Eine Heuristik zur Lösung von Stapelproblemen

L. Kämmerer
L. Hempel

Bauhaus-Universität Weimar

1 Formale Beschreibung von Stapelproblemen

Um die Problemstellung in einem streng mathematischen Sinn erfassen und damit arbeiten zu können, wollen wir kurz eine formal-theoretische Beschreibung von Stapelproblemen angeben.

Definition 1.1 Ein Graph $G = (V, A)$ heißt genau dann *(verzweigter) Stapel*, wenn G endlich, einfach, gerichtet und zyklenfrei ist. Im folgenden Text werden wir auch $V(G)$ für V und $A(G)$ für A schreiben.

Definition 1.2 Es sei $G = (V, A)$ ein Stapel, dann sind SPITZE(G) und GRUND(G) wie folgt definiert:

$$\text{SPITZE}(G) := \{v \in V : d_G^-(v) = 0\} \text{ und } \text{GRUND}(G) := \{v \in V : d_G^+(v) = 0\},$$

wobei $d_G^-(v)$ die Anzahl der in v eingehenden Kanten in G angibt und $d_G^+(v)$ die Anzahl der aus v herausführenden Kanten.

Definition 1.3 $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$ heißt genau dann *verzweigtes Stapelproblem*, wenn $G_\mathrm{A} = (V, A_\mathrm{A})$ und $G_\mathrm{Z} = (V, A_\mathrm{Z})$ verzweigte Stapel sind und k eine natürliche Zahl ist. G_A nennen wir *Ausgangsstapel*, G_Z nennen wir *Zielstapel*; k gibt die Anzahl der *Hilfsstapelplätze* an. Die Knoten v aus V nennen wir *Stapelelemente*.

Definition 1.4 $S = (G_1, G_2, H, M)$ heißt genau dann *Stapelkonfiguration eines verzweigten Stapelproblems* $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$, wenn gilt:

i.) $G_1 = (V_1, A_1)$ und $G_2 = (V_2, A_2)$ sind Stapel,

ii.) $H \subseteq M \subseteq V$, $|M| \leq k$,

iii.) $H \cup V_1 \cup V_2 = V$ und

iv.) H, V_1 und V_2 sind paarweise disjunkt.

Eine solche Stapelkonfiguration S bezeichnet den Zustand des Ausgangs- und des Zielstapelplatzes G_1 und G_2 und gibt die Menge H aller ausgestapelten Elemente an, die auf den k Hilfsstapelplätzen liegen. Dabei darf auf jedem Hilfsstapelplatz nur ein Stapelelement liegen.

Wird vom Ausgangsstapel ein Element weggenommen und auf einen anderen Stapel gelegt oder von einem der Hilfsstapelplätze ein Element auf den Zielstapel gebracht, so ergibt sich eine neue Stapelkonfiguration. Diesen Vorgang nennen wir *Umstapelung*.

Definition 1.5 Gegeben seien 2 Stapelkonfigurationen $S = (G_1, G_2, H, M)$ und $S' = (G'_1, G'_2, H', M')$ eines verzweigten Stapelproblems $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$. S' heißt genau dann *Umstapelung von* S, wenn ein Stapelelement $v \in V(G_\mathrm{A})$ existiert, für das eine der drei folgenden Bedingungen erfüllt ist:

i.) $v \in \text{SPITZE}(G_1)$, $H' = H$, $G'_1 = G_1 \setminus \{v\}$, $M' = M$ und es ist $G_2 = (\emptyset, \emptyset)$ und $G'_2 = (\{v\}, \emptyset)$, oder es existiert ein $u \in \text{GRUND}(G_2)$ mit $(u, v) \in A(G_\mathrm{Z})$ und $(u, v) \in G'_2$,

ii.) $v \in \text{SPITZE}(G_1)$, $H' = H \cup \{v\}$, $G'_1 = G_1 \setminus \{v\}$, $G'_2 = G_2$, $M' = M \cup \{v\}$,

iii.) $v \in H$, $H' = H \setminus \{v\}$, $G'_1 = G_1$, $M' = M$ und es ist $G_2 = (\emptyset, \emptyset)$ und $G'_2 = (\{v\}, \emptyset)$, oder es existiert ein $u \in \text{GRUND}(G_2)$ mit $(u, v) \in A(G_\mathrm{Z})$ und $(u, v) \in G'_2$.

