

6 Ergebnisse

6.1 Testmodelle und Einstellungen

Das Testset ist in 10 leichte und 10 schwere Modelle aufgeteilt und beinhaltet gemischte Binär-Modelle (MB), gemischt-ganzzahlige Modelle (M), reine Binärmodelle (B) und reine Integermodelle (I). Die Modelle sind Bestandteil der [MIPLIB3], [MIPLIB2003] und [MILPlib].

Die ausgewählten Modelle des leichten Testsets (Tab. 6-1) können mit mindestens einer der Strategien innerhalb einer Stunde gelöst werden.

Tab. 6-1: Testset leichte Modelle

Modelle	Art	Zeilen	Spalten	NZ	0-1	allg. Int.	kont.	IP-Lösung
air04	B	823	8904	72965	8904	-	-	56137
air05	B	426	7195	52121	7195	-	-	26374
bienst1	MB	576	505	2184	28	-	477	46,75
cap6000	B	2176	6000	48243	6000	-	-	-241377
mas76	MB	12	151	1640	150	-	1	40005,1
mod011	MB	4480	10958	22254	96	-	10862	-54558535
neos20	M	2446	1165	7428	937	30	198	-434
pk1	MB	45	86	915	55	-	31	11
qiu	MB	1192	840	3432	48	-	792	-132,8731
swath1	MB	885	6805	34966	2406	-	4399	379,07

Die schweren Modelle werden innerhalb des Zeitlimits von 60 Minuten nicht gelöst. Die Modelle, bei denen die Optimalität bis jetzt von keiner gängigen Optimierungssoftware bewiesen wurde, werden in der Tab. 6-2 durch einen Stern (*) in der Spalte der IP-Lösung gekennzeichnet.

Tab. 6-2: Testset schwere Modelle

Modelle	Art	Zeilen	Spalten	NZ	0-1	allg. Int.	kont.	IP-Lösung
alcls1	MB	3312	3648	10178	192	-	3456	11557,09*
arki001	MB	1048	1388	20439	442	96	850	7580813,0459*
dano3mip	MB	3202	13873	79655	552	-	13321	728,1111*
liu	MB	2178	1156	10626	1089	-	67	1448*
markshare2	MB	7	74	434	60	-	14	1
mkc	MB	3411	5325	17038	5323	-	2	-563,85
nsrand-ixx	MB	735	6621	223261	6620	-	1	51200
roll3000	M	2295	1166	29386	246	482	428	12890
seymour	B	4944	1372	33549	1372	-	-	423
swath	MB	885	6805	34966	6724	-	81	497,603*

Jedes Modell wird maximal 60 Minuten und maximal $1 \cdot 10^7$ Knoten gelöst. Der maximale Speicherplatz für das Nodefile beträgt 1000 MB. Die wichtigsten Parametereinstellungen der verwendeten MOPS-Version 8.13 sind im Anhang C nachzulesen.

Die Benchmarks wurden mit einem Intel Pentium 4 Prozessor mit 3,0 GHz ermittelt und sind für jedes Modell im Anhang D aufgelistet.

6.2 Bewertungsmaßstab

Die Bewertung der Testsets wird u.a. mit Hilfe eines Rankings durchgeführt. Dabei werden die numerischen Ergebnisse der einzelnen Modelle für alle n Strategien verglichen und je nach Güte der Ergebnisse aufsteigend gewichtet, d.h. die Strategie, die ein Modell am besten löst, bekommt eine Eins und die Strategie, die das Modell am schlechtesten löst, bekommt n zugewiesen. Bei der Gewichtung wird unterschieden, ob ein Modell optimal gelöst oder ob ein Abbruchkriterium erreicht wurde.

Bei einer optimalen Lösung ist eine Strategie besser als eine andere, wenn die Lösungszeit schneller ist.

Wurde die Optimalität nicht bewiesen, dann ist eine Strategie besser als eine andere, wenn eine bessere Integer-Lösung gefunden wurde bzw. bei keiner gefundenen Integer-Lösung die Lower Bound besser ist. Besteht eine Gleichheit der Integer-Lösungen, dann ist die Strategie mit dem kleineren globalen Gap besser.

In einer Patt-Situation bekommen die Modelle das gleiche Gewicht zugewiesen. Beim Ranking wird für jede Strategie das kleinste (min), das durchschnittliche (avg) und das größte Gewicht (max) ermittelt. Werden allerdings nur zwei Strategien miteinander verglichen, dann wird nur das durchschnittliche Gewicht angezeigt.

Da die leichten Modelle in der Regel bis zur Optimalität gelöst werden und die schweren Modelle an das Ressourcenlimit stoßen, ist die weitere Bewertung der beiden Testreihen unterschiedlich.

Leichte Modelle

Für die leichten Modelle wird zusätzlich zu dem Ranking eine statistische Auswertung vorgenommen. Dazu werden die benötigten Ressourcen wie Knoten, Iteration und Zeit über alle optimal gelösten Modelle der Testreihe aufsummiert. Zusätzlich wird der Median von der Ressource Zeit gebildet. Die Summe der Zeit und der Median beziehen sich auf Minuten. Die Spalte „nicht opt.“ gibt die Anzahl der Modelle an, die mit der betreffenden Strategie nicht optimal gelöst werden konnten.

Neben den statistischen Daten wird ein Performance Profile [DoMo01] erstellt. Das Performance Profile analysiert grafisch die Ergebnisse einer Testreihe und wird an dieser Stelle kurz skizziert.

Ein Testset besteht aus n_p Modellen und wird mit n_s Strategien gelöst. Für jedes Modell p kann die Zeit $t_{p,s}$ bestimmt werden, die benötigt wird, wenn das Problem p mit der Strategie s optimal gelöst wird. Nun wird die Lösungszeit des Problems p mit der Strategie s mit der besten Zeit, die eine Strategie für das Problem p benötigt hat, verglichen. Dazu wird das Performance Verhältnis $r_{p,s}$ gebildet. Es ist wie folgt definiert:

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}} \quad (6-1)$$

Löst die Strategie s_1 das Modell p am schnellsten, dann ist $r_{p,s_1}=1$ und ist $r_{p,s_1}>1$, dann benötigt die Strategie s_1 r_{p,s_1} -mal mehr Zeit als die schnellste Strategie. Ein Vorteil dieses Verhältnisses ist, dass mögliche große Zeitdifferenzen bei Problemen mit einer verhältnismäßigen langen Laufzeit anders als bei der statistischen Analyse geglättet werden. Wird ein Modell im Ressourcenlimit nicht optimal gelöst, dann ist $r_{p,s}=r_M$, wobei r_M eine benutzerdefinierte Zahl ist und $r_M>r_{p,s}$ für alle optimal gelösten Modelle gilt. Das Performance Profile $\rho_s(\tau)$ gibt die Wahrscheinlichkeit an, wie viele Modelle p mit einer Strategie s innerhalb der τ -fachen Zeit der schnellsten Strategie gelöst werden. Die Verteilungsfunktion des Performance Verhältnisses wird wie folgt definiert:

