}} - r_{\mathbf{h}}^R \leq A^R(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \leq \varepsilon_{\mathbf{h}} + r_{\mathbf{h}}^R, \quad \forall \mathbf{h}\in\Pi, && (5.9)\\
& -\varepsilon_{\mathbf{h}} - r_{\mathbf{h}}^I \leq A^I(\mathbf{h},\mathbf{z},\mathbf{y}(\mathbf{h})) \leq \varepsilon_{\mathbf{h}} + r_{\mathbf{h}}^I, \quad \forall \mathbf{h}\in\Pi, && (5.10)\\
& \frac{p}{100}\cdot M_1M_2M_3 \leq \sum_{\mathbf{h}\in\Pi} z_{\mathbf{j}}, && (5.11)\\
& -5 \leq z_{\mathbf{j}} - \sum_{\tilde{\mathbf{j}}\,\mathfrak{n}\,\mathbf{h}} z_{\tilde{\mathbf{j}}} \leq 0, \quad \forall \mathbf{j},\tilde{\mathbf{j}}\in\Pi, && (5.12)\\
& z_{\mathbf{j}},\ y_{\mathbf{h}}^R,\ y_{\mathbf{h}}^I \in \{0,1\},\ r_{\mathbf{h}}^R \leq 0,\ r_{\mathbf{h}}^I \leq 0, \quad \forall \mathbf{j},\mathbf{h}\in\Pi, && (5.13)\\
& \bullet \text{ the binary picture represented by the solution} && \\
& \quad \text{contains at most } K \text{ components}. && (5.14)
\end{aligned}$$

For the connectivity constraint introduced in section 4.2, supplementary variables $e_{\mathbf{j_1},\mathbf{j_2}} \in \{0,1\}$, $\mathbf{j_1}\mathbf{j_2} \in \Pi$ are introduced for the edges of a graph with the nodes $\mathbf{j} \in \Pi$ and an edge between every two neighboured points in Π.
These variables should be 1, if the corresponding edge connects two neighbouring nodes $\mathbf{j_1} \in \Pi$ and $\mathbf{j_2} \in \Pi$ with $z_{\mathbf{j_1}} = z_{\mathbf{j_2}} = 1$ and 0 otherwise:

$$e_{\mathbf{j_1}\mathbf{j_2}} = \begin{cases} 1, & \text{if } z_{\mathbf{j_1}} = z_{\mathbf{j_2}} = 1 \text{ and } \mathbf{j_1}\ \mathfrak{n}\ \mathbf{j_2} \\ 0, & \text{otherwise.} \end{cases} \qquad (5.15)$$

The following constraints (5.16), (5.17) and (5.18) ensure, that the edge variables satisfy (5.15).
For all $\mathbf{j_1}, \mathbf{j_2} \in \Pi$ with $\mathbf{j_1} \not\equiv \mathbf{j_2}$

$$e_{\mathbf{j_1 j_2}} \leq z_{\mathbf{j_1}}, \tag{5.16}$$
$$e_{\mathbf{j_1 j_2}} \leq z_{\mathbf{j_2}}, \tag{5.17}$$
$$z_{\mathbf{j_1}} + z_{\mathbf{j_2}} - 1 \leq e_{\mathbf{j_1 j_2}} \tag{5.18}$$
$$z_{\mathbf{j_1}},\ z_{\mathbf{j_2}},\ e_{\mathbf{j_1 j_2}} \in \{0,1\}. \tag{5.19}$$

In chapter 4, different ways have been introduced to model a constraint for the maximum number of components $K \in \mathbb{N}$ in a graph, supposing that this graph is already known. But, in the crystallographic context, the graph is defined by the values of the binary grid electron density variables $z_{\mathbf{j}}$ for all $\mathbf{j} \in \Pi$ and therefore not known in advance. So, these connectivity models can only be used after a binary electron density distribution has been found and thus, these constraints are only valid locally, i.e., for one special electron density distribution. If this distribution also satisfies the component-constraint, it will be accepted as solution, if not, a no-good-cut can be used to get rid of this solution.
If a solution $z^* \in \{0,1\}^{|\Pi|}$ with the corresponding edge values $e_{\mathbf{j_1 j_2}}$, $\forall \mathbf{j_1}, \mathbf{j_2} \in \Pi$ with $\mathbf{j_1} \not\equiv \mathbf{j_2}$ of the optimization problem described above has been found, we can test if this solution also fulfils the connectivity constraint. This is the case, if the solution satisfies either of the following constraint systems (5.20) to (5.22) or (5.24).
Using the subtour elimination constraints from Theorem 4.24, we get for all $\mathbf{j} \in \Pi, \forall \mathbf{j_1}, \mathbf{j_2} \in \Pi$ with $\mathbf{j_1} \not\equiv \mathbf{j_2}$:

$$\sum_{\mathbf{j_1}, \mathbf{j_2} \in \Pi} e_{\mathbf{j_1 j_2}} \geq |\Pi| - K, \tag{5.20}$$
$$\sum_{\substack{\mathbf{j_1}, \mathbf{j_2} \in \Pi \\ \mathbf{j_1} \not\equiv \mathbf{j_2}}} e_{\mathbf{j_1 j_2}} \leq |T| - 1, \quad \forall\ T:\ \emptyset \neq T \subseteq \Pi \tag{5.21}$$
$$z_{\mathbf{j}},\ e_{\mathbf{j_1 j_2}} \in \{0,1\}, \tag{5.22}$$
$$\tag{5.23}$$

or, using the cutset formulation from Corollary 4.33

$$|\delta_{min}(G^*_\Pi)| > 0,\ \text{for a minimum } (K+1)\text{-cut } \delta_{min}(G^*_\Pi), \tag{5.24}$$

with $G^*_\Pi = (V^*_\Pi, E^*_\Pi)$, $V^*_\Pi = \{\mathbf{j} \in \Pi \mid\ z_{\mathbf{j}} = 1\}$ and $E^*_\Pi = \{e_{\mathbf{j_1 j_2}}, |\ e_{\mathbf{j_1 j_2}} = 1, \forall\ \mathbf{j_1}, \mathbf{j_2} \in \Pi \text{ with } \mathbf{j_1} \not\equiv \mathbf{j_2}\}$.

In order to find global constraints that can be used without knowing the values of the grid electron density variables, a new approach has been proposed.

We introduce three new binary variables $u_T, v_T, p_T \in \{0,1\}$ for every subset of the set of grid points $T \subset \Pi$. There are $2^{|\Pi|-1}-1$ such subsets.
The variable u_T indicates, if the considered subset T contains grid points $\mathbf{j} \in \Pi$ where the variable $z_{\mathbf{j}}$ takes the value 1, v_T indicates if $\Pi \setminus T$ contains such grid points, the variable p_{T_i} indicates if both, T and $\Pi \setminus T$ contain such grid points.

$$u_{\mathbf{T}} = \begin{cases} 1, & \text{if } \sum\limits_{\mathbf{j} \in T} z_{\mathbf{j}} \geq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.25}$$

$$v_{\mathbf{T}} = \begin{cases} 1, & \text{if } \sum\limits_{\mathbf{j} \in \Pi \setminus T} z_{\mathbf{j}} \geq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.26}$$

$$p_{\mathbf{T}} = \begin{cases} 1, & \text{if } u_{\mathbf{T}} = v_{\mathbf{T}} = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.27}$$

If $p_{\mathbf{T}} = 1$, both subsets T and $\Pi \setminus T$ partitioning Π contain grid points with a binary grid electron density value of 1 and therefore it has to be ensured that in case $K = 1$ these subsets are connected by at least one edge.

This approach is presented for the case that the graph representing the binary grid electron density distribution consists of only one component. Afterwards it is extended for ensuring that the graph representing the binary grid electron density distribution consists of at most $K \in \mathbb{N}$ components.

Theorem 5.1.

Any binary grid electron density distribution $z^* \in \{0,1\}^{M_1 \times M_2 \times M_3}$ satisfying the following constraints contains only one component. Additionally, no binary solution $z^* \in \{0,1\}^{M_1 \times M_2 \times M_3}$ containing only one component is cut off by these constraints.

$$\forall\ \emptyset \neq T \subset \Pi,\ \forall \mathbf{j} \in \Pi, \mathbf{j_1}, \mathbf{j_2} \in \Pi \text{ with } \mathbf{j_1} \ \mathfrak{n}\ \mathbf{j_2}:$$

$$
\begin{aligned}
e_{\mathbf{j_1 j_2}} &\leq z_{\mathbf{j_1}} && (5.28)\\
e_{\mathbf{j_1 j_2}} &\leq z_{\mathbf{j_2}} && (5.29)\\
z_{\mathbf{j_1}} + z_{\mathbf{j_2}} - 1 &\leq e_{\mathbf{j_1 j_2}} && (5.30)\\
\frac{1}{|T|}\sum_{\mathbf{j}\in T} z_{\mathbf{j}} &\leq u_T \leq \sum_{\mathbf{j}\in T} z_{\mathbf{j}} && (5.31)\\
\frac{1}{|\Pi \setminus T|}\sum_{\mathbf{j}\in \Pi\setminus T} z_{\mathbf{j}} &\leq v_T \leq \sum_{\mathbf{j}\in \Pi\setminus T} z_{\mathbf{j}} && (5.32)\\
p_T &\leq u_T && (5.33)\\
p_T &\leq v_T && (5.34)\\
u_T + v_T - 1 &\leq p_T && (5.35)\\
p_T &\leq \sum_{\mathbf{j_1}\in T, \mathbf{j_2}\in \Pi\setminus T} e_{\mathbf{j_1 j_2}} && (5.36)\\
u_T, v_T, p_T, z_{\mathbf{j}}, e_{\mathbf{j_1 j_2}} &\in \{0,1\}. && (5.37)
\end{aligned}
$$

Proof: 1. In a first step it is shown that an arbitrary binary picture $z \in \{0,1\}^{M_1\times M_2\times M_3}$ satisfying the constraints above contains exactly one component.
Assume that a given solution $z^* \in \{0,1\}^{M_1\times M_2\times M_3}$ satisfies all constraints, but consists of more than one component. Consider a graph $G_\Pi = (V_\Pi, E_\Pi)$, every grid point $\mathbf{j} \in \Pi$ is identified with a node $\mathbf{j} \in V$ with a corresponding variable $z^*_{\mathbf{j}} \in \{0,1\},\ \forall \mathbf{j} \in \Pi$ and there are edges $(\mathbf{j_1}, \mathbf{j_2}) \subseteq E_\Pi$ between all neighbouring grid points $\mathbf{j_1}, \mathbf{j_2} \in \Pi$. There is a variable $e_{\mathbf{j_1 j_2}} \in \{0,1\}$ for each edge $(\mathbf{j_1}, \mathbf{j_2})$.
Let $G^*_\Pi = (V^*_\Pi, E^*_\Pi)$ be the subgraph of G_Π, defined by $V^*_\Pi = \{\mathbf{j} \in V \mid z^*_{\mathbf{j}} = 1\} \subseteq V_\Pi$ and $E^*_\Pi = \{(\mathbf{j_1}, \mathbf{j_2}) \in E_\Pi \mid e_{\mathbf{j_1 j_2}} = 1\} \subseteq E_\Pi$.
If z^* consists of more than one component, G^*_Π is not connected. Thus, there exists a cut $\tilde{\delta}(T) = \emptyset$ in G^*_Π partitioning G^*_Π into 2 disjoint sets T and $G^*_\Pi \setminus T$, each containing at least one vertex $\mathbf{j}$ with $z^*_{\mathbf{j}} > 0$.
Thus, the cut $\delta(T)$ in G_Π partitioning G_Π into the 2 disjoint sets T and $\Pi\setminus T$, contains only edges with corresponding variable-values of zero:

$$\sum_{(\mathbf{j_1},\mathbf{j_2})\in\delta(T)} e_{\mathbf{j_1 j_2}} = 0,\ \mathbf{j_1}, \mathbf{j_2} \in \Pi. \tag{5.38}$$

Therefore,

$$0 = \sum_{(\mathbf{j_1},\mathbf{j_2})\in\delta(T)} e_{\mathbf{j_1 j_2}} \overset{(5.36)}{\geq} p_T \overset{(5.37)}{\geq} 0 \quad \Rightarrow \quad p_T = 0. \tag{5.39}$$

Inequality (5.35) then shows

$$0 = p_T \geq u_T + v_T - 1 \quad \Rightarrow \quad u_T + v_T \leq 1 \tag{5.40}$$

$$\Rightarrow \quad u_T = 0 \vee v_T = 0 \tag{5.41}$$

and with (5.31) respectively (5.32) one gets

$$\sum_{i \in T} z_i = 0 \text{ or } \sum_{i \in \Pi \setminus T} z_i = 0, \tag{5.42}$$

each contradicting the assumption of T and $\Pi \setminus T$ each containing at least one vertex $\mathbf{j}$ with $z^*_{\mathbf{j}} > 0$.

2. Now, in a second step it will be proven that no integer solution will be cut off by the constraints above.
For an arbitrary integer solution $z^* \in \{0,1\}^{M_1 \times M_2 \times M_3}$ containing only one component, the variables $e_{\mathbf{j_1 j_2}} \in \{0,1\}$, $\forall \mathbf{j_1}, \mathbf{j_2} \in \Pi$ with $\mathbf{j_1} \ \mathfrak{n} \ \mathbf{j_2}$, as well as $u_T, v_T, p_T \in \{0,1\}$ are chosen in the following way:

$$e_{\mathbf{j_1 j_2}} = \begin{cases} 1, & \text{if } z^*_{\mathbf{j_1}} = z^*_{\mathbf{j_2}} = 1 \text{ and } \mathbf{j_1} \ \mathfrak{n} \ \mathbf{j_2} \\ 0, & \text{otherwise.} \end{cases} \tag{5.43}$$

$$u_T = \begin{cases} 1, & \text{if } \sum_{\mathbf{j} \in T} z^*_{\mathbf{j}} \geq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.44}$$

$$v_T = \begin{cases} 1, & \text{if } \sum_{\mathbf{j} \in \Pi \setminus T} z^*_{\mathbf{j}} \geq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.45}$$

$$p_T = \begin{cases} 1, & \text{if } u_T = v_T = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.46}$$

Then obviously the constraints (5.28) to (5.35) are satisfied for arbitrary partitions $T, \Pi \setminus T \subseteq \Pi$.