In die Menge M werden all diejenigen Stapelelemente eingetragen, für die es eine Stapelkonfiguration gibt, in der sie auf einem der Hilfsstapelplätze liegen. Jedem Hilfsstapelplatz wird somit genau ein Stapelelement zugewiesen – kein anderes Element darf mehr auf diesem Platz liegen ($|M| \leq k$ nach Definition 1.4).

Ist S' Umstapelung von S, so schreiben wir auch $S \overset{v}{\longmapsto} S'$. Wird v dabei auf den Zielstapelplatz gelegt, dann nennen wir diesen Vorgang auch *Einstapelung*. Wird v auf einen Hilfsstapelplatz gelegt, dann nennen wir diese Umstapelung *Ausstapelung*.

Definition 1.6 Es sei $Q = (G_{\mathrm{A}}, G_{\mathrm{Z}}, k)$ ein verzweigtes Stapelproblem. Eine Folge $\mathcal{S} = [S_1, \ldots, S_l]$ von Stapelkonfigurationen $S_i = (G_1^i, G_2^i, H^i, M^i)$ heißt genau dann *zulässige Stapelfolge von Q*, wenn gilt:

i.) $G_1^1 = G_{\mathrm{A}}$, $G_2^1 = (\emptyset, \emptyset)$, $M^1 = \emptyset$ und

ii.) für $i \in \{1, \ldots, l-1\}$ ist S_{i+1} Umstapelung von S_i.

Die Stapelkonfigurationen S_i nennen wir dann zulässige Stapelkonfigurationen von Q.

Definition 1.7 Ein verzweigtes Stapelproblem $Q = (G_{\mathrm{A}}, G_{\mathrm{Z}}, k)$ heißt genau dann *lösbar*, wenn eine zulässige Stapelfolge $\mathcal{S} = [S_1, \ldots, S_l]$ existiert, so daß für $S_l = (G_1^l, G_2^l, H^l, M^l)$

$$G_1^l = (\emptyset, \emptyset),\ G_2^l = G_{\mathrm{Z}} \text{ und } H^l = \emptyset$$

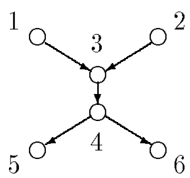
gilt. $\mathcal{S}$ heißt dann *l-Umstapelfolge für Q*.

Sind nun zwei verzweigte Stapel G_{A} und G_{Z} gegeben, so stellt sich die Frage nach der kleinsten Anzahl k an Hilfsstapelplätzen, für die das Stapelproblem $Q = (G_{\mathrm{A}}, G_{\mathrm{Z}}, k)$ lösbar ist. Das entspricht der Anzahl der Stapelelementen, die ausgestapelt werden müssen, damit Q gelöst werden kann.

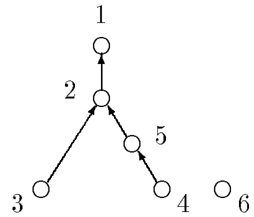
Beispiel: Es sei $Q = (G_{\mathrm{A}}, G_{\mathrm{Z}}, k)$ ein verzweigtes Stapelproblem mit

$$\begin{aligned} & G_{\mathrm{A}} = (\{1,2,3,4,5,6\},\ \{(1,3),\ (2,3),\ (3,4),\ (4,5),\ (4,6)\}) \\ \text{und}\quad & G_{\mathrm{Z}} = (\{1,2,3,4,5,6\},\ \{(4,5),\ (3,2),\ (5,2),\ (2,1)\}) \end{aligned}$$

Ausgangsstapel G_{A}:



Zielstapel G_{Z}:



Die kleinste Anzahl an Hilfsstapelplätzen, für die Q lösbar bleibt, ist $k = 2$. Eine Umstapelfolge:

$$S_1 \xmapsto{1} S_2 \xmapsto{2} S_3 \xmapsto{3} S_4 \xmapsto{4} S_5 \xmapsto{5} S_6 \xmapsto{2} S_7 \xmapsto{1} S_8, \xmapsto{6} S_9$$

mit $S_1 = (G_{\mathrm{A}},\ (\emptyset, \emptyset),\ \emptyset,\ \emptyset)$ und $S_9 = ((\emptyset, \emptyset),\ G_{\mathrm{Z}},\ \emptyset,\ \{1,2\})$.