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P : r_{p,s} \leq \tau\} \text{ ist und } 0 \leq \rho_s(\tau) \leq 1 \quad (6-2)$$

Eine Strategie s_1 ist besser als die Strategie s_2 , wenn für ein gleiches τ $\rho_{s_1}(\tau)>\rho_{s_2}(\tau)$ ist. Der Wert $\rho_s(1)$ gibt an, wie oft eine Strategie s in kürzester Zeit ein Problem des Testsets gelöst hat, und der Wert $1-\rho_s(\tau)$ ist der Anteil an Problemen im Testset, die die Strategie s in einem Faktor τ der optimalsten Strategie nicht lösen kann. Diese beste bzw. optimalste Strategie ist immer eine fiktive Strategie, die jedes Problem am schnellsten löst, d.h. wo $\rho_s(1)=1$ ist.

Neben der grafischen Darstellung des Performance Profiles werden zur Analyse in einer Tabelle die Werte $\rho_s(1)$, $\rho_s(r_M)$ und r_{\max} angeben. Für alle optimal gelösten Modelle ist $\rho_s(r_M)=1$ und r_{\max} gibt das größte Performance-Verhältnis der Strategie s an. Wurden bei einer Strategie nicht alle Modelle bis zur Optimalität gelöst, dann gibt $\rho_s(r_M)$ den Prozentsatz der optimal gelösten Modelle an und $r_{\max}=r_M$.

Damit alle Daten in der Grafik erfasst werden können, muss $r_{p,s}$ gegebenenfalls für alle Probleme p und Strategien s auf der Basis zwei logarithmiert werden.

$$\tau \mapsto \frac{1}{n_p} \text{size}\{p \in P : \log_2(r_{p,s}) \leq \tau\} \quad (6-3)$$

Durch die Logarithmierung des Performance Verhältnisses geht die Kardinalskalierung verloren. Ist z.B. in der Graphik abzulesen, dass eine Strategie n -mal mehr Zeit benötigt, benötigt sie tatsächlich 2^n -mal mehr Zeit. Die Werte in der dazugehörigen Tabelle sind nicht logarithmiert.

In einigen Fällen wird wegen der Übersichtlichkeit nur der wichtigste Ausschnitt eines Performance Profiles gezeigt. Anhand der Daten in der Tabelle ist der weitere Verlauf des Profiles absehbar.

Schwere Modelle

Bei den schweren Modellen wird zusätzlich zu dem Ranking aufgelistet, wie oft bei allen Modellen insgesamt keine Integer-Lösung gefunden wurde.

6.3 Ergebnisse und Auswertung

Die numerischen Resultate werden für die Branching-Strategien und die Heuristiken ermittelt. Für alle Strategien wird die Partitionierung $xparnd=-2$ gesetzt.

Bei vier der 20 Modelle führte die Benutzung einer Integer-Tabelle ($xnoitb=0$) zu einer schnelleren Lösung des Modells bzw. zu einer besseren Integer-Lösung. Der Verzicht auf eine Integer-Tabelle führt nur bei zwei Modellen zu einer geringfügig langsameren Lösungszeit. Demzufolge wird bei der Ermittlung der numerischen Resultate eine Integer-Tabelle verwendet. Des Weiteren erfolgt eine erweiterte Bound Reduction ($xbndrd=2$), die auch bei einigen Modellen zu einem schnelleren Lösen bzw. besseren Integer-Lösung führt (siehe Anhang D).

6.3.1 Branching-Strategien

In diesem Abschnitt werden die in Kapitel 4.2 beschriebenen Branching-Strategien untersucht. Ohne die Verwendung einer Heuristik ($xheutp=0$) werden die Variablen- bzw. Knotenauswahlstrategien anhand der Testsets getestet.

Wahl der Branchingvariablen

Die numerischen Resultate für die Wahl der Branchingvariablen werden mit der Knotenauswahlstrategie $xnodse=3$ ermittelt. Folgenden Strategien werden untersucht:

- *B3*: Wahl nach den Kosten ($xbrheu=3$)
- *B5*: Pseudo Kosten mit größter gewichteter Summe ($xbrheu=5$). Die Pseudo Kosten werden mit den Kosten initialisiert ($xpcost=1$).
- *B5a*: Pseudo Kosten mit größter gewichteter Summe ($xbrheu=5$). Die Pseudo Kosten werden durch das teilweise Lösen des relaxierten LPs initialisiert ($xpcost=2$).
- *B10*: Strong Branching ($xbrheu=10$)

Die weiteren vorgestellten Strategien Clique-Branching ($xbrheu=7$) und Logical Branching ($xbrheu=8$) können nur auf drei bzw. fünf Modelle angewandt werden (siehe Anhang B). Ein Vergleich mit der Basisstrategie $xbrheu=3$ ergab, dass das Clique-Branching keine Verbesserung erzielt hat. Das Logical Branching führte zu einer Verbesserung bei den drei schweren Modellen, aber zu einer Verschlechterung der zwei leichten Modelle.

Tab. 6-3: Numerische Resultate – Wahl der Branchingvariablen

	leicht					schwer		
	Knoten	Iteration	Zeit (min) Summe Median		nicht opt.	Ranking (min/avg/max)	keine	Ranking (min/avg/max)
B3	2624656	16750352	43,69	5,41	3	(1/2,6/4)	1	(1/2,6/4)
B5	2364045	12470234	40,37	5,52	0	(1/2,0/3)	0	(1/1,9/3)
B5a	2120042	12324253	37,13	5,83	1	(1/2,1/4)	0	(1/2,3/4)
B10	1299884	47981474	63,86	10,31	2	(1/3,3/4)	1	(1/2,8/4)

Leichte Modelle

In dem leichten Testset konnten die Strategien B3, B5a und B10 nicht alle Modelle optimal lösen. Deshalb wurden in die Summenbildung die Modelle cap6000 (B3), neos20 (B3,

B5a, B10), qiu (B3) und swath1 (B10) nicht einbezogen. Die restlichen Modelle löst die Strategie B5a am schnellsten. Die Strategie B10 braucht die meisten Ressourcen und weist auch das schlechteste durchschnittliche Ranking auf. Laut Ranking ist die zweitschnellste Strategie B5 am besten, welche kein Modell am schlechtesten löst. In dem Ausschnitt des Performance Profiles in Abb. 6-1 dominiert die Strategie B5a die restlichen Strategien. Diese unterschiedliche Bewertungsreihenfolge von B5 und B5a zwischen statistischer Bewertung und Performance Profile liegt in den geringen zeitlichen Differenzen zwischen B5 und B5a (air04, air05), die bei dem Performance Profile kaum gewertet werden.