Now, the remaining constraint (5.36) has to be considered. Assuming, for the integer solution z^* there exists a partition not satisfying (5.36), i.e.,

$$p_T > \sum_{\mathbf{j_1} \in T, \mathbf{j_2} \in \Pi \setminus T} e_{\mathbf{j_1 j_2}}, \tag{5.47}$$

then $p_T = 0$ can directly be excluded as it implies

$$\sum_{\mathbf{j_1} \in T, \mathbf{j_2} \in \Pi \setminus T} e_{\mathbf{j_1 j_2}} < 0, \tag{5.48}$$

which is a contradiction to

$$e_{\mathbf{j_1 j_2}} \in \{0,1\},\ \forall \mathbf{j_1}, \mathbf{j_2} \in \Pi. \tag{5.49}$$

So,

$$p_T > \sum_{\mathbf{j_1} \in T, \mathbf{j_2} \in \Pi \setminus T} e_{\mathbf{j_1 j_2}} \tag{5.50}$$

$$\Rightarrow \quad p_T = 1 \wedge \sum_{\mathbf{j_1} \in T, \mathbf{j_2} \in \Pi \setminus T} e_{\mathbf{j_1 j_2}} = 0. \tag{5.51}$$

Using the inequalities (5.35) as well as (5.31) and (5.32) it can be deduced that

$$p_T = 1 \overset{(5.46)}{\Longrightarrow} u_T = v_T = 1 \tag{5.52}$$

$$\overset{(5.31),(5.32)}{\Longrightarrow} \sum_{\mathbf{j} \in T} z_{\mathbf{j}} \geq 1 \wedge \sum_{\mathbf{j} \in \Pi \setminus T} z_{\mathbf{j}} \geq 1 \tag{5.53}$$

$$\Rightarrow \exists \tilde{\mathbf{j}}_{\mathbf{1}} \in T : z^*_{\tilde{\mathbf{j}}_{\mathbf{1}}} = 1 \wedge \exists \tilde{\mathbf{j}}_{\mathbf{2}} \in \Pi \setminus T : z^*_{\tilde{\mathbf{j}}_{\mathbf{2}}} = 1. \tag{5.54}$$

From $\sum_{\mathbf{j_1} \in T, \mathbf{j_2} \in \Pi \setminus T} e_{\mathbf{j_1 j_2}} = 0$ (5.51) it can be concluded that $e_{\mathbf{j_1 j_2}} = 0,\ \forall \mathbf{j_1} \in T, \mathbf{j_2} \in \Pi \setminus T$ and thus $e_{\tilde{\mathbf{j}}_{\mathbf{1}} \tilde{\mathbf{j}}_{\mathbf{2}}} = 0$.
Inserting these values in (5.30) then directly leads to a contradiction:

$$1 = z^*_{\tilde{\mathbf{j}}_{\mathbf{1}}} + z^*_{\tilde{\mathbf{j}}_{\mathbf{2}}} - 1 \leq e_{\tilde{\mathbf{j}}_{\mathbf{1}} \tilde{\mathbf{j}}_{\mathbf{2}}} = 0. \tag{5.55}$$

Thus, for an arbitrary integer solution $z^*_{\mathbf{j}}, \mathbf{j} \in \Pi$ and for every partition of Π into T and $\Pi \setminus T$, the values of the remaining variables can always be chosen in a way, that all constraints are satisfied.

The number of new variables introduced in the approach above can be reduced to just two new variables for each subset of Π. For every $T \subset \Pi$, there is a $\tilde{T} \subset \Pi$ with $T = \Pi \setminus \tilde{T}$ and thus $u_T = v_{\tilde{T}}$. So, we can replace $v_{\tilde{T}}$ by u_T for every $\tilde{T} \subset \Pi$ with $T = \Pi \setminus \tilde{T}$.
Now, this approach will be extended for a binary picture, containing at most $K \in \mathbb{N}$ components. The proof for this extension is quite similar to the one of Theorem 5.1. We introduce just one new variable u_T for each subset T of Π, indicating if T contains grid points $\mathbf{j} \in \Pi$ satisfying $z_{\mathbf{j}} = 1$.

Theorem 5.2.

Any binary grid electron density distribution $z^* \in \{0,1\}^{M_1 \times M_2 \times M_3}$ *satisfying the following constraints contains at most* $K \in \mathbb{N}$ *components. Additionally no binary solution* $z^* \in \{0,1\}^{M_1 \times M_2 \times M_3}$ *consisting of at most* $K \in \mathbb{N}$ *components is cut off by these constraints.*

$$\forall \emptyset \neq T_1, \ldots, T_{K+1} \subset \Pi, \bigcup_{i=1}^{K+1} T_i = \Pi,\ T_i \cap T_j = \emptyset,\ \forall i \neq j,\ i,j \in \{1, \ldots K+1\},$$

$\forall \mathbf{j}, \mathbf{j_1}, \mathbf{j_2} \in \Pi$ *with* $\mathbf{j_1}\ \mathfrak{n}\ \mathbf{j_2}$:

$$e_{\mathbf{j_1 j_2}} \leq z_{\mathbf{j_1}} \tag{5.56}$$

$$e_{\mathbf{j_1 j_2}} \leq z_{\mathbf{j_2}} \tag{5.57}$$

$$z_{\mathbf{j_1}} + z_{\mathbf{j_2}} - 1 \leq e_{\mathbf{j_1 j_2}} \tag{5.58}$$

$$\frac{1}{|T_i|} \sum_{\mathbf{j} \in T_i} z_{\mathbf{j}} \leq u_{T_i} \leq \sum_{\mathbf{j} \in T_i} z_{\mathbf{j}}, \quad \forall\ i \in \{1, \ldots K+1\} \tag{5.59}$$

$$\sum_{i=1}^{K+1} u_{T_i} - K \leq \sum_{(\mathbf{j_1}, \mathbf{j_2}) \in \delta(T_1, \ldots, T_{K+1})} e_{\mathbf{j_1 j_2}} \tag{5.60}$$

$$z_{\mathbf{j}}, e_{\mathbf{j_1 j_2}}, u_{T_i} \in \{0,1\}, \quad \forall\ i \in \{1, \ldots K+1\}. \tag{5.61}$$

Proof: 1. The proof is quite analogous to the one of the previous theorem. In a first step it is shown that an arbitrary binary picture $z \in \{0,1\}^{M_1 \times M_2 \times M_3}$ satisfying the constraints above contains at most $K \in \mathbb{N}$ components.
Assume that a given solution $z^* \in \{0,1\}^{M_1 \times M_2 \times M_3}$ satisfies all constraints, but consists of more than K components. Consider a graph $G_\Pi = (V_\Pi, E_\Pi)$, every grid point $\mathbf{j} \in \Pi$ is identified with a node $\mathbf{j} \in V$ with a corresponding variable $z^*_{\mathbf{j}} \in \{0,1\},\ \forall \mathbf{j} \in \Pi$ and there are edges $(\mathbf{j_1}, \mathbf{j_2}) \in E_\Pi$ between all neighbouring grid points $\mathbf{j_1}, \mathbf{j_2} \in \Pi$. There are variables $e_{\mathbf{j_1 j_2}} \in \{0,1\}$ for each edge $(\mathbf{j_1}, \mathbf{j_2})$.
Let $G^*_\Pi = (V^*_\Pi, E^*_\Pi)$ be the subgraph of G_Π, defined by $V^*_\Pi = \{\mathbf{j} \in V \mid z^*_{\mathbf{j}} = 1\} \subseteq V_\Pi$ and $E^*_\Pi = \{(\mathbf{j_1}, \mathbf{j_2}) \in E_\Pi \mid e_{\mathbf{j_1 j_2}} = 1\} \subseteq E_\Pi$.
As z^* contains more than K components, a $(K+1)$-cut $\tilde{\delta}(\tilde{T}_1, \ldots, \tilde{T}_{K+1})$ exists, partitioning G^*_Π into $K+1$ disjoint sets $\tilde{T}_1, \ldots, \tilde{T}_{K+1}$, of which no two of them are connected and each contains at least one vertex $\mathbf{j}$ with $z^*_{\mathbf{j}} > 0$. Thus, in the graph G_Π a cut $\delta(T_1, \ldots, T_{K+1})$ can be found, satisfying $\tilde{T}_1 \subseteq T_1, \ldots, \tilde{T}_{K+1} \subseteq T_{K+1}$ and

$$\sum_{(\mathbf{j_1}, \mathbf{j_2}) \in \delta(T_1, \ldots, T_{K+1})} e_{\mathbf{j_1 j_2}} = 0. \tag{5.62}$$

Thus,

$$0 = \sum_{(\mathbf{j_1},\mathbf{j_2})\in\delta(T_1,\ldots,T_{K+1})} e_{\mathbf{j_1}\mathbf{j_2}} \overset{(5.60)}{\geq} \sum_{i=1}^{K+1} u_{T_i} - K \tag{5.63}$$

$$\Rightarrow \quad \sum_{i=1}^{K+1} u_{T_i} \leq K \tag{5.64}$$

$$\Rightarrow \quad \exists i \in \{1,\ldots,K+1\} : u_{T_i} = 0 \tag{5.65}$$

$$\overset{(5.59)}{\Rightarrow} \quad \exists i \in \{1,\ldots,K+1\} : \sum_{\mathbf{j}\in T_i} z^*_{\mathbf{j}} = 0, \tag{5.66}$$

contradicting the assumption of $T_1,\ldots,T_{K+1}$ containing at least one vertex $\mathbf{j}$ with $z^*_{\mathbf{j}} > 0$.

2. Now, in a second step it will be proven, that no integer solution consisting of at most $K \in \mathbb{N}$ components will be cut off by the constraints above.
It can easily be seen that for an arbitrary integer solution $z^* \in \{0,1\}^{M_1\times M_2\times M_3}$ consisting of at most $K \in \mathbb{N}$ components, the variables $e_{\mathbf{j_1}\mathbf{j_2}} \in \{0,1\}$, for all $\mathbf{j_1},\mathbf{j_2} \in \Pi$ with $\mathbf{j_1} \,\mathfrak{n}\, \mathbf{j_2}$, as well as $u_T \in \{0,1\}$ for all subsets $T \subset \Pi$ can be chosen in a way, that the constraints (5.56) to (5.61) are satisfied for arbitrary partitions of Π into nonempty subsets $T_1,\ldots,T_{K+1} \subseteq \Pi$, $\bigcup_{i=1}^{K+1} T_i = \Pi$, $T_i \cap T_j = \emptyset$, $\forall i \neq j,\ i,j \in \{1,\ldots K+1\}$:

$$e_{\mathbf{j_1}\mathbf{j_2}} = \begin{cases} 1, & \text{if } z^*_{\mathbf{j_1}} = z^*_{\mathbf{j_2}} = 1 \text{ and } \mathbf{j_1} \,\mathfrak{n}\, \mathbf{j_2} \\ 0, & \text{otherwise,} \end{cases} \tag{5.67}$$

$$u_{T_i} = \begin{cases} 1, & \text{if } \sum\limits_{\mathbf{j}\in T_i} z^*_{\mathbf{j}} \geq 1 \\ 0, & \text{otherwise.} \end{cases} \qquad \forall\, i \in \{1,\ldots K+1\} \tag{5.68}$$

So, only the remaining constraint (5.60) has to be regarded. Assuming, for the integer solution z^* there exists a partition not satisfying (5.60), i.e.,

$$\sum_{i=1}^{K+1} u_{T_i} - K > \sum_{(\mathbf{j_1},\mathbf{j_2})\in\delta(T_1,\ldots,T_{K+1})} e_{\mathbf{j_1}\mathbf{j_2}}, \tag{5.69}$$

then $\sum_{i=1}^{K+1} u_{T_i} \leq K$ can directly be excluded as it implies

$$\sum_{(\mathbf{j_1},\mathbf{j_2})\in\delta(T_1,\ldots,T_{K+1})} e_{\mathbf{j_1}\mathbf{j_2}} < 0, \tag{5.70}$$

which is a contradiction to (5.61).

So, $\sum_{i=1}^{K+1} u_{T_i} > K$ and thus with (5.61) $\sum_{i=1}^{K+1} u_{T_i} = K+1$. Applying (5.69) yields

$$\sum_{(\mathbf{j_1},\mathbf{j_2})\in\delta(T_1,\ldots,T_{K+1})} e_{\mathbf{j_1}\mathbf{j_2}} = 0, \tag{5.71}$$

but (5.60) implies

$$\sum_{(\mathbf{j_1},\mathbf{j_2})\in\delta(T_1,\ldots,T_{K+1})} e_{\mathbf{j_1}\mathbf{j_2}} \geq \sum_{i=1}^{K+1} u_{T_i} - K = 1, \tag{5.72}$$

which is a contradiction.