2 Weitere Eigenschaften und NP-Vollständigkeit

In diesem Abschnitt werden wir die Struktur des Problems näher untersuchen.

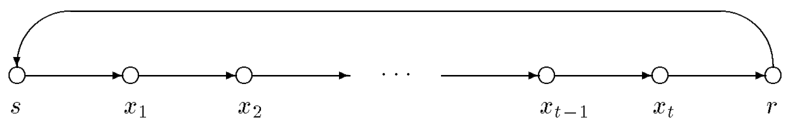
Lemma 2.1 Es sei $Q = (G_{\mathrm{A}}, G_{\mathrm{Z}}, k)$ ein lösbares, verzweigtes Stapelproblem. Dann gibt es eine Knotenmenge $V' \subseteq V(G_{\mathrm{A}})$ mit $|V'| \leq k$, so daß $(G_{\mathrm{A}} \setminus \{(v,u) : \forall v \in V' : (v,u) \in A(G_{\mathrm{A}})\}) \cup G_{\mathrm{Z}}$ kreisfrei ist.

Beweis: Es sei $\mathcal{S} = [S_1, \ldots, S_l]$ eine l-Umstapelfolge von Q mit $S_i = (G_1^i, G_2^i, H^i, M^i)$. Als V' wählen wir die Menge aller Stapelelemente, die auf die Hilfsstapelplätze augestapelt wurden: $V' := M^l$ Damit ist klar, daß $|V'| \leq k$ gilt. Wir zeigen nun, daß $(G_{\mathrm{A}} \setminus \{(v,u) : \forall v \in V' : (v,u) \in A(G_{\mathrm{A}})\}) \cup G_{\mathrm{Z}}$ kreisfrei ist.

Angenommen, $(G_{\mathrm{A}} \setminus \{(v,u) : \forall v \in V' : (v,u) \in A(G_{\mathrm{A}})\}) \cup G_{\mathrm{Z}}$ enthielte einen Kreis C: Da G_{A} und G_{Z} kreisfrei sind, enthält C sowohl Kanten aus G_{A}, als auch aus G_{Z}. Es sei also $(r,s) \in A(C) \cap A(G_{\mathrm{A}})$ eine Kante aus C, die auch in G_{A} liegt. Weiterhin gelte

$$(s, x_1),\ (x_1, x_2),\ \ldots,\ (x_{t-1}, x_t),\ (x_t, r) \in A(C).$$

C :



Wir wollen nun s vom Ausgangsstapelplatz auf den Zielstapelplatz bringen. In G_{A} liegt jedoch r auf s, so daß zunächst r vom Ausgangsstapel genommen werden muß. Wegen $(x_t, r) \in A(C)$ gilt $(x_t, r) \in A(G_{\mathrm{A}})$ oder $(x_t, r) \in A(G_{\mathrm{Z}})$. In jedem Fall muß x_t vom Ausgangsstapel auf den Zielstapel gebracht werden, bevor r dorthin gelangen kann, denn entweder liegt x_t im Ausgangsstapel auf r oder r liegt im Zielstapel auf x_t. Setzen wir diese Argumentation fort, so erhalten wir, daß s vor x_1, x_1 vor x_2, …, x_{t-1} vor x_t vom Ausgangsstapel genommen werden müssen. Somit ist auch s vor r vom Ausgangsstapelplatz auf den Zielstapelplatz zu bringen, d. h. es muß ein Knoten $x_i \in \{x_1, \ldots, x_t\}$ auf einen Hilfsstapelplatz ausgestapelt werden. Für x_i gilt damit $x_i \in V'$ und $(x_i, x_{i+1}) \in A(G_{\mathrm{A}})$. Das widerspricht jedoch der Wahl von C, da C in $(G_{\mathrm{A}} \setminus \{(v,u) : \forall v \in V' : (v,u) \in A(G_{\mathrm{A}})\}) \cup G_{\mathrm{Z}}$ liegt. Damit ist $(G_{\mathrm{A}} \setminus \{(v,u) : \forall v \in V' : (v,u) \in G_{\mathrm{A}}\}) \cup G_{\mathrm{Z}}$ kreisfrei. $\square$

Betrachten wir nun Stapelprobleme ohne Hilfsstapelplätze in bezug auf ihre Lösbarkeit. Es gilt:

Satz 2.2 *Ein verzweigtes Stapelproblem $Q = (G_A, G_Z, 0)$ ist genau dann lösbar, wenn $G_A \cup G_Z$ kreisfrei ist.*

Beweis: („⇒") Nach Lemma 2.1 folgt, daß $G_A \cup G_Z$ kreisfrei ist.