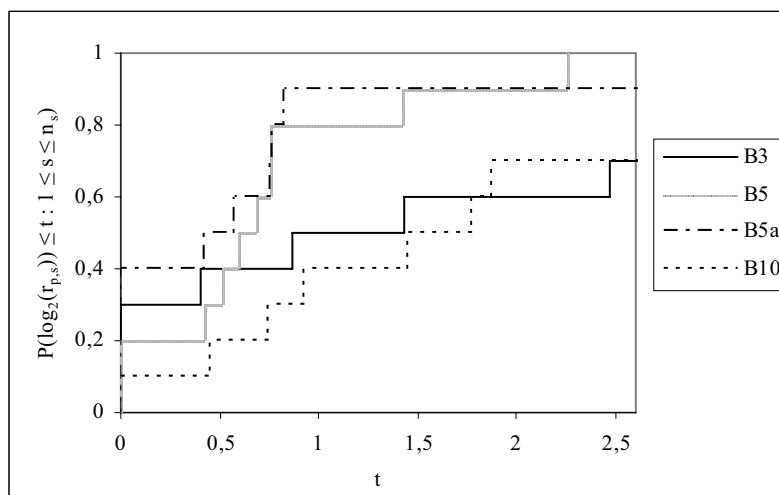


Abb. 6-1: Ausschnitt Performance Profil – Wahl der Branchingvariablen

Tab. 6-4: Werte Branchingvariable

	$\rho(1)$	$\rho(r_M)$	r_{\max}
B3	0,3	0,7	r_M
B5	0,2	1	4,77
B5a	0,4	0,9	r_M
B10	0,1	0,8	r_M

Schwere Modelle

Die Strategie B3 findet bei dem Modell liu und die Strategie B10 bei dem Modell arki001 im Zeitlimit keine Integer-Lösung. Die Branching-Strategie B5 schneidet bei den schweren Modellen deutlich am besten ab und die Strategie B10 am schlechtesten.

Fazit

Bei beiden Testsets schneidet nach dem Ranking die Strategie B5 am besten ab. Die Strategie B5a löst einige Modelle besonders im leichten Testset besser als die Strategie B5, hat aber auch ein paar Ausreißer (z.B. neos20, a1c1s1, dano3mip). In beiden Testsets ist die Strategie B10 nach dem durchschnittlichen Ranking und dem Performance Profile mit Abstand am schlechtesten. Allerdings hat diese Strategie bei den Modellen cap6000 und swath den größten Erfolg.

Knotenauswahlstrategie

Die Knotenauswahlstrategien entscheiden darüber, welcher aktive Knoten aus der Knotentabelle als nächstes zum Branching gewählt wird. Von den in Kapitel 4.2.2 beschriebenen Strategien werden die folgenden untersucht:

- *N3*: Best-Projektion ($xnodse=3$)
- *N3a*: Best-Projektion ($xnodse=3$) mit jedem 100sten Knoten Best-First-Strategie ($xnodbf=1$)

- *N5*: Estimation basierend auf den Pseudo Kosten ($xnodse=5$)
- *N6*: Geringste Unzulässigkeit bis Integer-Lösung gefunden wird, dann Estimation basierend auf den Pseudo Kosten und, wenn der Gap kleiner ist als 5%, prozentualer Fehler ($xnodse=6$)
- *N7*: Geringste Unzulässigkeit bis Integer-Lösung gefunden wird, dann Estimation basierend auf den Pseudo Kosten ($xnodse=7$)
- *NL0*: Mixed Lifo mit $xlifo=0$ und $xnodse=-3$
- *NL2*: Mixed Lifo mit $xlifo=1$ und $xnodse=-3$
- *NL3*: Mixed Lifo mit $xlifo=2$ und $xnodse=-3$

Die numerischen Resultate werden mit der Branchingheuristik $xbrheu=5$ ermittelt. Zur besseren Übersicht werden zunächst alle Nicht-LIFO-Strategien miteinander verglichen, und danach folgt ein Vergleich von der Strategie N3 mit den LIFO-Strategien.

Tab. 6-5: Numerische Resultate – Knotenauswahlstrategie - Nicht-LIFO

	leicht					schwer		
	Knoten	Iteration	Zeit (min) Summe Median		nicht opt.	Ranking (min/avg/max)	keine	Ranking (min/avg/max)
N3	2516072	18968488	96,76	7,51	0	(1/2,5/5)	0	(1/2,6/5)
N3a	2982335	19937485	87,73	8,78	0	(1/2,6/5)	0	(1/2,4/5)
N5	3482052	22074467	91,08	7,34	1	(1/2,9/5)	1	(1/2,9/5)
N6	3212368	24764338	147,29	13,25	0	(3/3,9/5)	0	(1/3,0/5)
N7	2624300	21887234	132,00	9,13	0	(1/2,9/5)	0	(1/2,7/4)

Tab. 6-6: Numerische Resultate – Knotenauswahlstrategien - LIFO

	leicht					schwer		
	Knoten	Iteration	Zeit (min) Summe Median		nicht opt.	Ranking (min/avg/max)	keine	Ranking (min/avg/max)
N3	2516072	18968488	91,39	6,6	0	(1/1,6/4)	0	(1/2,2/4)
NL0	2978730	18057793	74,39	6,15	0	(1/2,2/3)	0	(1/2,6/4)
NL1	3216099	19849252	87,29	6,66	0	(1/2,5/4)	0	(1/2,4/4)
NL2	3144573	22773424	117,66	11,6	1	(2/3,6/4)	0	(1/2,5/4)

Leichte Modelle

Die getesteten Knotenauswahl-Strategien N5 und NL2 haben das Modell swath1 nicht optimal gelöst. Demzufolge wird dieses Modell aus der Ressourcenberechnung ausgegliedert. Bei den Nicht-LIFO-Strategien schneidet die Strategie N3a beim Ressourcenverbrauch am besten ab und liegt im durchschnittlichen Ranking nur knapp hinter der Strategie N3. Die meisten Ressourcen benötigt die am schlechtesten bewertete Strategie N6. Diese dreiphasige Strategie unterscheidet sich im Vergleich zur zweiphasigen Strategie N7 nur durch die zusätzliche dritte Phase „percentage error“, die bei diesem Testset zu einer Verschlechterung der Ergebnisse führt. Die zusätzliche zweite Phase bei der Strategie N7 im Vergleich zu N5 führt zu einem erhöhten Ressourcenverbrauch bei gleich bleibendem Ranking.

Bei dem Vergleich der Mixed-LIFO-Strategien mit der Ausgangsstrategie N3 schneidet diese Nicht-LIFO-Strategie nach dem durchschnittlichen Ranking deutlich am besten ab.