For simplification and reduction of the number of constraints, the constraints (5.28), (5.29) and (5.30) respectively (5.56), (5.57) and (5.58) can be combined and replaced by a single one:

$$-1 \leq 2e_{\mathbf{j_1}\mathbf{j_2}} - z_{\mathbf{j_1}} - z_{\mathbf{j_2}} \leq 0,\ \forall \mathbf{j}, \mathbf{j_1}, \mathbf{j_2} \in \Pi \text{ with } \mathbf{j_1} \ \mathfrak{n}\ \mathbf{j_2} \tag{5.73}$$

asserting that $e_{\mathbf{j_1}\mathbf{j_2}} = 1$, if and only if $z_{\mathbf{j_1}} = z_{\mathbf{j_2}} = 1$. So, the whole system of inequalities is given by

$$\forall \emptyset \neq T_1, \ldots, T_{K+1} \subset \Pi, \bigcup_{i=1}^{K+1} T_i = \Pi,\ T_i \cap T_j = \emptyset,\ \forall i \neq j,\ i,j \in \{1,\ldots K+1\},$$
$$\forall \mathbf{j}, \mathbf{j_1}, \mathbf{j_2} \in \Pi \text{ with } \mathbf{j_1} \ \mathfrak{n}\ \mathbf{j_2}:$$

$$\begin{aligned}
-1 &\leq 2e_{\mathbf{j_1}\mathbf{j_2}} - z_{\mathbf{j_1}} - z_{\mathbf{j_2}} \leq 0, && (5.74)\\
\frac{1}{|T_i|}\sum_{\mathbf{j}\in T_i} z_{\mathbf{j}} &\leq u_{T_i} \leq \sum_{\mathbf{j}\in T_i} z_{\mathbf{j}}, \quad \forall\, i \in \{1,\ldots K+1\} && (5.75)\\
\sum_{i=1}^{K+1} u_{T_i} - K &\leq \sum_{(\mathbf{j_1},\mathbf{j_2})\in\delta(T_1,\ldots,T_{K+1})} e_{\mathbf{j_1}\mathbf{j_2}} && (5.76)\\
z_{\mathbf{j}}, e_{\mathbf{j_1}\mathbf{j_2}}, u_{T_i} &\in \{0,1\}, \quad \forall\, i \in \{1,\ldots K+1\}. && (5.77)
\end{aligned}$$

5.1 Cutting plane algorithm

As the number of constraints grows exponentially in the number of nodes of the underlying graph, for the implementation of the constraints derived in this chapter, a cutting plane algorithm is used.
Given a solution of one of the phasing problem formulations presented in 2.4, a minimal $K+1$-cut in the corresponding graph defined in 4.2.3 can be calculated, for example using the K-cut algorithm presented in [Goldschmidt and Hochbaum, 1994].
Then, like described in 4.2.9, we can find constraints excluding solutions that violate the inequalities (5.74) to (5.77). New variables $u_i \in \{0,1\}, \quad \forall\ i \in \{1,\dots K+1\}$ are created and the following constraints are added:

$$0 \leq \quad |T_i| \cdot u_i - \sum_{\mathbf{j} \in T_i} z_{\mathbf{j}} \quad \leq \infty, \quad \forall\ i \in \{1,\dots K+1\}, \tag{5.78}$$

$$-\infty \leq \quad u_i - \sum_{\mathbf{j} \in T_i} z_{\mathbf{j}} \quad \leq 0, \quad \forall\ i \in \{1,\dots K+1\}, \tag{5.79}$$

$$-K \leq \sum_{(\mathbf{j_1},\mathbf{j_2}) \in \delta(T_1,\dots,T_{K+1})} e_{\mathbf{j_1 j_2}} - \sum_{i=1}^{K+1} u_i \quad \leq \infty. \tag{5.80}$$

These constraints are valid globally, i.e., they generate cutting planes that are valid for the whole problem.

For the implementation, a different way to generate cutting planes has been chosen. Given a solution of one of the phasing problem formulations presented in 2.4, the current solution values of the variables corresponding to the graph nodes are rounded to be integer. In general, these integer solutions are not feasible any more, but instead of using the K-cut algorithm presented in [Goldschmidt and Hochbaum, 1994], the faster Breadth-First-Search (BFS) algorithm can be used to generate feasible cutting planes.
If there are graph node variables of value one, the number of components of this current binary solution is calculated using a BFS algorithm. Starting with one node with a binary solution value of 1, the BFS-algorithm is performed, considering the graph $G = (V, E)$ containing only nodes and edges with current solution values of 1. After one of the graph's components is explored in this way, starting with an unexplored node with solution value 1, a new BFS is started. Continuing this way, the whole graph with each of its components can be explored and the number of components $\tilde{K} \in \mathbb{N}$ is determined as well as an assignment of the nodes $z_{\mathbf{j}},\ \mathbf{j} \in \Pi$ with current solution value 1 to the different components $T_i,\ i \in \{1,\dots,\tilde{K}\}$. The running time of this algorithm obviously is $O(|V| + |E|)$.
Again, constraints can be found, excluding solutions violating the K-component-constraint and are added after creating the necessary new variables. Instead of

considering only $K+1$ subsets of nodes, all $\tilde{K}$ subsets are considered in the inequalities.

$$0 \leq \quad |T_i| \cdot u_i - \sum_{\mathbf{j} \in T_i} z_{\mathbf{j}} \quad \leq \infty, \quad \forall\, i \in \{1, \dots \tilde{K}\}, \tag{5.81}$$

$$-\infty \leq \quad u_i - \sum_{\mathbf{j} \in T_i} z_{\mathbf{j}} \quad \leq 0, \quad \forall\, i \in \{1, \dots \tilde{K}\}, \tag{5.82}$$

$$-K \leq \sum_{(\mathbf{j_1}, \mathbf{j_2}) \in \delta(T_1, \dots, T_{\tilde{K}})} e_{\mathbf{j_1 j_2}} - \sum_{i=1}^{\tilde{K}} u_i \quad \leq \infty. \tag{5.83}$$

If the electron density solution values of the currently considered solution are all binary and do not satisfy the component constraint, additionally this current solution can be cut off as it is not feasible. Let Π_1 be the set of nodes with a current binary solution value of 1 and Π_0 be the set of nodes with a current binary solution value of 0. Then the following constraint cuts off the current solution and only this one:

$$1 - |\Pi_1| \leq -\sum_{\mathbf{j} \in \Pi_1} z_{\mathbf{j}} + \sum_{\mathbf{j} \in \Pi_0} z_{\mathbf{j}} \leq |\Pi|. \tag{5.84}$$

In this chapter we derived an integer linear program, such that the solution of this problem represents a binary grid electron density distribution satisfying also the topological constraints excluding isolated points, ensuring a minimum covering and restricting the maximum number of connected components. Additionally, a separation algorithm, which is used for the implementation is described. In chapter 6, solution strategies and their implementation will be described.

Chapter 6

Implementation and Computational Results

> Constraint Programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem, the computer solves it.
>
> Eugene C. Freuder

The integer linear programming model derived in chapter 5 has been implemented and tested on real protein data. In this chapter, starting with a short overview over several programming methods and solvers, the implementation, the different tests and their results are presented.

The phase problem is modelled as a Binary Integer Programming Problem in (2.89) to (2.91). After adding an objective function like described in chapter 5, we get the Mixed Integer Programming Problem specified in (2.97) to (2.101), respectively (2.92) to (2.96). Here the variables for the electron density values as well as for the phases are binary, the penalty variables can be arbitrary real variables. The connectivity property can be described by a symbolic constraint, involving a family of linear constraints necessary for its description. Thus, for the implementation a solver is needed, which can handle combined (symbolic) constraint and integer programming problems efficiently.

SCIP and CPLEX/SOPLEX - Combined Integer and Constraint Programming

SCIP is a non-commercial framework for solving mixed-integer programming as well as constraint integer programming problems developed at ZIB - Konrad-Zuse-Zentrum für Informationstechnik Berlin [Achterberg, 2004, 2007; Achterberg et al., 2008]. It is implemented as a C callable library. One main advantage of SCIP is that the user is able to control every step of the solution process. SCIP needs an IP-solver for the IP instances. Different ones are supported, the leading commercial solver CPLEX [CPLEX, 2010], which recently is also free available for academic purposes, as well as the non-commercial solver SOPLEX [Wunderling, 1996] also developed at ZIB have been used for this work.

6.1 Implementation

6.1.1 Data Extraction

In order to get significant results, real protein data have been used. The considered Protein G is shown in Figure 6.1. This is a small protein with 61 residues containing one α-helix and one β-sheet. The protein was crystallized in the space group $P2_12_12_1$ with the unit-cell dimensions $34.9 \times 40.3 \times 42.2$Å. Protein data can be extracted from the Protein Data Bank [Protein Data Bank, 2010]. It contains, amongst others, information about macromolecular structures determined by X-ray diffraction, presented in pdb-format [Berman et al., 2003; Protein Data Bank Contents Guide, 2008]. The pdb-format provides information about atomic coordinates, names of protein molecules, primary and secondary structure information as well as other details about the structures and their solution processes.

A software package 'all_calc_map' developed by A. Urzhumtsev, L. Urzhumtseva, N. Lunina and V. Lunin has been used to extract atomic model information from the Protein Data Bank, to generate a list of reflections for any desired resolution, independently of the resolution in which the reflection data in experiment has been measured, and to calculate the corresponding structure factors. The software also provides the possibility to calculate the grid electron density values from these structure factors for an arbitrary grid satisfying some conditions on the grid size. Starting with these grid electron density files in xplor-format [XPLOR, 1999], the grid structure factors on this grid are calculated. Alternatively a threshold can be specified, absolute or procentual, then the corresponding binary grid electron

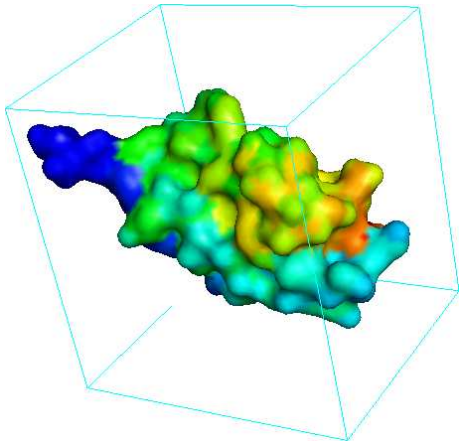


Figure 6.1: Unit cell of Protein G

density values are calculated together with the grid structure values belonging to the binary grid electron density values. A Matlab-program [MATLAB, 2010] has been written, doing all these steps automatically after it has been provided with the necessary informations like the used xplor-file and, if necessary, the threshold. It also stores the structure factors in a *.fc file, which is structured as follows:

Entries in .fc-file	Description
m n p	grid dimensions
V_{cell}	volume of unit cell
t_{nu}	dimensions of unit cell
κ	proportionality factor (2.73)
components	maximum number of components
symmetry	symmetry of grid electron density values
a	a=0 for acentric, a=1 for centric reflections
b	b= 1 if isolated points constraint activated, 0 otherwise
c	c= 1 if covering constraint activated, 0 otherwise
d	d= 1 if component constraints activated, 0 otherwise
1 1 1 $F_g(1,1,1)$	**h** $F_g(\mathbf{h})$:
$\vdots$	array of structure factors
m n p $F_g(m,n,p)$	

6.1.2 Program structure in SCIP

The SCIP-implementation is divided into the files cppmain.cpp, Readerfc.h, Readerfc.cpp, MyGraph.h, MyGraph.cpp, Conshdlrcomp.h, Conshdlrcomp.cpp, see Fig-

ure 6.2.
In *cppmain* a reading method is specified, a SCIP-solving method is started and in the end the output is specified.
In *MyGomoryHuTree*, data structures for a graph, the graph nodes and the graph edges are defined. The structure for the graph contains arrays of graph nodes and graph edges, the number of nodes and edges in the graph as well as the number of components the graph is allowed to consist of. The structure for the graph nodes contains entries necessary for a Breadth First Search in a given graph. Each grid electron density variable is connected with a node. The structure for the graph edges contains pointers to the nodes adjacent to an edge and to the next edge in the incidence list of the node from which the edge emanates. Binary edge variables are assigned to every edge in the graph.
In *Readerfc* the input file, which has to be in the fc-format described above, is read. Then all necessary variables are created. There are $|\Pi|$ binary variables for the grid electron density values, one for every grid point of the grid Π. For the phases, two variables for every grid point, i.e., $2|\Pi|$ variables, have to be introduced. Additionally, there are $2|\Pi|$ real penalty variables, needed in the MIP-formulation specified in (2.92) to(2.96). In a grid with $|\Pi| = m \times n \times p$ grid points, $(m * n * (p-1) + m * (n-1) * p + (m-1) * n * p)$ edges are defined, connecting neighboured grid points and therefore $(m*n*(p-1)+m*(n-1)*p+(m-1)*n*p)$ binary variables assigned to the edges are needed.
After that the constraints are specified, starting with the constraints for the phase calculation specified in chapter 2, namely the constraints in (2.92) to (2.96). In SCIP, the objective function is specified by adding the coefficient of every variable in the objective function to the variable definition.
Depending on which constraints are activated in the input file, additional constraints are called. These are the covering (4.9) and isolated nodes constraints (4.8).
The symbolic constraint 'Kcomponents($\mathbf{X}$)', $\mathbf{X} \in \{0,1\}^{m \times n \times p}$, $m, n, p \in \mathbb{N}$ states, that the binary picture $\mathbf{X}$ contains at most K components.
This constraint can be described by a set of ILP-constraints, developed in chapters 4 and 5. The component constraint is satisfied if these constraints are fulfilled. If the component constraints are activated, for the ILP-constraints whose number grows exponentially in the number of nodes, separation methods like described in 4.2.9 will be used. These separation methods are implemented in SCIP in so-called constraint handlers. The constraints on the graph edges, specified in (5.74), are added, ensuring, that the edges connecting two nodes with variables taking the value 1 get also the value 1 and that the other edges get a value of 0. Then the constraint handler *Conshdlrcomp* is called.
In *Conshdlromp* the global component constraints (5.74) derived in chapter 5 are

implemented in a separation method. As the number of constraints is growing exponentially in the number of nodes, the separation method does not add all constraints, but only violated ones every time it is called.

6.1.3 Constraint handler

In SCIP, a constraint handler defines the algorithms necessary to process all constraints belonging to its constraint class. In the corresponding header-file, the properties of the constraint handler can be adjusted, such as the priorities for separating or checking feasibility as well as the default frequency for separating cuts. With the help of data structures, in C/C++ a set of data elements can be grouped together under one name. In the constraint handler *Conshdlromp*, such a structure SCIP_ConsData is defined containing the data that is needed for defining the constraints belonging to the constraint handler. This data consists of the graph on which the problem is defined and where every graph node and edge is linked to the corresponding binary variable.
Then, different callback methods are defined in the constraint handler. The main methods are *scip_sepalp* and *scip_sepasol*, which generate cutting planes for separating the current solution at different times of the solution process. The method *scip_sepalp* is called, when already a valid LP-solution exists, while *scip_sepasol* separates an arbitrary primal solution. The cutting planes are created like described in 5.1.
In the constraint handler, also a checking method is implemented. The method *scip_check* tests, if the component constraint is fulfilled for the current solution. Here, also the BFS-algorithm is used to calculate the number of components in the current solution and to compare it to the number of components allowed. The return value of the method specifies if the component constraint is satisfied or not.

is then added, such that the K nodes, to which the flow from the startnode is greatest and the startnode are separated by the cut. A checking method checking if the component constraints are already fulfilled has been implemented, if they are not, the separation method adding more constraints from (5.74) is called again.