(„⇐") Die Vereinigung von Ausgangs- und Zielstapel $G_A \cup G_Z$ sei kreisfrei. Angenommen, Q sei nicht ohne Ausstapelungen lösbar – d. h. es existiert keine Umstapelfolge für Q. Somit gilt für jede zulässige Stapelfolge maximaler Länge $\mathcal{S} = [S_1, \ldots, S_l]$ von Q:

$$S_l \neq ((\emptyset, \emptyset), G_Z, \emptyset, \emptyset).$$

Wir wählen nun eine solche Stapelfolge $\mathcal{S}$ von Q und betrachten die Stapelkonfiguration $S_l = (G_1^l, G_2^l, H^l, M^l)$. Da für die Lösung von Q kein Zwischenstapelplatz zur Verfügung steht, gilt $H^l = M^l = \emptyset$. Somit muß $G_1^l \neq (\emptyset, \emptyset)$ und $G_2^l \neq G_Z$ gelten. Weil $\mathcal{S}$ eine maximale Stapelfolge von Q ist, läßt sich keine Stapelkonfiguration S_{l+1} finden, die Umstapelung von S_l ist. Es läßt sich demnach kein Element mehr vom Ausgangsstapel auf den Zielstapel bringen.

Wir führen folgende Bezeichnungen ein: $V_l := \{v : d^-_{G_1^l}(v) = 0\}$ und $U_l := \{v : d^-_{G_1^l}(v) > 0\}$. Die Stapelelemente aus V_l sind also diejenigen auf dem Ausgangsstapelplatz, auf die sofort zugegriffen werden kann. Da nun kein Knoten aus V_l mehr auf den Zielstapelplatz gebracht werden kann, gilt:

$$\forall v \in V_l : \exists u \in U_l \text{ so daß in } G_Z \text{ ein Pfad von } u \text{ nach } v \text{ führt.}$$

D. h. bevor irgendein v aus V_l auf den Zielstapelplatz gelegt werden kann, muß ein u aus U_l dorthin gebracht werden. Andererseits wissen wir nach Definition von U_l:

$$\forall u \in U_l : \exists v \in V_l \text{ so daß in } G_A \text{ ein Pfad von } v \text{ nach } u \text{ führt.}$$

Zu jedem Knoten aus U_l gibt es von einem Knoten aus V_l einen Weg in $G_A \cup G_Z$. Ebenso gibt es zu jedem Knoten aus V_l von einem Knoten aus U_l einen Weg in $G_A \cup G_Z$. Da U_l und V_l aber endliche Mengen sind, muß $G_A \cup G_Z$ einen Kreis enthalten. Das widerspricht jedoch der Voraussetzung, damit ist die Annahme falsch und Q ist lösbar. □

Es ergibt sich die formale Notation für unser Stapelproblem.

Definition 2.1 Wir definieren: SP:=$\{ Q = (G_A, G_Z, k)$ lösbares verzweigtes Stapelproblem$\}$.

Satz 2.3 (SCHMIEDEL[3]) *Die Entscheidung, ob für ein gegebenes Stapelproblem Q, $Q \in SP$ gilt, ist NP-vollständig.*

3 Lösungsstrategien

3.1 Obere und untere Schranken

Durch Satz 2.2 haben wir erfahren, daß Kreise in der Vereinigung aus Ausgangs- und Zielgraph dafür verantwortlich sind, daß Elemente auf Hilfsstapelplätze gebracht werden müssen. Wenn wir eine solche Stapelkonfiguration betrachten, in der kein Element mehr vom Ausgangsstapelplatz direkt in den Zielstapelplatz eingestapelt werden kann, dann gibt es für jedes Element v des Ausgangsstapelplatzes auf das wir zugreifen können einen Kreis C in der Vereinigung von Ausgangs- und Zielgraph, in dem es enthalten ist: $v \in V(C)$. Darüberhinaus gibt es im Ausgangsgraphen einen Nachbarn u von v, so daß diese Kante (v, u) des Ausgangsstapels in C enthalten ist: $(v, u) \in A(C)$.