Allerdings benötigt N3 wesentlich mehr Zeit als NL0 und NL1, was auf das Modell cap6000 zurückzuführen ist. Die LIFO-Strategie NL2 schneidet sowohl beim Ressourcenverbrauch also auch beim Ranking am schlechtesten ab.

Der Ausschnitt des Performance Profile in Abb. 6-2 zeigt den Vergleich der Strategien N3 und N3a mit den Pseudo Kosten Strategien N5-N7. Bei diesem Performance Profiles ist keine Strategie klar dominierend. Die vom Ranking am schlechtesten bewertete Strategie N6 löst kein Modell am schnellsten und ist auch in dem Performance Profile den anderen Strategien unterlegen. Die Strategie N3a benötigt den geringsten Faktor an Zeit im Vergleich zur optimalen Strategie, um alle Modelle optimal zu lösen.

Bei dem Vergleich der LIFO-Strategien in dem Ausschnitt des Performance Profiles (Abb. 6-3) hat die Strategie N3 den höchsten Ausgangswert, d.h. sie löst 70% der Modelle am schnellsten. Bis auf das Modell cap6000, bei dem die Strategie 3,58mal mehr Zeit benötigt als die optimale Strategie, ist N3 den LIFO-Strategien überlegen. Bei den LIFO-Strategien ist NL0 überlegen, und NL2 wird wie schon bei der statistischen Analyse von den anderen beiden Strategien dominiert.

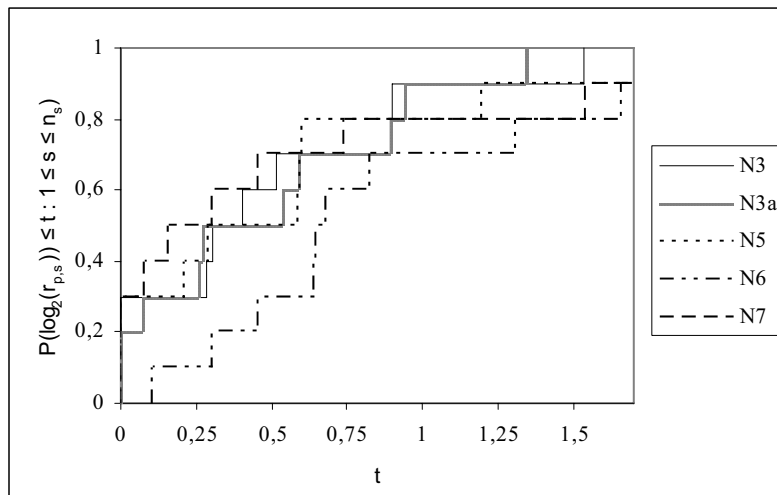


Abb. 6-2: Ausschnitt Performance Profil – Knotenauswahl

Tab. 6-7: Werte Knotenauswahl

	$\rho(1)$	$\rho(r_M)$	r_{\max}
N3	0,3	1	2,92
N3a	0,2	1	2,54
N5	0,3	0,9	r_M
N6	0	1	7,71
N7	0,3	1	8,04

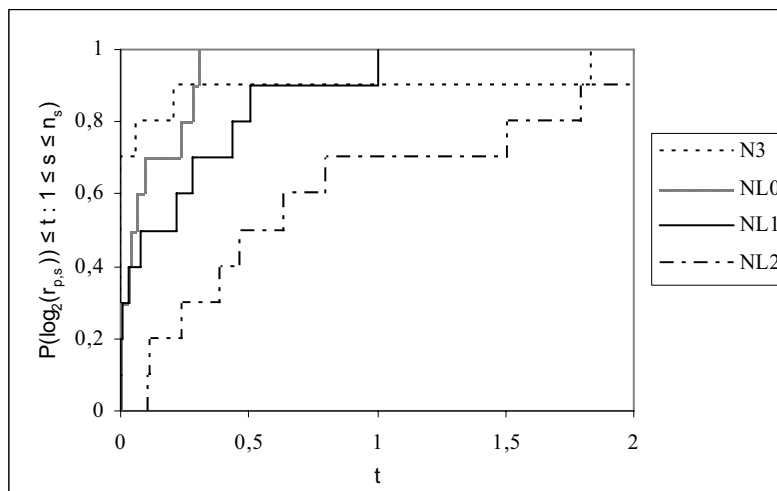


Abb. 6-3: Ausschnitt Performance Profile – Knotenauswahl mit LIFO

Tab. 6-8: Werte LIFO

	$\rho(1)$	$\rho(r_M)$	r_{\max}
N3	0,7	1	3,58
NL0	0,2	1	1,23
NL1	0,2	1	2,01
NL2	0	0,9	r_M

Schwere Modelle

Die Strategie N5 findet bei dano3mip keine Integer-Lösung. Die Strategien N6 und N7 weisen bis auf das Modell arki001 die gleichen numerischen Resultate auf, da mit Ausnahme von arki001 innerhalb der 60min kein Gap der schweren Modelle unter 5% kam und somit die dritte Phase nicht gestartet werden konnte. Die Strategie N7 fand bei dem Modell arki001 eine bessere Integer-Lösung als die Strategie N6 und weist somit ein besseres durchschnittliches Ranking auf. Demzufolge hat die Strategie N7 auch nur einen maximalen Rankingwert von vier, da im Fall des schlechtesten Resultats beide Strategien auf Platz vier gesetzt wurden. Die Strategie N3a weist das beste durchschnittliche Ranking auf und die Strategie N6 das schlechteste.

Bei dem Vergleich der Mixed-LIFO-Strategien mit N3 schneidet die Strategie N3 auch bei der schweren Testreihe besser ab als die LIFO-Strategien. Die durchschnittlichen Rankings der LIFO-Strategien liegen dicht beieinander.

Fazit

Sowohl bei den leichten als auch bei den schweren Modellen schneiden die beiden Best Projection Strategien N3 und N3a am besten ab. Beim Vergleich der Pseudo Kosten Strategien ist die zweiphasige Strategie N7 tendenziell besser als die einphasige Strategie N5. Die dreiphasige Strategie N6 schneidet bei den leichten Modellen sehr schlecht ab und kam aufgrund des Zeitlimits bei den schweren Modellen nur bei arki001 zum Einsatz. Bei den LIFO-Strategien gibt es keine klar dominierende Strategie.

6.3.2 Heuristiken

Bei den Heuristiken werden sowohl die Heuristiken vor dem Branch-and-Bound als auch die Heuristiken während des Branch-and-Bounds miteinander verglichen.

