

7 Beurteilung und Ausblick

Die numerischen Ergebnisse wurden nur mit einem Testset von insgesamt 20 Modellen über jeweils maximal 60 Minuten ermittelt. Diese Ergebnisse geben somit nur eine grobe Einschätzung der Strategien wieder. Bei diesen Modellen hat sich keine klar dominierende Strategie herauskristallisiert. Am Besten schnitten die Strategie N3a und die RINS-Strategie L1 ab. Eine Heuristik vor dem Branch-and-Bound führte bei vielen der getesteten Modelle zu keinem Erfolg, da ohne Heuristik im Branch-and-Bound bei diesen Modellen schnell eine Integer-Lösung gefunden wurde. Wenn allerdings der Branch-and-Bound Schwierigkeiten hat, eine Integer-Lösung zu finden, dann ist der Einsatz einer Heuristik empfehlenswert.

Die anderen getesteten Strategien weisen bei einigen Modellen Schwächen auf. So schnitt z.B. die Local Total Rounding Strategie (L4) bei den leichten Modellen am Besten ab, fand aber bei den schweren Modellen zu wenige gute Integer-Lösungen. Wird beim Einsatz des neuen Branch-and-Bounds ein Modell mit den Standardeinstellungen nicht zufriedenstellend gelöst, dann empfiehlt es sich, auch die anderen Strategien zu testen, da sie bei manchen Modellen optimal waren.

Deutlich war allerdings erkennbar, dass der neue Branch-and-Bound dem alten Branch-and-Bound überlegen ist. Diese Überlegenheit wird durch die veränderte Struktur, die neuen Strategien, die veränderte Bound Reduction und die neuen Heuristiken vor- und während des Branch-and-Bound-Prozesses erzielt.

Das Ziel der Dissertation, durch den Einsatz der genannten Veränderungen schneller und bessere IP-Lösungen zu finden, wurde bei einem Großteil der Modelle erfüllt. Insgesamt wurden bei den Strategien N3a/G1 und L1/G2 im Vergleich zu den Strategien des alten Branch-and-Bounds A1/A1 bei 70% der leichten Modelle schneller eine optimale Lösung gefunden. Bei den schweren Modellen fand der neue Branch-and-Bound bei 60% der Modelle erheblich schneller eine Integer-Lösung, während der alte Branch-and-Bound nur bei 20% der Modelle signifikant schneller eine IP-Lösung fand.

Aber nicht jede entwickelte Strategie bzw. Heuristik erreicht an jedem Modell die gewünschten Ansprüche. Demzufolge gibt es weitere Verbesserungsmöglichkeiten, die kurz skizziert werden:

- *Lösungsverlauf analysieren:* Der Einsatz von einer Local Search kann insbesondere bei recht einfachen Modellen zu einer Verschlechterung der Lösungszeit führen. Darum sollte erst eine Local Search gestartet werden, wenn sicher ist, dass das Finden einer Integer-Lösung schwer ist, bzw. der Gap im Laufe des Branch-and-Bound-Prozesses konstant relativ groß bleibt.
- *Heuristiken weiter verbessern:* Bei den entwickelten Heuristiken ist zwar die Anzahl der gefundenen IP-Lösungen größer als bei der Standard-Heuristik des alten Branch-and-Bound-Verfahrens, aber dennoch wird bei einer Teilmenge der Modelle noch keine IP-Lösung gefunden. Bei dem schweren Testset wurde bei 2/3 der Modelle, bei denen in der Heuristik eine Integer-Lösung gefunden wurde, eine bes-

sere Integer-Lösung gefunden. Somit könnte eine verbesserte Heuristik bei den schweren Modellen zu besseren Ergebnisse führen. Außerdem müsste überprüft werden, ob eine Heuristik vor dem Branch-and-Bound insbesondere bei einfachen Modellen überhaupt sinnvoll ist.

- *Flexiblere Anpassung der Parameter:* Manche Parameter sollten nicht fest durch den Benutzer gegeben sein, sondern ihren Wert aufgrund des Lösungsverlaufs ändern können, wie z.B. die Knotenlimits der Local Search *xnlbab* oder *xnlstr*, der Heuristik-Gap *xhgap* oder der Gap für den Start von LIFO oder Local Search an einem Knoten *xhligp* und *xhlsqp*. So würden sich die Strategien flexibler den Modellgegebenheiten anpassen und eventuell mehr Nutzen bringen.
- *Modelle analysieren:* Eine Analyse der Modelldaten würde zu einer genaueren Einstellung der Parameter, wie z.B. dem Rundungsverhalten bei der Heuristik und den Branching-Strategien, führen.
- *Verbesserung der Lower Bound *xzlbnd* durch einen Branch-and-Cut-Ansatz:* Bei einigen Modellen war zu beobachten, dass relativ früh im Branch-and-Bound eine optimale bzw. eine gute IP-Lösung gefunden wurde. Allerdings wurde danach viel Zeit benötigt, diese Optimalität zu beweisen bzw. den Gap zu verkleinern. Durch einen Branch-and-Cut-Ansatz würden eventuell Schnittebenen die Lower Bound *xzlbnd* anheben und somit überflüssige Knoten abschneiden. Folglich könnte dadurch die Optimalität schneller bewiesen werden bzw. der Gap verkleinert werden.