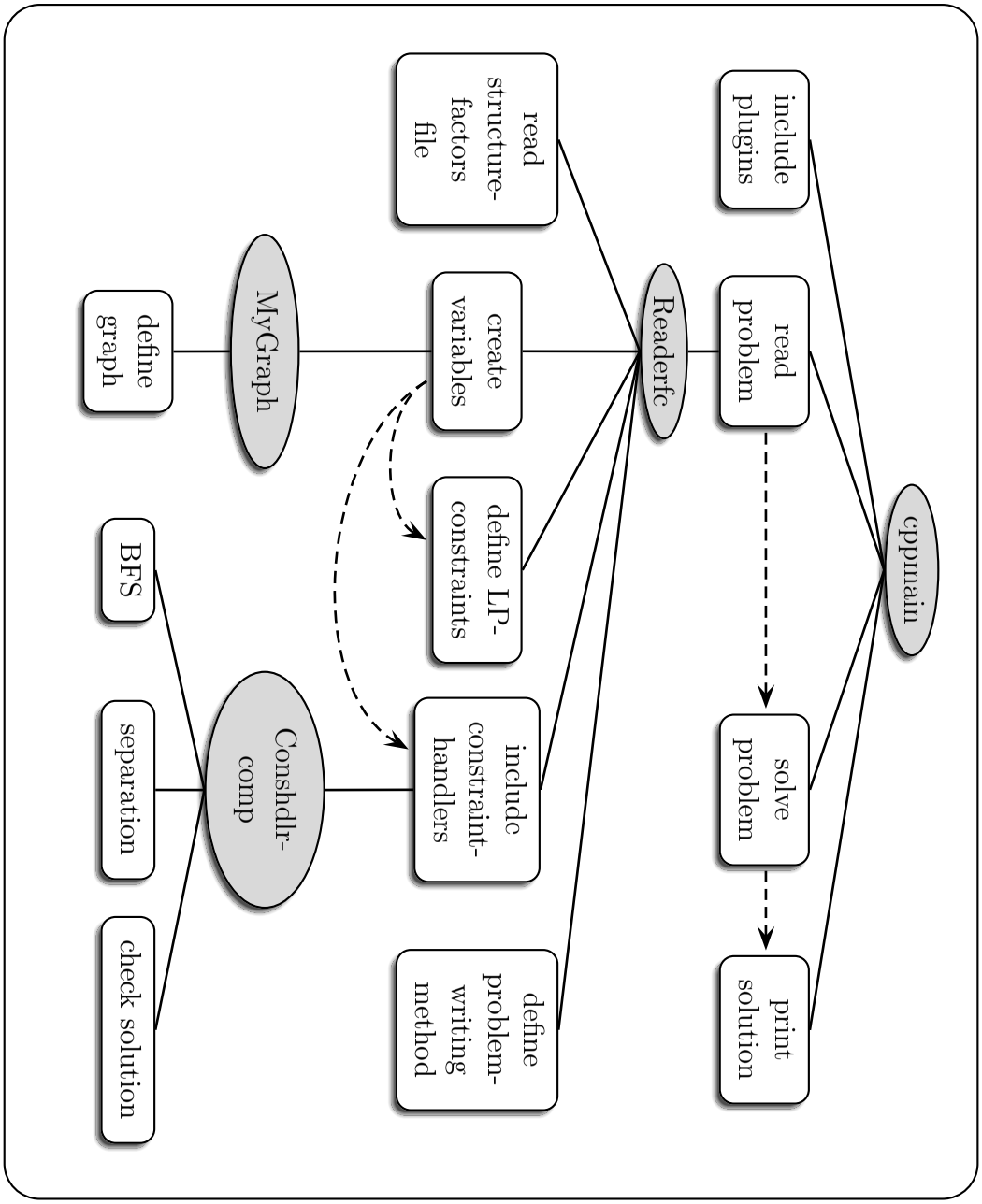


Figure 6.2: Program-structure flowchart

Breaking down the problem

For testing the approach for solving the phase problem introduced before, different simplification steps have been made.
In order to show the effectivity of the described approach, we tried to reconstruct known protein structures taken from the Protein Data Bank [Protein Data Bank, 2010]. The necessary working steps will now be described.

- A protein structure description in pdb-format is taken from the protein data bank [Protein Data Bank, 2010].
- The corresponding electron density distribution is calculated via the software package 'all_calc_map' mentioned above. With the help of these programs one can pass from an atomic model of a crystal to the structure factors corresponding to a given resolution and directly calculate the electron density at given grid points.
- Via Fourier-transformation (cf. chapter 2, equation (2.9)) from this grid electron density, magnitude and phase of the grid structure factors can be calculated.

The reconstruction process will be started with the knowledge of only the magnitudes of the structure factors. Normally X-ray experiments provide intensities of reflections, from which the magnitudes of structure factors can directly be calculated (1.41).
Given only the magnitudes of the structure factors in the reconstruction process, the previously described simplifications during this process result in different errors that will have to be taken into account.

- Grid structure factors belonging to a grid electron-density function are calculated instead of structure factors belonging to the real electron density distribution. This leads to the error estimated in (2.19).
- A binarized electron density distribution is searched for. As the known structure factors' magnitudes belong to a real electron density distribution and not to a binary one, here an error occurs that has to be taken into account (2.73).
- Only a small number of phase values is allowed, resulting in errors (2.47).
- In the formulation of the ILP to be solved a splitting into a real and an imaginary part of the regarded inequality has been done in (2.26) and (2.27).

The other constraints should not lead to errors that have to be taken into account, but only restrict the set of different solutions, satisfying the integer linear programming constraints.

6.1.4 Testing the programs

In order to verify the correctness of the programs and to see, where the simplifications lead to significant errors, tests are made in different steps, each based on the previous one.

1. The given electron density distribution is binarized by defining a threshold and setting all electron density values above the threshold to 1. Starting with this binary electron density distribution the corresponding grid structure factors are calculated. Using the exact phases and magnitudes of the grid structure factors, the picture is reconstructed. Here, the binary electron density can be recalculated exactly.

2. Again starting with a binary electron density distribution, the corresponding grid structure factor magnitudes are calculated. Using these magnitudes and allowing only four different phase values, the reconstruction process is done.

3. In a next step the real electron density distribution is taken and the corresponding grid structure factor magnitudes are calculated. Again, only a small number of phase values is allowed in the reconstruction process.

4. The last step is to take a protein structure description from the protein data bank, calculate the structure factor magnitudes in an arbitrary resolution and use them instead of the grid structure factor magnitudes. Again, only a small number of phase values is allowed in the reconstruction process.

In each step, some or all of the additional constraints, described in chapters 4 and 5 can be added or left out.
Using the whole structure factor values, i.e., magnitudes and phases of the grid structure factors, some tests have been run in order to find out the influence of rounding on the results. Using rounded constants (e.g., using $\pi = 3.14$) has great influence on the results, while rounding the values of the magnitudes of the structure factors keeps the solution quite stable. In case the phases of the grid structure factors are not known, this increases the running time significantly.

6.2 Test results

For each of the 4 test scenarios in 6.1.4, a number of tests has been run with the data of Protein G, using different grid sizes and different molecular volumes specifying the number of non-zero grid values in the binary electron density distribution.
The number of independent grid points has been varied between 64 points ($4\times4\times4$-grid) and 216 grid points ($6 \times 6 \times 6$-grid). Due to the increasing running time, no larger grids have been considered. For the bigger grid size with 216 independent grid points, different resolutions have been chosen, varying from 4 to 16Å, the low-resolution cut-off limit is chosen in a way, that it does not exclude any structure factor values, see (1.40). For the smaller grid size with 64 independent grid points, the change of the resolution does not show any effect on the binary grid electron density values and thus has not been considered.
In general, the asymmetric unit of a unit cell containing the independent grid points is smaller than the whole unit cell. For e.g. space group $P2_12_12_1$, the asymmetric unit is only a quarter of the unit cell.
Depending on the space group, different choices of the unit cell origin are possible. If only the amplitudes of the Fourier coefficients are known, different phase sets may lead to identical images except that the origin is shifted from one point to another [Lunin and Lunina, 1996]. These possible shiftings have not been taken into account. Including them when evaluating the results could lead to even better results.
For the implementation, SCIP Version 1.2.0 [Achterberg, 2007] has been used with CPLEX [CPLEX, 2010], version 11.0 as IP-solver for the IP instances. The solutions have been calculated on a i686 with 4 processors, a 3GHz CPU and 3GB RAM. If we start with the magnitudes of grid structure factors belonging to a binary electron density distribution, the running time to calculate a solution on a $4 \times 4 \times 4$-grid, i.e., with 64 independent grid points, takes about 30 seconds CPU time without additional constraints and about 40 seconds with all constraints added. On a $6 \times 6 \times 6$-grid, i.e., with 216 independent grid points, it takes about 10 minutes CPU time to calculate a solution without additional constraints and about 50 minutes CPU time with all constraints added.
Starting with magnitudes of grid structure factors belonging to a real electron density distribution increases the running time significantly - also without additional constraints, the calculation of a solution on a $4 \times 4 \times 4$-grid with all constraints takes 40 minutes and on a $6 \times 6 \times 6$-grid 7,6 hours. The creation of a new cutting plane for this grid size takes less than a second, such that the main amount of time is taken by the ILP-solver.
For calculating a set of best solutions and not only one solution, a shell-script has

been used which, each time a solution has been found, automatically changes the file Reader_fc.cpp. An additional constraint is added, cutting off the current solution for the grid electron density values. Here again, inequality (5.84) is used. Then the SCIP-code is compiled again and a solution of the new problem is searched. Figure 6.3 shows, which steps are necessary for the whole solution and evaluation process and which programs and which file-formats are used.

6.2.1 Evaluation of the results

Once a set of solutions has been calculated, the quality of those solutions has to be evaluated. Therefore, in a Matlab-program [MATLAB, 2010], the exact electron density is calculated on the specified grid Π. Using the minimal molecular volume that has been defined in the solution process to specify the number of non-zero grid values, this electron density distribution is binarised. The distance $D(\rho_{exact}, \rho^i_{calc})$ between the resulting binary electron density ρ_{exact} and the calculated ones $\rho_{calc}(i)$, $i \in \{1, \ldots, N\}$, $N \in \mathbb{N}$, is defined using the Hamming distance:

$$D(\rho_{exact}, \rho^i_{calc}) \stackrel{def}{=} \sum_{\mathbf{j} \in \Pi} \left| \rho_{exact}(\mathbf{j}) - \rho^i_{calc}(\mathbf{j}) \right|. \tag{6.1}$$

The smaller the distance value, the better the quality of the considered solution. The smallest distance reached by one of the test runs is $D_{min} \stackrel{def}{=} \min_{i=1}^{N} D(\rho_{exact}, \rho^i_{calc})$. As the exact solution normally is not known in advance, one solution has to be calculated from the set of solutions we get. The methods we used are similar to an approach suggested by V.Y. Lunin in [Lunin, 2003] for the one-dimensional case and have been presented in [Heldt and Bockmayr, 2010].
One possibility to calculate such an average solution for the set of $N \in \mathbb{N}$ solutions is the following one:

$$\rho_{av}(\mathbf{j}) \stackrel{def}{=} \frac{1}{N} \sum_{i=1}^{N} \rho^i_{calc}(\mathbf{j}), \ \forall \mathbf{j} \in \Pi. \tag{6.2}$$

Obviously, in general ρ_{av} is not a binary function. Using the defined molecular volume value, it can be binarised, resulting in a solution ρ^{bin}_{av} and it can be compared to the exact solution:

$$D_{av} \stackrel{def}{=} D(\rho_{exact}, \rho^{bin}_{av}). \tag{6.3}$$

Another possibility would be to choose the solution to which the distance of all other solutions is smallest. Therefore, the distance of every solution to all others has to be calculated. For every solution, the distances to all others are summed

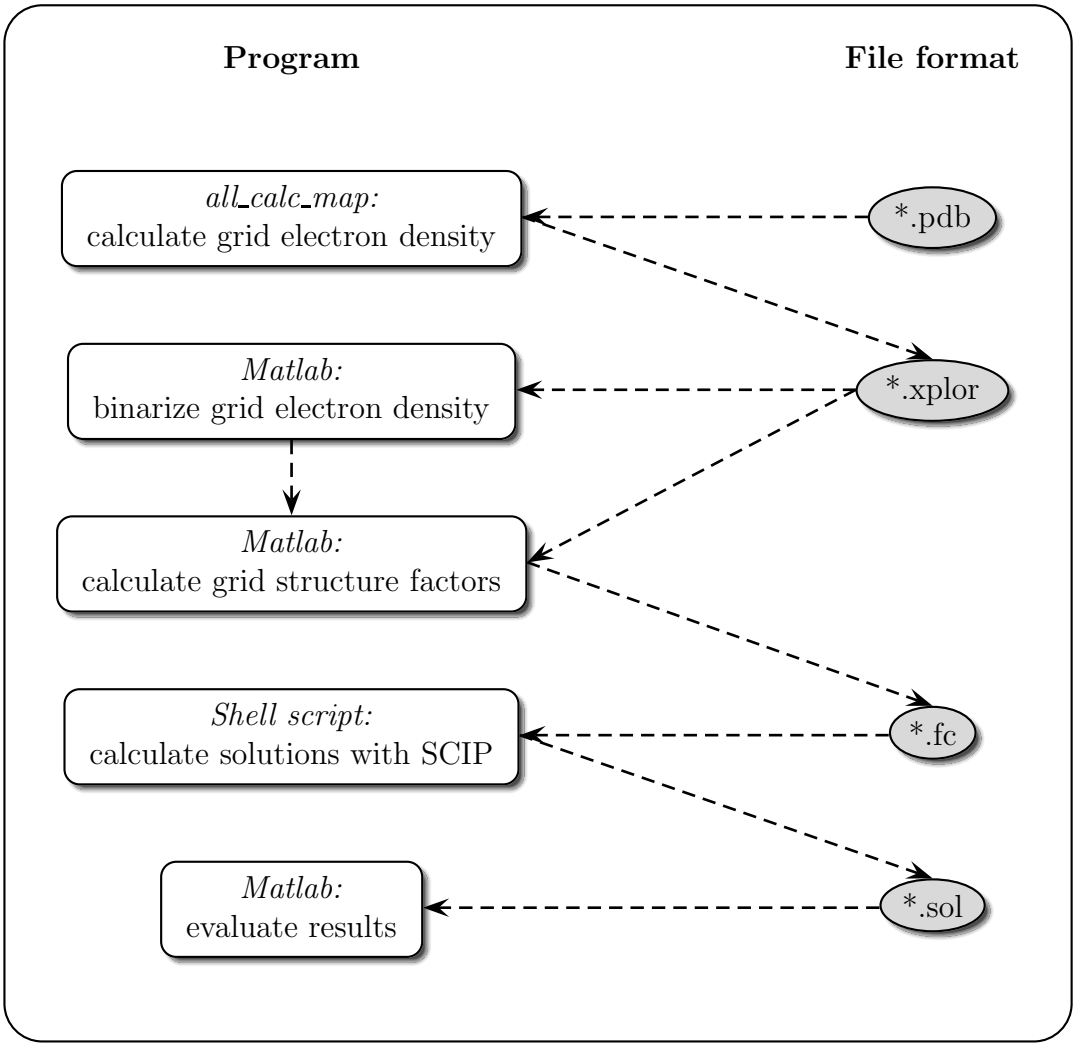


Figure 6.3: Evaluation process and file formats

up, the solution, for which this sum is the smallest, is chosen as reference solution ρ_{ref}:

$$D_{sum}(i) \stackrel{def}{=} \sum_{k=1}^{N} \sum_{\mathbf{j}\in\Pi} \left|\rho^i_{calc}(\mathbf{j}) - \rho^k_{calc}(\mathbf{j})\right|, \ \forall i \in \{1,\ldots,N\}, \tag{6.4}$$

$$\rho_{ref} = \rho^i_{calc}, \text{ with } D_{sum}(i) \stackrel{def}{=} \min_{k=1}^{N}\{D_{sum}(k)\}, \ D_{ref} \stackrel{def}{=} D(\rho_{exact}, \rho_{ref}). \tag{6.5}$$