Heuristiken vor dem Branch-and-Bound

Beim Aufruf des Branch-and-Bounds in der Heuristik werden die Branching-Strategien $xbbheu=3$ und $xhnods=3$ verwendet. Um die Qualität der Heuristiken zu vergleichen, wird der Heuristik-Gap $xhgap=0.0$ gesetzt. Folgende Heuristiken werden untersucht:

- *H1*: Totales Runden ($xheutp=1$)
- *H2*: Local Branchen ($xheutp=2$)
- *H3a*: Relaxation Induced Neighborhood Search ($xheutp=3$). RINS startet vom Anfang an ($xvrins=1$).
- *H3b*: Relaxation Induced Neighborhood Search ($xheutp=3$). RINS startet erst, wenn im Totalen Runden eine Integer-Lösung gefunden wurde ($xvrins=0$).
- *H4*: Relaxation-based Search Space mit LIFO ($xheutp=4$)
- *H5*: Relaxation-based Search Space ($xheutp=5$)
- *H6*: Relaxation-based Search Space Heuristik mit Totalem Runden ($xheutp=6$). Das Totale Runden wird an jedem sechsten Knoten ($xnlbab=5$) ausgeführt.

Um das Verhalten einer Heuristik genau zu untersuchen, werden sowohl numerische Resultate mit $xmnheu=50$ als auch mit $xmnheu=200$ Knoten durchgeführt. Bei der Heuristik

$xheutp=1$ werden nur Rundungsdurchläufe ausgeführt und somit keine Knoten gelöst. Demzufolge wird beim ersten Durchlauf die Anzahl der Iterationen auf $xhmxit=21$ und beim zweiten Durchlauf auf $xhmxit=42$ gesetzt.

Finden mehrere Heuristiken bei einem Modell keine Integer-Lösungen, dann wird die Heuristik im Ranking besser bewertet, die weniger Zeit benötigt hat.

Tab. 6-9: Numerische Resultate – Heuristik vor dem Branch-and-Bound

		leicht			schwer		
		Ranking (min/avg/max)	Zeit (min)	Anzahl (Lösungen)	Ranking (min/avg/max)	Zeit (min)	Anzahl (Lösungen)
xmnheu=50	H1	(2/4,3/6)	1,98	6	(1/3,0/6)	9,07	6
	H2	(1/4,1/7)	1,46	6	(3/5,3/7)	8,27	1
	H3a	(1/4,4/7)	1,06	4	(2/4,9/7)	6,98	2
	H3b	(1/3,2/6)	1,42	6	(2/3,4/6)	9,78	6
	H4	(3/5,6/7)	1,0	7	(3/4,4/6)	6,74	3
	H5	(1/2,8/6)	0,9	8	(1/2,3/5)	6,49	6
	H6	(1/2,1/4)	1,0	9	(1/1,9/5)	6,81	6
xmnheu=200	H1	(2/5,8/7)	3,2	6	(2/4,8/7)	11,6	6
	H2	(1/3,6/6)	3,55	7	(1/4,2/7)	11,5	6
	H3a	(1/4,1/7)	1,28	5	(2/4,2/6)	9,08	5
	H3b	(1/3,2/6)	2,22	6	(1/3,6/6)	11,57	6
	H4	(2/5,1/7)	1,81	7	(1/4,7/7)	8,6	7
	H5	(1/2,9/7)	1,77	8	(1/2,2/4)	6,78	6
	H6	(1/2,4/7)	1,24	9	(1/2,0/4)	7,26	6

Leichte Modelle

Die Heuristik H6 findet nur bei dem Modell neos20 keine Integer-Lösung und schneidet beim Ranking deutlich am besten ab. Demzufolge führt das Totale Runden in der RSS-Strategie im Vergleich zu der reinen RSS-Strategie H5 zu besseren Ergebnissen, da die Heuristik H5 zusätzlich bei dem Modell cap6000 keine Lösung findet und im Ranking nur auf Platz zwei liegt.

Die RINS-Strategie mit dem Sofortstart (H3a) findet bei den wenigsten Modellen Integer-Lösungen und schneidet schlechter ab als H3b, wo erst das Totale Runden vorgeschaltet ist. Diese Kombination aus Totalem Runden und RINS führt zu einem besseren Ergebnis und zu einem geringeren Ressourcenverbrauch als das komplette Totale Runden.

Die gefundenen Lösungen bei der LIFO-Heuristik H4 sind im Vergleich zu den anderen Heuristiken schlecht, so dass trotz gefundener sieben Lösungen diese Heuristik das schlechteste bzw. im zweiten Durchlauf das zweitschlechteste durchschnittliche Ranking aufweist.

Beim zweiten Durchlauf mit erhöhter Knotenanzahl finden die Heuristiken H2 und H3a bei jeweils einem Modell eine Integer-Lösung mehr, als bei der geringeren Knotenanzahl. Die Erhöhung der Iterationanzahl beim Totalen Runden H1, führt zu einer Verschlechterung.

Schwere Modelle

Bei den schweren Modellen ist die Heuristik H6 gefolgt von der Heuristik H5 am besten. Bei dem ersten Durchlauf mit $xmnheu=50$ findet die Heuristik H2 nur bei dem Modell *markshare2* eine Integer-Lösung. Bei der erhöhten Knotenanzahl wird mit H2 bei fünf weiteren Modellen eine Integer-Lösung gefunden und bei der Heuristik H3a bei vier Modellen. Die Erhöhung der Knotenanzahl führt auch bei der Heuristik H4 dazu, dass bei vier weiteren Modellen eine Integer-Lösung gefunden wird. Somit findet die Heuristik H4 im zweiten Durchlauf bei den meisten Modellen Integer-Lösungen, welche aber im Vergleich zu den anderen Strategien schlechter sind, so dass die Heuristik nach dem Ranking im Hinterfeld liegt. Die Erhöhung der Anzahl der Iterationen bei der Heuristik H1 führt wie bei den leichten Modellen zu einer Verschlechterung.

Fazit

Bei beiden Durchläufen findet die Heuristik H6 bei 75% der Modelle Integer-Lösungen, die im Durchschnitt am besten zu bewerten sind.

Insgesamt gesehen bewirkt eine erhöhte Knotenanzahl nur bei den Heuristiken H2, H3a und H4 einen Anstieg der Anzahl der gefundenen Integer-Lösungen. Eine Erhöhung der Anzahl der Iterationen bei der Heuristik H1 führt sowohl bei den leichten als auch bei den schweren Modellen zu einer Verschlechterung. Die RINS-Variante H3b schneidet bei beiden Testsets besser ab als H3a.

Local Search während dem Branch-and-Bound

Die Local Search im Branch-and-Bound wird alle $xnlbab=1000$ Knoten aufgerufen und für maximal $xnlstr=1000$ Knoten durchgeführt. Die gewählte Branchingheuristik ist $xbrheu=5$ und die Knotenauswahlstrategie $xnodse=3$ mit $xnodbf=1$. Folgende Strategien werden untersucht:

- *L0*: ohne Local Search ($xlocs=0$)
- *L1*: Relaxation Induced Neighborhood Search ($xlocs=1$)
- *L2*: Local Branching ($xlocs=2$)
- *L3*: Local Rounding ($xlocs=3$)
- *L4*: Local Total Rounding ($xlocs=4$), wobei $xnlstr=1$ und $xnlbab=5$ ist.

Tab. 6-10: Numerische Resultate – Local Search

	leicht					schwer		
	Knoten	Iteration	Zeit		nicht opt.	Ranking (min/avg/max)	keine	Ranking (min/avg/max)
			Summe	Median				
L0	3104725	20851671	116,21	8,78	0	(2/3,2/5)	0	(1/2,4/4)
L1	3111849	19842051	111,67	6,72	0	(1/3,1/5)	0	(1/2,2/4)
L2	3060505	19308093	129,42	11,21	0	(1/3,5/5)	0	(1/3,0/5)
L3	3135472	20179434	112,48	7,54	0	(1/2,6/4)	0	(1/2,6/5)
L4	2821899	18863487	104,94	6,7	0	(1/2,3/4)	0	(1/3,6/5)

Leichte Modelle

Bei den fünf Strategien werden alle Modelle optimal gelöst. Die Strategie L4 löst die Modelle insgesamt am schnellsten und schneidet beim durchschnittlichen Ranking mit Abstand am besten ab. Die beiden Rundungsstrategien L3 und L4 lösen kein Modell am schlechtesten. Die Strategie L2 weist den größten Ressourcenverbrauch und das schlechteste Ranking auf. Diese Strategie löst sieben der zehn Modelle am schlechtesten und die restlichen drei Modelle am Besten. Die Strategie ohne Local Search L0 liegt mit dem Ressourcenverbrauch und dem Ranking im Mittelfeld, löst aber keines der Modelle am besten.

Der Ausschnitt des Performance Profils in Abb. 6-4 zeigt, dass die Strategie L4 den anderen Strategien überlegen ist. Zwar lösen die Strategien L2 und L3 auch jeweils 30% der Modelle am schnellsten, aber die Strategie L4 benötigt nur den Faktor 1,17mal mehr Zeit als die optimale Strategie, um alle Modelle zu lösen. Die Strategie L2 wird, wie schon bei der statistischen Analyse, von den anderen Strategien dominiert.

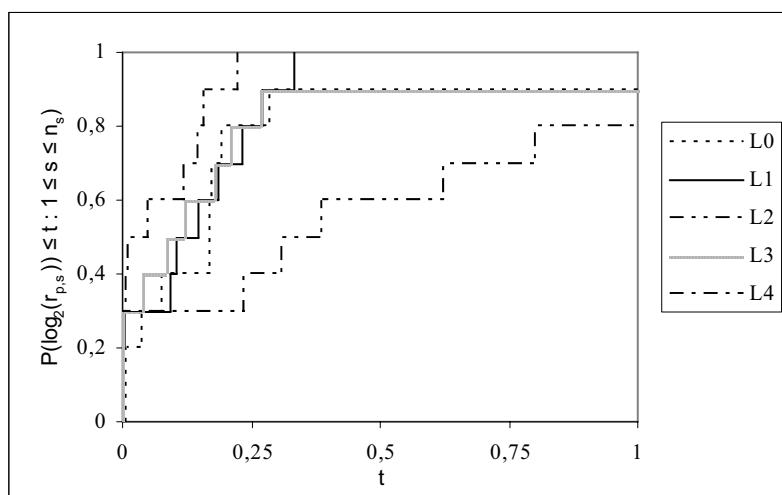


Abb. 6-4: Ausschnitt Performance Profile – Local Search

Schwere Modelle

Bei den schweren Modellen schneidet die RINS-Strategie L1 am Besten ab, die kein Modell am Schlechtesten löst. Die Strategie ohne Local Search L0 hat ein besseres Ranking als die restlichen Strategien L2-L4 und löst auch kein Modell am Schlechtesten. Das schlechteste durchschnittliche Ranking weist die Total-Rounding-Strategie L4 auf.

Tab. 6-11: Werte Local Search

	$\rho(1)$	$\rho(r_M)$	r_{\max}
L0	0	1	3,18
L1	0,2	1	1,26
L2	0,3	1	3,48
L3	0,3	1	2,56
L4	0,3	1	1,17

Fazit

Insgesamt gesehen bewirkt der Einsatz einer Local Search nicht unbedingt eine schnellere Lösung der Modelle bzw. das Finden einer besseren Integer-Lösung. Keine der Local Search Strategien ist in beiden Testsets dominierend. So ist im leichten Testset die Strategie L4 überlegen, scheitert aber am schweren Testset. Im schweren Testset hat die Strategie L1 den größten Erfolg, liegt aber beim leichten Testset nur im Mittelfeld. Besonders anhand der Local Branching Strategie L2 ist zu erkennen, dass diese Strategien bei einigen Modellen zum Erfolg führen, aber wiederum bei einzelnen Modellen am schlechtesten abschneiden.

6.3.3 Branch-and-Bound mit Heuristik

Zum Abschluss werden die Modelle mit Heuristik ($xheutp=6$) vor dem Branch-and-Bound gelöst. Die Branching-Strategien sind $xbrheu=5$ und $xnodse=3$ mit $xnodbf=1$. Folgende Strategien werden untersucht:

- G1: Branch-and-Bound mit Heuristik
- G2: Branch-and-Bound mit Heuristik und RINS
- G3: Branch-and-Bound mit Heuristik und Probing während des Branch-and-Bounds

Tab. 6-12: Numerische Resultate – Branch-and-Bound mit Heuristik

	leicht					schwer		
	Knoten	Iteration	Zeit		nicht opt.	Ranking (min/avg/max)	keine	Ranking (min/avg/max)
			Summe	Median				
G1	3053190	23552760	154,53	9,74	0	(1/1,8/3)	0	(1/1,8/3)
G2	3078813	23427852	154,99	9,38	0	(1/1,7/3)	0	(1/1,8/3)
G3	2946655	22000144	194,21	9,76	0	(1/2,4/3)	0	(1/1,7/3)

Leichte Modelle

Die beiden Strategien G1 und G2 benötigen zum Lösen aller Modelle ungefähr die gleichen Ressourcen und liegen beim durchschnittlichen Ranking dicht beieinander. Die Probing-Strategie G3 braucht wesentlich mehr Ressourcen und liegt beim Ranking abgeschlagen hinten.

Der Ausschnitt des Performance Profiles in Abb. 6-5 zeigt, dass die Strategie G3 den anderen beiden Strategien unterlegen ist. Die Strategie G2 löst die meisten Modelle (50%) in kürzester Zeit und benötigt auch nur den Faktor 1.19 im Vergleich zur optimalen Strategie, um alle Modelle optimal zu lösen. Die Strategie G1 ist trotz einer optimalen Lösungsrate von nur 40% am Anfang der Strategie G2 überlegen, hat aber das Modell neos20 als Ausreißer.