Additionally, the number of connected components is counted for the binarised original electron density distribution as well as for the best calculated binary solution ρ_{min} and for the solutions ρ_{av} and ρ_{ref}.
In the tables below, the results of the tests on a $4 \times 4 \times 4$-grid, i.e., a grid with 64 independent grid points and on $6\times6\times6$-grids, i.e., grids with 216 independent grid points are shown. In the latter case, different resolutions have been considered.
The first column of these tables specifies the grid size, the second the percentage of non-zero grid points, the third the resolution, the fourth the number of connected components of the binarised original electron density distribution. In the fifth column, the used topological constraints are specified - either none, or only the constraint excluding isolated points (iso) or the connectivity constraint (connected). If the connectivity constraint is activated, in brackets the maximum number of components allowed is given.
The sixth column specifies the number of test solutions satisfying these constraints. In columns seven to nine, the percentages of correct solution values using the different distance measures D_{min}, $D_{av} \stackrel{def}{=} D(\rho_{exact}, \rho_{av})$ and $D_{ref} \stackrel{def}{=} D(\rho_{exact}, \rho_{ref})$ are listed, i.e.

$$p_{min} = \frac{|\Pi| - D_{min}}{|\Pi|}, \ p_{av} = \frac{|\Pi| - D_{av}}{|\Pi|}, \ p_{ref} = \frac{|\Pi| - D_{ref}}{|\Pi|}. \tag{6.6}$$

The numbers in brackets are the numbers of components in the corresponding solutions.
For the small grid size of $4\times4\times4$, the 100 best solutions and for the bigger $6\times6\times6$ grid, the 70 best solutions were considered. While the exclusion of isolated points does not seem to have much influence on the results, the restriction of the number of components reduces the number of reasonable solutions significantly.
The tests in the tables 6.1 and 6.2 have been done starting with the magnitudes of grid structure factors belonging to a binary electron density distribution.
In table 6.3, the results using the grid structure factors magnitudes belonging to a real electron density distribution are presented and compared to the results we get in the same test case using the ones belonging to a binary one.

Grid size	covering	resolution	K_{exact}	constraints	# sol	p_{min} (K_{min})	p_{av} (K_{av})	p_{ref} (K_{ref})
$4 \times 4 \times 4$	10%	12-999	1	none	100	92% (1)	78% (1)	77% (1)
$4 \times 4 \times 4$	10%	12-999	1	iso	100	92% (1)	78% (1)	77% (1)
$4 \times 4 \times 4$	10%	12-999	1	connected (2)	90	92% (1)	78% (1)	77% (1)
$4 \times 4 \times 4$	10%	12-999	1	connected (1)	46	92% (1)	84% (1)	80% (1)
$4 \times 4 \times 4$	10%	12-999	1	iso, connected (2)	90	92% (1)	78% (1)	77% (1)
$4 \times 4 \times 4$	10%	12-999	1	iso, connected (1)	46	92% (1)	84% (1)	80% (1)
$4 \times 4 \times 4$	25%	12-999	1	none	100	89% (1)	69% (1)	64% (1)
$4 \times 4 \times 4$	25%	12-999	1	iso	100	89% (1)	69% (1)	64% (1)
$4 \times 4 \times 4$	25%	12-999	1	connected (2)	97	89% (1)	69% (1)	64% (1)
$4 \times 4 \times 4$	25%	12-999	1	connected (1)	74	89% (1)	81% (1)	83% (1)
$4 \times 4 \times 4$	25%	12-999	1	iso, connected (2)	97	89% (1)	69% (1)	64% (1)
$4 \times 4 \times 4$	25%	12-999	1	iso, connected (1)	74	89% (1)	81% (1)	83% (1)
$4 \times 4 \times 4$	30%	12-999	1	none	100	81% (1)	75% (2)	81% (1)
$4 \times 4 \times 4$	30%	12-999	1	iso	100	81% (1)	75% (2)	81% (1)
$4 \times 4 \times 4$	30%	12-999	1	connected (2)	96	81% (1)	75% (2)	81% (1)
$4 \times 4 \times 4$	30%	12-999	1	connected (1)	71	81% (1)	78% (1)	81% (1)
$4 \times 4 \times 4$	30%	12-999	1	iso, connected (2)	96	81% (1)	75% (2)	81% (1)
$4 \times 4 \times 4$	30%	12-999	1	iso, connected (1)	71	81% (1)	78% (1)	81% (1)

Table 6.1: Results on a grid with 64 independent grid points, calculation started with structure factor magnitudes belonging to a binary grid electron density distribution

Grid size	covering	resolution	K_{exact}	constraints	# sol	p_{min} (K_{min})	p_{av} (K_{av})	p_{ref} (K_{ref})
$6 \times 6 \times 6$	30%	4-999	2	none	70	73% (1)	80% (2)	62% (2)
$6 \times 6 \times 6$	30%	4-999	2	iso	67	73% (1)	80% (1)	62% (2)
$6 \times 6 \times 6$	30%	4-999	2	connected (3)	55	73% (1)	74% (1)	67% (1)
$6 \times 6 \times 6$	30%	4-999	2	connected (2)	49	73% (1)	74% (1)	67% (1)
$6 \times 6 \times 6$	30%	4-999	2	iso, connected (3)	55	73% (1)	74% (1)	67% (1)
$6 \times 6 \times 6$	30%	4-999	2	iso, connected (2)	49	73% (1)	74% (1)	67% (1)
$6 \times 6 \times 6$	30%	12-999	1	none	70	72% (1)	56% (4)	54% (2)
$6 \times 6 \times 6$	30%	12-999	1	iso	67	72% (1)	62% (2)	54% (2)
$6 \times 6 \times 6$	30%	12-999	1	connected (2)	49	72% (1)	66% (1)	63% (1)
$6 \times 6 \times 6$	30%	12-999	1	connected (1)	28	72% (1)	74% (1)	65% (1)
$6 \times 6 \times 6$	30%	12-999	1	iso, connected (2)	47	72% (1)	69% (1)	68% (1)
$6 \times 6 \times 6$	30%	12-999	1	iso, connected (1)	26	72% (1)	74% (1)	70% (1)

Table 6.2: Results on a grid with 216 independent grid points, calculation started with structure factor magnitudes belonging to a binary grid electron density distribution

Grid size	covering	resolution	K_{exact}	constraints	# sol	p_{min} (K_{min})	p_{av} (K_{av})	p_{ref} (K_{ref})
$4 \times 4 \times 4$	30%	12-999	1	none	100	81% (1)	72% (1)	63% (1)
$4 \times 4 \times 4$	30%	12-999	1	iso	100	81% (1)	72% (1)	63% (1)
$4 \times 4 \times 4$	30%	12-999	1	connected (2)	100	81% (1)	72% (1)	63% (1)
$4 \times 4 \times 4$	30%	12-999	1	connected (1)	86	81% (1)	75% (1)	63% (1)
$4 \times 4 \times 4$	30%	12-999	1	iso, connected (2)	100	81% (1)	72% (1)	63% (1)
$4 \times 4 \times 4$	30%	12-999	1	iso, connected (1)	86	81% (1)	75% (1)	63% (1)
$6 \times 6 \times 6$	30%	12-999	1	none	30	72% (1)	53% (4)	55% (4)
$6 \times 6 \times 6$	30%	12-999	1	iso	30	72% (1)	53% (4)	55% (4)
$6 \times 6 \times 6$	30%	12-999	1	connected (2)	23	72% (1)	55% (4)	40% (2)
$6 \times 6 \times 6$	30%	12-999	1	connected (1)	6	72% (1)	63% (1)	72% (1)
$6 \times 6 \times 6$	30%	12-999	1	iso, connected (2)	23	72% (1)	55% (4)	40% (2)
$6 \times 6 \times 6$	30%	12-999	1	iso, connected (1)	6	72% (1)	63% (1)	72% (1)

Table 6.3: Results of a calculation started with structure factor magnitudes belonging to a real grid electron density distribution

The tables show that in general, the values of p_{av} as well as p_{ref} increase by adding stricter constraints, displaying the increasing quality of the regarded solutions. Especially the component-constraint reduces the number of reasonable solutions and increases the quality of the remaining ones significantly.
The last table shows, that also starting with the structure factor magnitudes belonging to a real electron density distribution and, due to the running time, only a small number of calculated solutions, quite good results can be expected.

Chapter 7

A Different Approach: Integer Points in Polyhedra

> The best way to have a good idea is to have lots of ideas.
>
> Linus Pauling

This chapter covers an approach to find integer points in polyhedra. This differs from the usual methods to solve integer linear programs, implemented in standard ILP-solvers. Its main advantage is, that instead of finding one optimal solution, we get all feasible solutions of the original phase problem stated in (2.89) to (2.91). Starting with the phase problem defined in (2.89) to (2.91), it will be shown that this problem can be described by a minimization problem, whose set of solutions is an ellipsoid. Then two different methods to search for integer points in ellipsoids are shortly presented and finally the whole procedure is illustrated in a flow chart. The results presented in this chapter are especially of interest, if not only binary values for the grid electron density distribution are searched, but a discretisation $\rho(\mathbf{j})_g \in \{1, \ldots, k\}$, $\forall \mathbf{j} \in \Pi$ and a $k \in \mathbb{N}$.

7.1 Application in X-ray crystallography

Using the notations from chapter 2, we define

$$A \stackrel{def}{=} \begin{pmatrix} a^R_{\mathbf{j}}(\mathbf{h}) & \mathrm{diag}(-2b^R_{\mathbf{h}}) & 0 \\ a^I_{\mathbf{j}}(\mathbf{h}) & 0 & \mathrm{diag}(-2b^I_{\mathbf{h}}) \\ & \mathcal{I}_n & \end{pmatrix}, \tag{7.1}$$

$$b \stackrel{def}{=} \begin{pmatrix} -c^R_{\mathbf{h}} \\ -c^I_{\mathbf{h}} \\ \mathbf{1}_n \end{pmatrix}, \; x \stackrel{def}{=} \begin{pmatrix} (z(\mathbf{j})) \\ (y^R_{\mathbf{h}}) \\ (y^I_{\mathbf{h}}) \end{pmatrix}, \; \varepsilon \stackrel{def}{=} \begin{pmatrix} \varepsilon_{\mathbf{h}} \\ \mathbf{1}_n \end{pmatrix}. \tag{7.2}$$

Then the set of variables satisfying the constraints for the phase problem (2.89) to (2.91) is described by

$$\{x \in \{0,1\}^n \;\mid\; |Ax - b| \leq \varepsilon\}. \tag{7.3}$$

The last n inequalities of the inequality system, namely

$$|\mathfrak{I}_n x - \frac{1}{2} \cdot \mathbf{1}_n| \leq \frac{1}{2} \cdot \mathbf{1}_n, \tag{7.4}$$

ensure, that all variables are bounded with 0 being the lower and 1 being the upper bound.
Regarding each row of the inequality system $|Ax - b| \leq \varepsilon$, we get

$$\begin{aligned} |a_1^T x &- b_1| \leq \varepsilon_1, \\ |a_2^T x &- b_2| \leq \varepsilon_2, \\ &\vdots \\ |a_m^T x &- b_m| \leq \varepsilon_m. \end{aligned}$$

This inequality system can also be written in the form

$$\begin{aligned} |\frac{1}{\varepsilon_1} a_1^T x &- \frac{b_1}{\varepsilon_1}| \leq 1, \\ |\frac{1}{\varepsilon_2} a_2^T x &- \frac{b_2}{\varepsilon_2}| \leq 1, \\ &\vdots \\ |\frac{1}{\varepsilon_m} a_m^T x &- \frac{b_m}{\varepsilon_m}| \leq 1, \end{aligned}$$

which is equivalent to

$$\max_{i\in\{1,\ldots,m\}} |\frac{1}{\varepsilon_i} a_i^T x - \frac{b_i}{\varepsilon_i}| \leq 1. \tag{7.5}$$

So, using the maximum norm we can write

$$\|A_\varepsilon x - b_\varepsilon\|_\infty \leq 1, \tag{7.6}$$

with

$$A_\varepsilon = \begin{pmatrix} \frac{a_{11}}{\varepsilon_1} & \frac{a_{12}}{\varepsilon_1} & \cdots & \frac{a_{1n}}{\varepsilon_1} \\ \frac{a_{21}}{\varepsilon_2} & \frac{a_{22}}{\varepsilon_2} & \cdots & \frac{a_{2n}}{\varepsilon_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{m1}}{\varepsilon_m} & \frac{a_{m2}}{\varepsilon_m} & \cdots & \frac{a_{mn}}{\varepsilon_m} \end{pmatrix},$$

$$\begin{aligned} b_\varepsilon^T &= (\frac{b_1}{\varepsilon_1} \quad \frac{b_2}{\varepsilon_2} \quad \ldots \quad \frac{b_m}{\varepsilon_m}), \\ x^T &= (x_1 \quad x_2 \quad \ldots x_n). \end{aligned} \tag{7.7}$$

So, in the context of the phase problem, we search the integer points in the set

$$S = \{x \in \mathbb{R}^n \mid \|A_\varepsilon x - b_\varepsilon\|_\infty \leq \mathbf{1}\}. \tag{7.8}$$

Subsequently, in this chapter we will show methods to find the set

$$S = \{x \in \mathbb{Z}^n \mid \|A_\varepsilon x - b_\varepsilon\|_2 \leq \mathbf{1}\}. \tag{7.9}$$

So, the relations between the euclidean norm used in those methods and the maximum norm, we used in (7.8) to describe the phase problem, have to be considered. On $\mathbb{R}^n$ the euclidean and the maximum norm are equivalent and the following estimations are valid for $x \in \mathbb{R}^n$ [Plato, 2000]:

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty. \tag{7.10}$$

So, every vector contained in the set

$$L_2 \stackrel{def}{=} \{x \in \mathbb{R}^n \mid \|Ax - b\|_2 < \varepsilon\}, \tag{7.11}$$

is also contained in

$$L_\infty \stackrel{def}{=} \{x \in \mathbb{R}^n \mid \|Ax - b\|_\infty < \varepsilon\}. \tag{7.12}$$

Vice versa, every vector contained in

$$\{x \in \mathbb{R}^n \mid \|Ax - b\|_\infty < \varepsilon\} \tag{7.13}$$

is also contained in

$$L_2^{\sqrt{n}} \stackrel{def}{=} \left\{x \in \mathbb{R}^n \mid \|Ax - b\|_2 < \sqrt{n}\varepsilon\right\}. \tag{7.14}$$

So, by deciding to calculate either L_2 or $L_2^{\sqrt{n}}$ one decides to get only vectors also contained in L_∞, but not all of them, or to get all vectors contained in L_∞, but also some others.