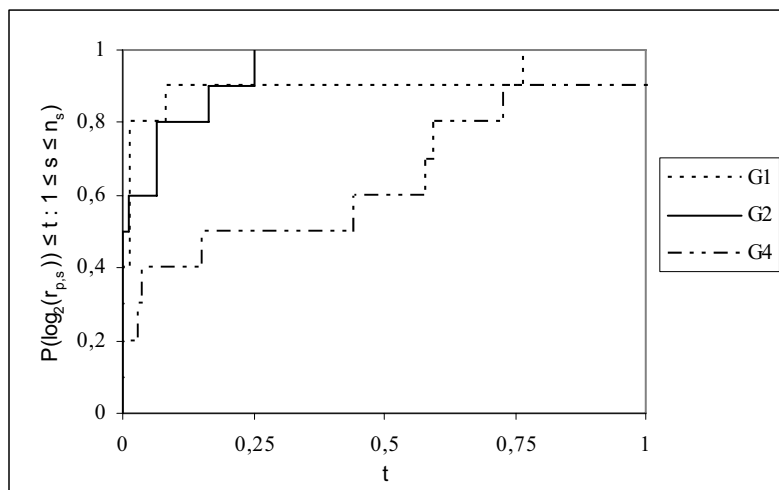


Abb. 6-5: Ausschnitt Performance Profile – B&B mit Heuristik

Tab. 6-13: Werte B&B und Heuristik

	$\rho(1)$	$\rho(r_M)$	r_{\max}
G1	0,4	1	1,7
G2	0,5	1	1,19
G3	0,2	1	2,2

Schwere Modelle

Bei dem schweren Testset ist die Probing-Strategie G3 vom Ranking minimal besser als die beiden anderen Strategien G1 und G2. Die Strategie G3 findet bei den zehn Modellen fünf Mal die beste Lösung, während G2 bei vier Modellen und G1 bei drei Modellen die beste Integer-Lösung findet.

Fazit

Zwischen den drei Strategien gibt es keine eindeutig beste Strategie. Die Probing-Strategie G3 benötigt besonders bei einigen leichten Modellen zu viel Zeit, um diese zu lösen. Die RINS-Strategie löst im Vergleich zur Strategie G1 jeweils ein Modell schneller bzw. findet bei einem Modell mehr eine bessere Integer-Lösung.

6.4 Diskussion der Gesamtergebnisse

Vergleich mit und ohne Heuristik vor dem Branch-and-Bound

Nicht immer führt eine gefundene IP-Lösung in der Heuristik vor dem Branch-and-Bound zu einer schnelleren Optimierung. Um zu untersuchen, inwieweit eine Heuristik die Optimierung beeinflusst, werden folgende Strategien mit $xnodse=3$ und $xnodbf=1$ miteinander verglichen:

- *Vergleich 1 (V1)*: Branchingheuristik $xbrheu=5$ mit (G1) und ohne (N3a) Heuristik vor dem Branch-and-Bound
- *Vergleich 2 (V2)*: Branchingheuristik $xbrheu=5$ und Local Search $xlocs=1$ mit (G2) und ohne (L1) Heuristik vor dem Branch-and-Bound

Tab. 6-14: Numerische Resultate – Vergleich mit und ohne B&B

		leicht					schwer		
		Knoten	Iteration	Zeit		nicht opt.	Ranking avg	keine	Ranking avg
Summe	Median								
V1	N3a	3101738	20461150	115,02	8,58	0	1,1	0	1,6
	G1	3053190	23552760	154,53	9,74	0	1,9	0	1,4
V2	L1	3111849	19842051	111,67	6,72	0	1,2	0	1,4
	G2	3075314	23165542	153,17	8,95	0	1,6	0	1,6

Leichte Modelle

Die Strategien ohne Heuristik vor dem Branch-and-Bound benötigen weniger Ressourcen und weisen ein besseres Ranking auf als die Strategien mit Heuristik.

Schwere Modelle

Bei der schweren Testreihe ist im Vergleich V1 die Strategie mit Heuristik überlegen. Bei dem Vergleich V2 kommt die Strategie ohne Heuristik auf ein besseres durchschnittliches Ranking.

Fazit

Besonders im leichten Testset hat die Heuristik vor dem Branch-and-Bound keinen positiven Einfluss auf die Optimierung der Modelle. In diesem Testset wird von der Heuristik $x_{heutp}=6$ bis auf das Modell neos20 für jedes Modell eine Integer-Lösung gefunden. Wenn allerdings die Modelle ohne Heuristik gelöst werden, wird bei sieben der zehn Modelle schon innerhalb der ersten 100 Knoten eine Integer-Lösung gefunden. Somit ist bei diesen Modellen eine Heuristik eigentlich überflüssig und das zweite Supernode Processing bringt bei diesen Modellen auch keinen Vorteil. Nur bei den Modell neos20 und cap6000 benötigt der Branch-and-Bound wesentlich länger, eine Integer-Lösung zu finden. Das Modell cap6000 wird durch die Heuristik wesentlich schneller gelöst.

Bei den schweren Modellen findet die Heuristik bei sechs Modellen eine Integer-Lösung. Bei beiden Vergleichen wird bei vier von den sechs Modellen durch die in der Heuristik gefundene Integer-Lösung eine bessere IP-Lösung gefunden. Wenn in der Heuristik keine Integer-Lösung gefunden wird, dann startet der Branch-and-Bound mit einer anderen Basis als bei dem Durchlauf ohne Heuristik. Demzufolge ist es von der Startbasis abhängig, ob bei einem Modell eine bessere Integer-Lösung gefunden wird oder nicht.

Vergleich der numerischen Resultate vom neuen B&B und alten B&B

Die Ergebnisse des neuen Branch-and-Bounds werden mit den Ergebnissen des alten Branch-and-Bounds verglichen, um eventuelle Verbesserungen aufzuzeigen. Dabei wird einmal ein Vergleich mit und einmal ohne Heuristik vor dem Branch-and-Bound durchgeführt. Damit eventuelle Verbesserungen bzw. Verschlechterungen nicht nur auf eine andere Branching-Strategie zurückzuführen sind, wird der Vergleich mit den beiden Branching-Strategien drei und fünf durchgeführt:

- *Vergleich 3 (V3)*: Die Strategien B3 und N3a werden mit dem alten Branch-and-Bound ohne Heuristik (A1) verglichen

- *Vergleich 4 (V4)*: Die Strategien G1 und G4 werden mit dem alten Branch-and-Bound mit der Default-Heuristik $xheutp=4$ (A2) verglichen

Der alte Branch-and-Bound wird mit den Strategien $xbrheu=3$ und $xnodes=3$ ausgeführt.

Tab. 6-15: Numerische Resultate – Vergleich alter und neuer B&B

		leicht					schwer		
		Knoten	Iteration	Zeit		nicht opt.	Ranking (min/avg/max)	keine	Ranking (min/avg/max)
				Summe	Median				
V3	B3	824086	10645190	27,33	3,9	3	(1/2,0/3)	1	(1/2,1/3)
	N3a	797495	8561202	28,26	5,51	0	(1/1,6/3)	0	(1/1,7/3)
	A1	826623	6802761	48,68	5,61	5	(1/2,4/3)	0	(1/2,2/3)
V4	G1	895591	9880965	34,4	6,47	0	(1/1,6/3)	0	(1/1,5/3)
	G4	814676	10472435	27,84	4,16	4	(1/2,2/3)	0	(1/1,6/3)
	A2	728508	9528115	55,96	7,33	3	(1/2,2/3)	0	(2/2,8/3)

Leichte Modelle

Bei dem Vergleich V3 wurden von der Strategie B3 die Modelle cap6000, neos20 und qiu und von der Strategie A1 die Modelle cap6000, mas76, neos20, qiu und swath1 nicht optimal gelöst. Bei dem Modell neos20 fanden beide und bei dem Modell cap6000 nur die Strategie A1 keine Integer-Lösung. Diese fünf Modelle wurden aus den Ressourcensummen entfernt.

In dem Vergleich V4 wurden die Modelle cap6000 (G4,A2), mas76 (A2), neos20 (G4,A2), qiu (G4) und swath1 (G4) nicht optimal gelöst und wurden in die Ressourcensummen nicht einbezogen.

Für die jeweils restlichen fünf Modelle benötigt der alte Branch-and-Bound A1/A2 deutlich die meisten Ressourcen. Der neue Branch-and-Bound mit der Pseudo Kosten-Strategie N3a/G1 weist jeweils das beste durchschnittliche Ranking auf. Die Strategien B3/G4 und A1/A2 haben die gleiche Branchingheuristik. Nach der statistischen Analyse sind die Strategien B3/G4 den Strategien A1/A2 überlegen bzw. mit Heuristik gleich bewertet.

Das Performance Profile in Abb. 6-6 zeigt, wie schon die statistische Analyse ergeben hat, dass im Vergleich V3 die Strategie N3a klar überlegen ist. Der alte Branch-and-Bound ist den beiden Strategien des neuen Branch-and-Bounds deutlich unterlegen.

In der Abb. 6-7 zeigt das Performance Profile beim Vergleich V4, dass die Strategie G1 eindeutig den anderen Strategien überlegen ist. Der Vergleich zwischen G4 und A2 ist wie in der statistischen Analyse nicht eindeutig.

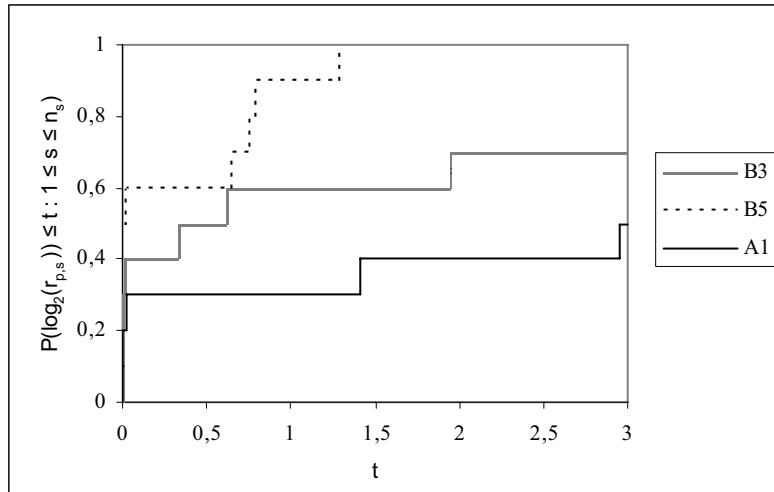


Abb. 6-6: Performance Profile V3

Tab. 6-16: Werte V3

	$\rho(1)$	$\rho(r_M)$	r_{\max}
B3	0,3	0,7	r_M
N3a	0,5	1	2,45
A1	0,2	0,5	r_M

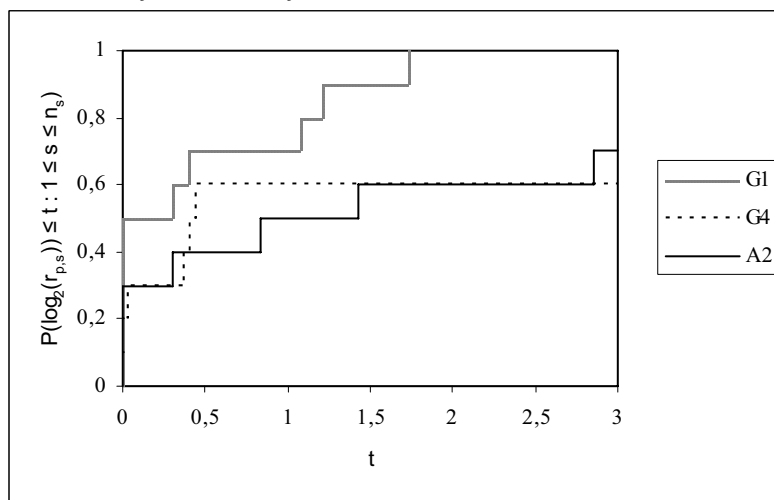


Abb. 6-7: Performance Profile V4

Tab. 6-17: Werte V4

	$\rho(1)$	$\rho(r_M)$	r_{\max}
G1	0,5	1	3,32
G4	0,2	0,6	r_M
A2	0,3	0,7	r_M

Schwere Modelle

Die Strategien mit Pseudo Kosten (N3a und G1) schneiden auch bei den schweren Modellen am besten ab. Ohne Heuristik ist B3 nur geringfügig besser als A1. Allerdings findet B3 bei dem Modell liu keine Integer-Lösung. Bei den Strategien mit Heuristik schneidet G4 nur geringfügig schlechter ab als die Pseudo-Kosten-Strategie G1. Der alte Branch-and-Bound findet bei allen Modellen bis auf das Modell roll3000 nur die schlechteste Integer-Lösung.

Fazit

Sowohl bei den leichten als auch bei den schweren Modellen schneiden die Pseudo Kosten Strategien in beiden Vergleichen deutlich am Besten ab.

Bei dem Vergleich der beiden Branch-and-Bounds mit der gleichen Branchingheuristik $xbrheu=3$ sind die Strategien B3/G4 des neuen Branch-and-Bounds dem alten Branch-and-Bound A1/A2 überlegen bzw. beim leichten Testset gleichwertig (V4).