7.2 Singular value decomposition, normal pseudosolution and perturbed systems

In the following, a diagonal matrix $(a_{ij}) \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}$ with entries

$$a_{ij} = \begin{cases} \alpha_k \in \mathbb{R}, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases} \tag{7.15}$$

with $1 \leq k \leq n$, will be written as $\text{diag}(\alpha_1, \ldots, \alpha_n)$.

Every matrix $A \in \mathbb{R}^{m\times n}$ can be decomposed in the following way, this decomposition is called *singular value decomposition* (SVD) [Hämmerlin and Hoffmann, 1994].

$$A = U\Sigma V^T,\ U \in \mathbb{R}^{m\times m},\ \Sigma \in \mathbb{R}^{m\times n},\ V \in \mathbb{R}^{n\times n}. \tag{7.16}$$

U and V are orthogonal matrices, that means $UU^T = I_m$, where I_m denotes the $m \times m$-identity matrix. Analogous for V, we get $VV^T = I_n$. The columns v^i of V are the orthonormalized eigenvectors of A^TA.
The matrix Σ is a diagonal matrix with

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k, 0, \ldots, 0),\ \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > 0. \tag{7.17}$$

The diagonal values $\sigma_i,\ i \in \{1, \ldots k\}$ of Σ are the singular values of A, i.e., the square roots of the non-zero eigenvalues of A^TA.

Now the following least-squares minimization problem will be regarded:

Definition 7.1. Minimization problem,
[Hämmerlin and Hoffmann, 1994]

Given $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and using the spectral norm, which is the matrix norm induced by the euclidean vector norm, the problem to find an $x \in \mathbb{R}^n$ fulfilling

$$\|Ax - b\|_2^2 = \inf_{\overline{x}\in\mathbb{R}^n} \|A\overline{x} - b\|_2^2 \tag{7.18}$$

is called least-squares minimization problem.

Using singular value decomposition, A can be decomposed: $A = U\Sigma V^T$, $U \in \mathbb{R}^{m\times m}$, $\Sigma \in \mathbb{R}^{m\times n}$, $V \in \mathbb{R}^{n\times n}$. Due to the orthogonality of U, the equation $\|x\|_2 = \|U^T x\|_2$ holds for all $x \in \mathbb{C}^n$. Setting $z \stackrel{def}{=} V^T x$, $d \stackrel{def}{=} U^T b$ and taking into account the orthogonality of U, one gets:

$$\|Ax - b\|_2^2 = \|U^T(Ax - b)\|_2^2 = \|\Sigma V^T x - U^T b\|_2^2 = \|\Sigma z - d\|_2^2. \tag{7.19}$$

Let r be the rank of A. Due to [Hämmerlin and Hoffmann, 1994], the set of solutions L^2_{min} of the least-squares minimization problem can be determined by:

$$\begin{aligned} L^2_{min} = \{ \quad & x \in \mathbb{R}^n \quad | \quad \|Ax - b\|_2^2 = \inf_{\overline{x}\in\mathbb{R}^n} \|A\overline{x} - b\|_2^2 , \\ & A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m \}, \qquad (7.20) \\ = \ & \left\{ x \in \mathbb{R}^n \ \middle| \ x = \sum_{i=1}^{r} \frac{1}{\sigma_i} d_i v^i + \ker(A) \right\}. \qquad (7.21) \\ & \qquad (7.22) \end{aligned}$$

The set of solutions L^2_{min} forms an affine vector space of dimension $\dim(\ker(A))$. One of these solutions is the normal pseudosolution x^+ given by

$$x^+ = \sum_{i=1}^{r} \frac{1}{\sigma_i} d_i v^i. \tag{7.23}$$

There exists exactly one normal pseudosolution x^+ of the minimization problem [Hämmerlin and Hoffmann, 1994].
Therefore a linear mapping $b \mapsto x^+$, $\mathbb{R}^m \to \mathbb{R}^k$ can be defined via

$$x^+ = \sum_{i=1}^{r} \frac{1}{\sigma_i} d_i v^i = \sum_{i=1}^{r} \frac{1}{\sigma_i} (U^T b)_i v^i. \tag{7.24}$$

This mapping is given by the pseudoinverse of A, denoted as $A^+ \in \mathbb{R}^{m\times n}$ and satisfying $A^+ b = x^+$.

The matrix $A^+ \in \mathbb{R}^{m \times n}$ can be decomposed in the following way [Hämmerlin and Hoffmann, 1994]:

$$A^+ = V \Sigma^+ U^T, \tag{7.25}$$

with

$$\Sigma^+ = \mathrm{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_k^{-1}, 0, \ldots, 0) \tag{7.26}$$

and U and V defined like above in the singular value decomposition of A.

7.3 Inequalities

Now, the singular value decomposition will be applied to find integer solutions of the phase problem defined in (2.89) to (2.91). We will show, that the set

$$L_\varepsilon = \{x \in \mathbb{R}^n \mid \|Ax - b\|_2 < \varepsilon\}, \tag{7.27}$$

with $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $\varepsilon \in \mathbb{R}^+$ describes an ellipsoid and thus, methods for finding integer points in ellipsoids can be used to find the integer solutions. The inequalities (7.4) ensure, that the set L_ε is bounded, as they force all variables to take values between 0 and 1.

Theorem 7.2.

If the set

$$L_\varepsilon = \{x \in \mathbb{R}^n \mid \|Ax - b\|_2 < \varepsilon\} \tag{7.28}$$

is bounded, it describes the inner part of the ellipsoid

$$E = \left\{x \in \mathbb{R}^n \quad | \quad x^T Q^{-1} x \leq 1\right\}, \text{ with } Q^{-1} \stackrel{def}{=} \frac{1}{\varepsilon^2} A^T A. \tag{7.29}$$

The center of the ellipsoid is given by x^+.

Proof: Following the argumentation in [Überhuber, 1995], showing that a sphere is mapped by linear mapping maps onto an ellipsoid, we show that the preimage of the sphere $\{y \in \mathbb{R}^n \mid \|y\|_2 < \varepsilon,\ y = Ax - b\}$ is an ellipsoid, if this preimage is assumed to be bounded.
With $y \stackrel{def}{=} Ax - b$ and $A = U \Sigma V^+$, we get $\Sigma V^T x = U^T (y + b)$. By defining $x' = V^T x$

and $y' = U^T(y+b)$ and thus $\Sigma x' = y'$, we change the regarded coordinate systems. Then

$$\begin{pmatrix} \sigma_1 x'_1 \\ \vdots \\ \sigma_r x'_r \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} y'_1 \\ \vdots \\ y'_m \end{pmatrix}. \tag{7.30}$$

So,

$$x'_i = \frac{y'_i}{\sigma_i},\ i \in \{1,\dots,r\}, \quad x'_i \in \mathbb{R},\ i \in \{r+1,\dots,n\}. \tag{7.31}$$

If the set $L_\varepsilon = \{x \in \mathbb{R}^n \mid \|Ax-b\|_2 < \varepsilon\}$ is bounded, there are no eigenvalues of value zero and thus $r = n$. The preimage of the set $\{y' \in \mathbb{R}^m \mid \|y'\|_2 < \varepsilon\}$ is given by

$$\{x' \in \mathbb{R}^n \mid \sum_{i=1}^{r} \left(\frac{y'_i}{\sigma_i}\right)^2 < \varepsilon\}, \tag{7.32}$$

which describes an ellipsoid.

$$\begin{aligned}
& \|Ax-b\|_2 < \varepsilon,\ \varepsilon > 0 \\
\Rightarrow\ & \|Ax-b\|_2^2 < \varepsilon^2 \\
\Leftrightarrow\ & (Ax-b)^T(Ax-b) < \varepsilon^2 \\
\Leftrightarrow\ & (U\Sigma V^T x - b)^T(U\Sigma V^T x - b) < \varepsilon^2 \\
\Leftrightarrow\ & (U\Sigma V^T x - U\Sigma V^T V\Sigma^+ U^T b)^T(U\Sigma V^T x - U\Sigma V^T V\Sigma^+ U^T b) < \varepsilon^2 \\
\Leftrightarrow\ & (x - V\Sigma^+ U^T b)^T (U\Sigma V^T)^T (U\Sigma V^T)(x - V\Sigma^+ U^T b) < \varepsilon^2 \\
\Leftrightarrow\ & (x - A^+ b)^T A^T A (x - A^+ b) < \varepsilon^2 \\
\Leftrightarrow\ & (x - x^+)^T A^T A (x - x^+) < \varepsilon^2 \\
\Leftrightarrow\ & (x - x^+)^T Q (x - x^+) < 1
\end{aligned} \tag{7.33}$$

The previous observations show that $L_\varepsilon = \{x \in \mathbb{R}^n \mid \|Ax-b\|_2 < \varepsilon\}$ describes the inner part of an ellipsoid and thus, the problem of finding the set of integer points satisfying $\{x \in \mathbb{Z}^n \mid \|Ax-b\|_2 < \varepsilon\}$ can be solved by methods to find integer points in ellipsoids.
If we consider only binary vectors in the set L_ε, i.e., we try to find the set $L_\varepsilon^{bin} = \{x \in \{0,1\}^n \mid \|Ax-b\|_2 < \varepsilon\}$, then as

$$\begin{aligned}
L_\varepsilon &= \{x \in \mathbb{R}^n \mid \|Ax-b\|_2 < \varepsilon\} \\
&= \left\{x^+ + y \mid y \in \tilde{L}\right\},
\end{aligned}$$

with

$$\tilde{L} \stackrel{def}{=} \left\{ y \in \mathbb{R}^n \;\mid\; \|Ay\|_2^2 < \varepsilon' \right\} \text{ with } \varepsilon' \stackrel{def}{=} \varepsilon^2 - \|Ax^+ - b\|_2^2,$$

we get

$$\begin{aligned} L_\varepsilon^{bin} &= \left\{ (x^+ + y) \in \{0,1\}^n \;\mid\; y \in \tilde{L} \right\} \\ &= \left\{ (x^+ + y),\; y_i \in \{-x_i^+, 1 - x_i^+\} \;\mid\; y \in \tilde{L} \right\}. \end{aligned}$$

7.4 Integer points in an ellipsoid

There exist different methods for finding integer points in ellipsoids. They estimate the set $L_\varepsilon^{bin} = \{x \in \mathbb{Z}^n \;\mid\; \|Ax - b\|_2 < \varepsilon\}$.

These methods use transformations of the ellipsoid, such that an efficient algorithm can be found allowing to find the points in the transformed object that correspond to the searched integer values.

for any given linearly independent vectors $b_1, \ldots, b_l \in \mathbb{Q}^l$, finds a reduced basis of the lattice $\Lambda(b_1, \ldots, b_l) \subseteq \mathbb{R}^n$.

Reduced basis in non-euclidean norm

Let a positive definite rational $n \times n$-matrix Q be given. Then a Gram-Schmidt orthogonalization with respect to orthogonality $\perp$ relative to the inner product defined by Q and a norm $\|\cdot\|_Q$ defined by Q can be defined [Schrijver, 1989]:

$$\|x\|_Q \stackrel{def}{=} \sqrt{x^T Q x},\; \forall x \in \mathbb{R}^n \tag{7.34}$$

$$x \perp y \Leftrightarrow x^T Q y = 0,\; \forall x, y \in \mathbb{R}^n. \tag{7.35}$$

Such a reduced basis consists of basis vectors that are as parallel as possible to the main axes of the ellipsoid described by

$$\{x \in \mathbb{R}^n \;\mid\; x^T Q x \leq \chi\},\; \chi \in \mathbb{R}, \tag{7.36}$$

see [Aardal et al., 2002; Cook et al., 1993].

Definition 7.3. Reduced basis with respect to the inner product *defined by* Q

A basis $b_1, \dots b_l$ of a lattice $\Lambda \subset \mathbb{R}^n$ is called reduced *in the sense of Lovász with respect to the inner product defined by Q, if and only if*

$$|\lambda_{ij}| \leq \frac{1}{2} \qquad \text{for } 1 \leq i < j \leq l \tag{7.37}$$

$$\|b_i^* + \lambda_{i,i-1} b_{i-1}^*\|_Q^2 \geq \frac{3}{4} \|b_{i-1}^*\|_Q^2 \qquad \text{for } 1 < i \leq l. \tag{7.38}$$

Here, $b_i = \lambda_1 b_1^* + \ldots + \lambda_{i-1} b_{i-1}^* + b_i^*$.

Let be B_i the matrix with columns $b_1, \dots, b_i$, where $b_1, \dots, b_i$ are the initial lattice basis vectors. The Gram Schmidt orthogonalization process [Plato, 2000] iteratively calculates the Gram-Schmidt-basis vectors $b_1^*, \dots, b_l^*$:

$$b_i^* = b_i - B_{i-1}(B_{i-1}^T Q B_{i-1})^{-1} B_{i-1}^T Q b_i. \tag{7.39}$$

The calculated basis fulfils $b_i^* \perp b_j^*,\ \forall i, j \in \{1, \dots, n\},\ i \neq j$.
The LLL-basis reduction method with respect to Q is a generalization of the LLL-basis reduction method using the euclidean norm [Lenstra et al., 1982].

Branching on hyperplanes

There are different integer programming algorithms using basis reduction. The main idea behind those is to enumerate the parallel hyperplanes containing the regarded lattice in order to find out if there is a lattice point inside the regarded area. Therefore a representation of the lattice hyperplanes is searched for, ensuring that for a given dimension n only a polynomial number of hyperplanes has to be enumerated. One of these algorithms is the algorithm of Grötschel, Lovász and Schrijver [Aardal et al., 2002, 2005]. They proofed, that in polynomial time one can either find an integer point in an regarded polytope or a non-zero integral direction c, such that the width of the polytope in this direction is bounded by a constant only depending on the dimension n. They show, that by applying the shallow-cut ellipsoid method, cf. [Aardal et al., 2005], this theorem is also valid, if an ellipsoid instead of the polytope is regarded.

If the algorithm does not lead to a feasible integer solution, but only to an integral vector c, a "branching on hyperplanes"-procedure can be applied to find feasible integer solutions [Cook et al., 1993].

7.4.1 The LAMBDA-method

The problem of searching for integer points in an ellipsoid also occurs in Geodesy in the context of the development of Global Positioning Systems (GPS). It is introduced in this context in [Teunissen, 1993], the theoretical background is explained in detail in [Teunissen, 1997a,b,c,d] and the implementation aspects are described in [de Jonge and Tiberius, 1996]. Here, the considered problem is to find $x \in \mathbb{Z}^n$ that solves

$$\min_{x \in \mathbb{Z}^n} (x-p)^T Q^{-1} (x-p), \tag{7.40}$$

where Q^{-1} is a symmetric, positive definite real matrix and $p \in \mathbb{R}^n$. To solve this problem, first a reduction process is performed. Then a search algorithm on a hyper ellipsoid will be used to evaluate the solution of (7.40).
The main idea of the algorithm is to calculate lower and upper bounds of every variable in order to be able to perform an efficient integer point search in the ellipsoid. The quality of the bounds is increased by calculating directly only the bounds of one variable and then to calculate the other ones in dependence of the already calculated. As only the previously calculated values of the variable-bounds are known, a decorrelation step is necessary, so that only the already calculated bounds are needed to find the ones for the next variable.
For a better efficiency of the algorithm a reduction process is used. The LAMBDA-reduction is based on the ideas presented in [Lenstra, 1981] that also led to the LLL-algorithm. Here also a reduction process is used to find grid basis vectors in a way, that they are nearly orthogonal, as parallel as possible to the grid axes of the ellipsoid, short and sorted in a decreasing order. As the process is performed on the matrix defining the ellipsoid and not on some vectors defining a grid basis, always the grid of integer points is regarded and one starts with the canonical basis.
Performing the same transformation, that was used to reduce the grid, on the ellipsoid leads to a new ellipsoid, that it is more sphere-like, without changing its volume [Teunissen, 1993]. Figure 7.1 shows how this reduction process increases the quality of the bounds.

Reduction process

An ellipsoid E is *defined by a matrix* Q^{-1}, if and only if

$$E = \left\{ x \in \mathbb{R}^n \;\middle|\; (x-x^+)^T Q^{-1} (x-x^+) < \chi,\ \chi \in \mathbb{R} \right\}. \tag{7.41}$$

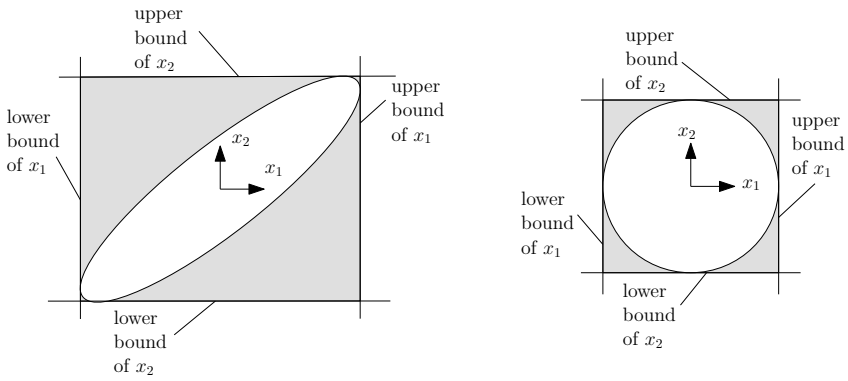


Figure 7.1: Two ellipsoids of the same volume with the best lower and upper bounds for the variables x_1 and x_2

The Cholesky-decomposition can be used to decompose Q^{-1} into $Q^{-1} = L^T DL$, with a unit lower triangular matrix L and a diagonal matrix [Plato, 2000]

$$D = \text{diag}(d_1, \dots d_n), d_i > 0, \ \forall i \in \{1, \dots n\}\,. \tag{7.42}$$

So, $Q = L^{-1}D^{-1}L^{T^{-1}}$.
Then, [Teunissen, 1995] suggests to search for a transformation matrix Z, such that

$$\tilde{E} \stackrel{def}{=} \left\{ z \in \mathbb{R}^n \ \mid \ (z - \hat{z})^T \tilde{Q}^{-1} (z - \hat{z}) < 1 \right\}, \tag{7.43}$$

with $\tilde{Q} \stackrel{def}{=} Z^T Q Z$ and $z \stackrel{def}{=} Z^T x$, $\hat{z} \stackrel{def}{=} Z^T p$ is an ellipsoid where the number of integer points is the same as in E, but with a different shape.
Z should be an integer approximation of L and unimodular, i.e.

$$|\det(Z)| = 1, \tag{7.44}$$

as unimodular matrices are volume-preserving, this means a transformation performed by Z changes the shape but not the volume of the regarded ellipsoid. Additionally, Z should be invertible and fullfill the following properties:

$$b = Za \in \mathbb{Z}^n \Leftrightarrow a \in \mathbb{Z}^n \ and \tag{7.45}$$

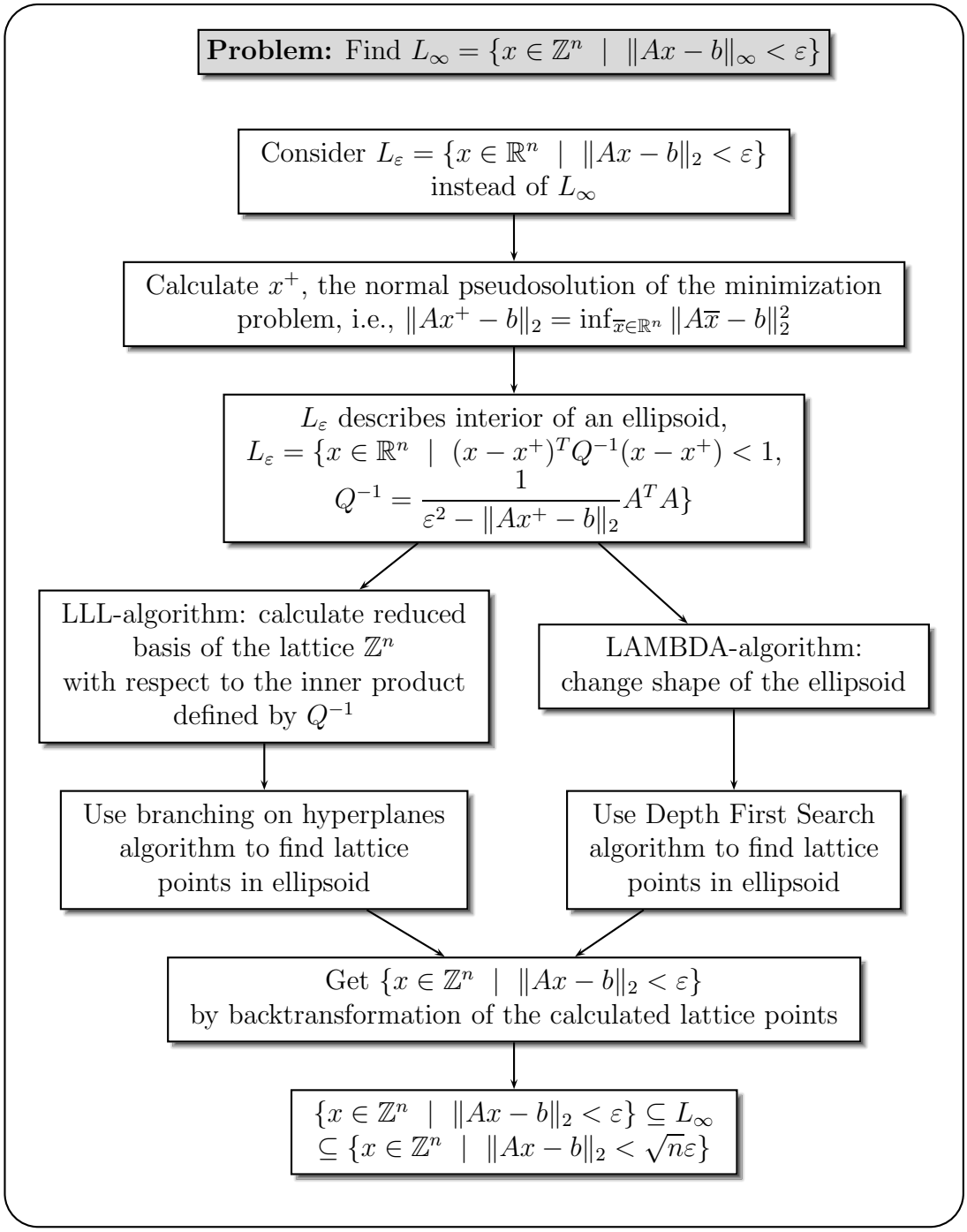
$$a = Z^{-1}b \in \mathbb{Z}^n \Leftrightarrow b \in \mathbb{Z}^n. \tag{7.46}$$

As Z is integer-preserving, cf. (7.45), the number of grid points inside the ellipsoid defined by $\tilde{Q}$ is the same as the number of grid points inside the ellipsoid defined by Q^{-1}.
By choosing a matrix Z approximating L, one tries to get a matrix $\tilde{Q}$ as diagonal as possible, i.e., to decorrelate the unknown parameters.

To search the ellipsoid for valid integer values, a kind of Depth First Search Algorithm can be used.
The upper and lower bounds defining valid intervals for every variable z_i can be calculated recursively. Starting with z_n the smallest integer value in the interval is taken as value for z_n, then go on and take the smallest integer value in the corresponding interval for z_{n-1} and so on. If there is no integer value in the interval for a z_i, return to z_{i-1} and choose the next integer value, which is the smallest, not already chosen one, in this interval and continue. Getting a value for z_1, an integer solution has been found. Now go on with z_1 and check, if there is another integer value in the interval for z_1, if there is one, it gives another integer solution. When there is none left, go back to z_2, take the next integer value and so on.

7.4.2 Flow chart

The main steps of the ways to determine the set $L_\infty = \{x \in \mathbb{Z}^n \mid \|Ax - b\|_\infty < \varepsilon\}$ using the previously described methods are summarized in the following flow chart.


Problem: Find $L_\infty = \{x \in \mathbb{Z}^n \mid \|Ax - b\|_\infty < \varepsilon\}$
Consider $L_\varepsilon = \{x \in \mathbb{R}^n \mid \|Ax - b\|_2 < \varepsilon\}$ instead of L_∞
Calculate x^+, the normal pseudosolution of the minimization problem, i.e., $\|Ax^+ - b\|_2 = \inf_{\overline{x} \in \mathbb{R}^n} \|A\overline{x} - b\|_2^2$
L_ε describes interior of an ellipsoid, $L_\varepsilon = \{x \in \mathbb{R}^n \mid (x - x^+)^T Q^{-1}(x - x^+) < 1, \; Q^{-1} = \frac{1}{\varepsilon^2 - \|Ax^+ - b\|_2} A^T A\}$
LLL-algorithm: calculate reduced basis of the lattice $\mathbb{Z}^n$ with respect to the inner product defined by Q^{-1}
LAMBDA-algorithm: change shape of the ellipsoid
Use branching on hyperplanes algorithm to find lattice points in ellipsoid
Use Depth First Search algorithm to find lattice points in ellipsoid
Get $\{x \in \mathbb{Z}^n \mid \|Ax - b\|_2 < \varepsilon\}$ by backtransformation of the calculated lattice points
$\{x \in \mathbb{Z}^n \mid \|Ax - b\|_2 < \varepsilon\} \subseteq L_\infty \subseteq \{x \in \mathbb{Z}^n \mid \|Ax - b\|_2 < \sqrt{n}\varepsilon\}$

Summary and Outlook

> We can only see a short distance ahead,
> but we can see plenty there
> that needs to be done.
>
> Alan Turing

The phase problem is the major problem in the field of X-ray crystallography. In the context of direct methods, that use mathematical techniques to compute an electron density map from the diffraction data without any further experiments, binary integer programming models for solving the phase problem have been developed.
Based on descriptions of topological properties of 2-dimensional binary pictures known from the field of discrete tomography, these models have been extended for the 3-dimensional case. As the formulations are in general not sufficient to describe the more complex properties of the shape of proteins, binary integer programs have been derived for describing different additional topological properties. In general, the binary integer program for solving the phase problem, leads to a set of different optimal solutions. The additional constraints increase the quality of the solution set.
The main property considered is one restricting the number of components in the resulting solution. Using graph theoretical methods and a separation algorithm, a model to describe this property has been found and implemented. Computational results have been presented and evaluated. It has been shown, that the added topological constraints increase significantly the quality of the solution set.
In the last chapter, a method to find the solutions all at once based on singular value decomposition and methods to find integer points in ellipsoids has been developed. In further work, the efficiency of this method for the phase problem should be evaluated and the method could be implemented and tested.
In order to further increase the solutions' quality, more additional constraints could be formulated and added.
If the running time of the solving algorithm could be decreased, a refinement of

the model would be possible. Bigger grids could be considered showing more details of the reconstructed protein. More phase values than just four ones could be introduced. A restriction of the electron density distribution to a finite number of states instead of regarding just the two binary ones would be a possible extension. So, based on the promising results presented here, lots of further work extending and refining the developed approaches is possible.

Zusammenfassung

Röntgenkristallographie ist derzeit die Standardmethode zur Ermittlung der dreidimensionalen Struktur biologischer Makromoleküle, wie z. B. von Proteinen, und liefert damit eine wichtige Basis der Strukturbiologie sowie der modernen Biotechnologie.
Aus Röntgenexperimenten erhält man Beugungsmuster, aus welchen dann die Struktur des zu untersuchenden Kristalls berechnet werden soll. Diese wird durch die zugehörige Elektronendichteverteilung beschrieben. Allerdings liefert das Beugungsmuster nur die Beträge der komplexen Fourierkoeffizienten der Elektronendichte, nicht die zugehörigen Phasenwerte. Das Problem, diese Phasenwerte zu ermitteln, ist das Phasenproblem in der Röntgenkristallographie.
Die Informationen, welche aus dem Röntgenexperiment gewonnen werden können, sind nicht ausreichend um dieses Phasenproblem zu lösen. Daher erhält man nicht nur eine, sondern eine Menge zulässiger Lösungen. Zusätzliche Informationen über die Elektronendichteverteilung können dann hinzugezogen werden um die Qualität dieser Lösungen zu verbessern.
In dieser Arbeit wird ein ganzzahliger linearer Optimierungsansatz zur Lösung des Phasenproblems entwickelt, in dem verschiedene topologische Eigenschaften von Proteinen modelliert und als zusätzliche Informationen verwendet werden. Die wichtigste Eigenschaft, die so modelliert und für die Problemlösung hinzugezogen wird, ist die Zusammenhangseigenschaft von Proteinen. Diese sichert, dass die berechnete Struktur nicht aus mehr als einer gegebenen Anzahl zusammenhängender Komponenten besteht. Bei der Modellierung dieser Zusammenhangseigenschaft werden graphentheoretische Methoden sowie ein Separationsalgorithmus genutzt.
Der Modellierungsansatz wurde implementiert und mit Daten von echten Proteinen getestet. Die Testergebnisse zeigen, dass die Qualität der Lösungen des Phasenproblems durch die Hinzunahme der topologischen Eigenschaften deutlich verbessert wird.

List of Figures

Bibliography

K. Aardal, R. Weismantel, and L. Wolsey. Non-Standard Approaches to Integer Programming. *Discrete Applied Mathematics*, 123:5–74, November 2002.

K. Aardal, G.L.Nemhauser, and R. Weismantel. *Handbooks in Operations Research and Management Science: Discrete Optimization*, volume 12. Elsevier, 2005.

T. Achterberg. SCIP - a framework to integrate Constraint and Mixed Integer Programming. Technical Report 04-19, Konrad-Zuse-Zentrum für Informationstechnik Berlin, June 2004.

T. Achterberg. *Constraint Integer Programming*. Ph.d. thesis, TU Berlin, July 2007. URL `http://opus.kobv.de/tuberlin/volltexte/2007/1611/`.

T. Achterberg, T. Berthold, T. Koch, and K. Wolter. Constraint Integer Programming: a New Approach to Integrate CP and MIP. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 5015, pages 6–20. CPAIOR 2008, Springer Berlin / Heidelberg, 2008.

K. R. Apt. *Principles of Constraint Programming*. Camebridge University Press, 2003.

M. I. Aroyo, U. Müller, and H. Wondratschek. International Tables for Crystallography. In *International Tables for Crystallography Volume A1: Symmetry relations between space groups*, volume A1. Springer, 2006. URL `http://www.springerlink.com/content/k1211127p1301353/`.

F. Barahona and A. R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36(2):157–173, June 1986.

G. Barrow. *Physikalische Chemie*, volume 2. Bohmann-Verlag Wien, 1979.

H. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide Protein Data Bank. *Nature Structural & Molecular Biology*, 10(12):980, December 2003.

D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, Massachusetts, 1997.

A. Bockmayr and J. Hooker. Constraint Programming. In R. W. K. Aardal, G. Nemhauser, editor, *Discrete Optimization. Handbooks in Operations Research and Management Science.*, 12, chapter 10, pages 559–600. Elsevier, 2005.

J. Bode. Verfeinerung eines gegebenen BIP-Modells für das Phasenproblem. Bachelor thesis, Freie Universität Berlin, Fachbereich Mathematik und Informatik, 2010.

W. Borchardt-Ott. *Kristallographie. Eine Einführung für Naturwissenschaftler*. Springer-Verlag, 2002.

W. L. Briggs and V. E. Henson. *The DFT: an owners manual for the discrete Fourier transform*. Society for Industrial and Applied Mathematics, 1995.

C. Brinkmann and A. Bockmayr. A Constraint-based Approach to the Phase Problem in X-ray Crystallography. In *WCB08 - Workshop on Constraint Based Methods for Bioinformatics associated to CPAIOR 2008 in Paris*, 2008.

S. Chopra and M. Rao. Facets of the k-partition polytope. *Discrete Applied Mathematics*, 61:27–48, 1995.

S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59(1-3):87–115, March 1993.

W. Cook, T. Rutherford, H. E. Scarf, and D. Shallcross. An Implementation of the Generalized Basis Reduction Algorithm for Integer Programming. *ORSA Journal on Computing*, 5(2):206–212, 1993.

CPLEX. http://www.ilog.com/products/cplex/, 2010. URL `www.cplex.com`.

P. de Jonge and C. Tiberius. The Lambda method for integer ambiguity estimation: implementation aspects. Technical Report 12, Delft Geodetic Computing Centre, August 1996.

W. L. DeLano. The PyMOL Molecular Graphics System. http://www.pymol.org, 2002.

M. Deza, M. Grötschel, and M. Laurent. Clique-Web Facets for Multicut Polytopes. *Mathematics of Operations Research*, 17(4):981–1000, November 1992.

M. T. Dove. *Structure and dynamics: an atomic view of materials*. Oxford Master Series in Condensed Matter Physics. Oxford University Press, 2003.

J. Drenth. *Principles of protein X-ray crystallography*. Springer-Verlag, 1994.

C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. Wolsey. The Node Capacitated Graph Partitioning Problem: A Computational Study. *Mathematical Programming*, 81:229–256, 1998.

M. Fourier. *Théorie analytique de la chaleur*. Firmin Didot, Père et fils, libraires pour les mathématiques, l'architecture hydrauliq et la marine, 1822.

M. G. Friedel. Sur les symétries cristallines que peut révéler la diffraction des rayons Röntgen. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 157:1533–1536, 1913.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

C. Gasquet and P. Witomski. *Fourier Analysis and Applications*. Springer-Verlag, 1999.

C. Giacovazzo. *Direct Phasing in Crystallograhpy*. Oxford University Press, 1998.

C. Giacovazzo, H. Monaco, G. Artioli, D. Viterbo, G. Ferraris, G. Gilli, G. Zanotti, and M. Catti. *Fundamentals of Crystallograhpy*. Oxford University Press, 2002.

O. Goldschmidt and D. S. Hochbaum. A polynomial algorithm for the k-cut Problem for fixed k. *Mathematics of Operations Research*, 19(1):24–37, February 1994.

B. I. Golubov. Multiple fourier series and integrals. *Journal of Mathematical Sciences*, 24(6):639–673, 1984.

R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.

M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1988.

G. Hämmerlin and K.-H. Hoffmann. *Numerische Mathematik*. Springer-Verlag, 1994.

H. A. Hauptman. Direct Methods and Anomalous Dispersion. *Nobel Lectures, Chemistry 1981-1990*, 1992.

C. Heldt and A. Bockmayr. Geometric Constraints for the Phase Problem in X-Ray Crystallography. *WCB10 - Workshop on Constraint Based Methods for Bioinformatics 2010 in Edinburgh*, 2010.

F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. Oldenbourg, 2001.

IUCR. IUCR Online Dictionary of Crystallography, 2006. URL `http://-reference.iucr.org/dictionary/Main_Page`.

J. Karle. Recovering Phase Information from Intensity Data. *Nobel Lectures, Chemistry 1981-1990*, 1992.

J. Karle and H. Hauptman. A theory of phase determination for the four types of non-centrosymmetric space groups $1P222$, $2P22$, $3P_12$, $3P_22$. *Acta Crystallographica*, 9:635–651, 1956.

J. C. Kendrew. Myoglobin and the Structure of Proteins. In *Nobel Lectures, Chemistry 1942-1962*, pages 676–698. Elsevier Publishing Company, 1964.

B. Korte and J. Vygen. *Combinatorial Optimization. Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer, 2000.

J. B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In *Proceedings of the American Mathematical Society*, volume 7, pages 48–50. American Mathematical Society, Feb. 1956.

A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

A. Lenstra, H. Lenstra, and L. Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, 261:515–534, 1982.

H. Lenstra. Integer Programming with a Fixed Number of Variables. Technical Report 81-03, University of Amsterdam, 1981.

V. Lunin. One dimensional X-ray analysis and BLP problem. Unpublished notes, June 2003.

V. Lunin, N. Lunina, A. Podjarny, A. Bockmayr, and A. Urzhumtsev. Ab initio phasing starting from low resolution. *Zeitschrift für Kristallographie*, 217:668–685, 2002a.

V. Y. Lunin and N. L. Lunina. The map correlation coefficient for optimally superposed maps. *Acta Cryst. A*, 52:365–368, 1996.

V. Y. Lunin, A. Urzhumtsev, and N. Lunina. Connectivity properties of high-density regions and *ab initio* phasing at low resolution. *Acta Cryst. A*, 56: 375–382, 2000.

V. Y. Lunin, A. Urzhumtsev, and A. Bockmayr. Direct phasing by binary integer programming. *Acta Crystallogr A*, 58(Pt 3):283–291, May 2002b.

N. Lunina, V. Y. Lunin, and A. Urzhumtsev. Connectivity-based ab initio phasing: from low resolution to a secondary structure. *Acta Crystallogr D Biol Crystallogr*, 59(Pt 10):1702–1715, Oct 2003.

T. L. Magnanti and L. A. Wolsey. Optimal Trees. In G.L.Nemhauser and A. R. Kan, editors, *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, chapter 9, pages 503–616. Massachusetts Institute of Technology, Operations Research Center, 1995.

R. Mannhold, H. Kubinyi, and G. Folkers, editors. *Protein Crystallography in Drug Discovery (Methods and Principles in Medicinal Chemistry)*. Wiley-VCH, 2003.

MATLAB, 2010. URL `http://www.mathworks.com`.

J. Müller, N. Lunina, A. Urzhumtsev, E. Weckert, U. Heinemann, and V. Lunin. Low-resolution ab initio phasing of Sarcocystis muris lectin SML-2. *Acta Cryst. D*, 62:1–8, 2006.

G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1988.

A. L. Patterson. A Direct Method for the Determination of the Components of Interatomic Distances in Crystals. *Zeitschrift für Kristallographie*, 76:177–201, 1931.

A. L. Patterson. A Fourier Series Method for the Determination of the Components of Interatomic Distances in Crystals. *Physical Review*, 46:372–376, 1934.

A. L. Patterson. Methods in Crystal Analysis. *Zeitschrift für Kristallographie*, 90: 517–543, 1935.

M. F. Perutz. X-ray Analysis of Haemoglobin. In *Nobel Lectures, Chemistry 1942-1962*, pages 653–673. Elsevier Publishing Company, Amsterdam, 1964.

R. Plato. *Numerische Mathematik kompakt*. Vieweg, 2000.

Protein Data Bank. Protein Data Bank, 2010. URL `http://www.rcsb.org`.

Protein Data Bank Contents Guide. Atomic Coordinate Entry Format Description. Version 3.20, 2008. URL `ftp://ftp.wwpdb.org/pub/pdb/doc/format_descriptions/Format_v32_A4.pdf`.

G. Rhodes. *Crystallograhpy Made Crystal Clear*. Academic Press, Inc., 1993.

D. Sayre. The Calculation of Structure Factors by Fourier Summation. *Acta Cryst.*, 4:362–367, 1951.

D. Sayre. The Squaring Method: a New Method for Phase Determination. *Acta Cryst.*, 5:60–65, 1952.

A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, New York, 1989.

A. Schrijver. Combinatorial Optimization. In *Polyhedra and Efficiency*, volume A. Springer, 2003.

D. Sherwood. *Crystals, X-rays and Proteins*. Longman, 1976.

M. M. Sørensen. Facet-defining inequalities for the simple graph partitioning polytope. *Discrete Optimization*, 4(1):221–231, June 2007.

H. Stöcker. *Taschenbuch der Physik*. Verlag Harry Deutsch, 1994.

L. F. TenEyck. Efficient Structure-Factor Calculation for Large Molecules by the Fast Fourier Transform. *Acta Cryst.*, A33:486–492, 1977.

P. Teunissen. Least-Squares Estimation of the Integer GPS ambiguities. *LGR (Delft Geodetic Computing Centre) Series*, 6, 1993.

P. Teunissen. The invertible GPS ambiguity transformations. *manuscripta geodaetica*, 20:489–497, 1995.

P. J. G. Teunissen. A canonical theory for short GPS baselines. Part I: The baseline precision. *Journal of Geodesy*, 71:320–336, 1997a.

P. J. G. Teunissen. A canonical theory for short GPS baselines. Part II: the ambiguity precision and correlation. *Journal of Geodesy*, 71:389–401, 1997b.

P. J. G. Teunissen. A canonical theory for short GPS baselines. Part III: the geometry of the ambiguity search space. *Journal of Geodesy*, 71:486–501, 1997c.

P. J. G. Teunissen. A canonical theory for short GPS baselines. Part IV: precision versus reliability. *Journal of Geodesy*, 71:513–525, 1997d.

G. P. Tolstov. *Fourier Series*. Dover Publications, 1962.

C. Überhuber. *Computer-Numerik 2*. Springer, 1995.

A. Vretblad. *Fourier Analysis and Its Applications*. Graduate Texts in Mathematics. Springer, 2003.

J. S. Walker. *Fast Fourier Transforms*. CRC Press, 1991.

J. Waser. Symmetry relations between structure factors. *Acta Cryst.*, 8:595, 1955.

G. J. Woeginger. The reconstruction of polyominoes from their orthogonal projections. *Information Processing Letters*, 77:225–229, 2001.

R. Wunderling. *Paralleler und Objektorientierter Simplex*. PhD thesis, TU Berlin, 1996.

XPLOR. File Format, 1999. URL `http://www.scripps.edu/rc/softwaredocs/-msi/xplor981/formats.html`.

W. H. Zachariasen. *Theory of X-ray Diffraction in Crystals*. Wiley, New York, 1945.