Mit dem Ausstapeln solcher Elemente v des Ausgangsstapels auf Hilfsstapelplätze werden die Kreise durch das Entfernen dieser Kanten (v, u) aufgeschnitten.

Sehen wir uns nun die Struktur dieser Kreise etwas genauer an: Da sowohl G_A als auch G_Z kreisfrei sind, muß jeder Kreis C in $G_A \cup G_Z$ Kanten aus G_A und auch aus G_Z enthalten. In einem solchen Kreis C finden sich demnach abwechselnd Pfade aus G_A und G_Z:

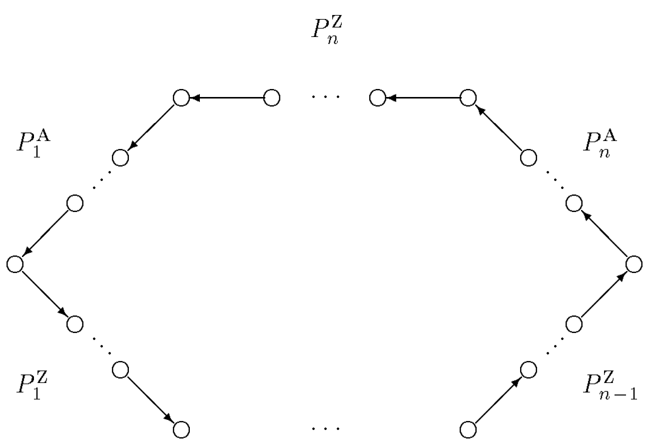


Abbildung 1:

Um ein verzweigtes Stapelproblem $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$, für das die Vereinigung $G_\mathrm{A} \cup G_\mathrm{Z}$ nicht kreisfrei ist, lösen zu können, muß jeder derartige Kreis (siehe Abbildung 1) aufgeschnitten werden. Dieses Aufschneiden kann nur durch das Ausstapeln eines Elementes des Ausgangsstapels auf einen Zwischenstapelplatz realisiert werden. Es ist demnach in geschickter Weise für jeden solchen Kreis ein Pfad P_i^A zu wählen, dessen Startknoten v ($d^-_{P_i^\mathrm{A}}(v) = 0$) dann auf einen Hilfsstapelplatz gebracht wird. Dabei kommen natürlich nur solche Knoten v in Frage, die während des Lösungsprozesses auf der Spitze des Ausgangsstapels liegen, da auf darunterliegende Knoten erst später zugegriffen werden kann.

Schmiedel[3] fand hierzu ein wesentliches Kriterium:
Besteht ein Kreis statt wie in Abbildung 1 jeweils nur aus einem Pfad P^A aus G_A und P^Z aus G_Z, so kommt nur ein Knoten v (nämlich der Startknoten von P^A) zum Aufschneiden des Kreises in Frage. Von diesem Knoten v wissen wir, daß er auf einen Zwischenstapelplatz gebracht werden *muß*.

Lemma 3.1 Schmiedel[3] Ist $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$ ein lösbares, verzweigtes Stapelproblem und existiert ein Kreis $C = [a_1, a_2, \ldots, a_x, b_1, b_2, \ldots, b_y, a_1]$ in $G_\mathrm{A} \cup G_\mathrm{Z}$ für den

$$\begin{array}{ll} & (a_i, a_{i+1}),\ (a_x, b_1) \in A(G_\mathrm{A}) \text{ für } i = 1, \ldots, x-1 \\ \text{und} & (b_i, b_{i+1}),\ (b_y, a_1) \in A(G_\mathrm{Z}) \text{ für } i = 1, \ldots, y-1 \end{array}$$

gilt, dann folgt für jede l-Umstapelfolge $\mathcal{S}$ von Q: $a_1 \in M^l$.

Definition 3.1 Für einen gerichteten Graphen $G = (V, A)$ heißt ein Teilgraph $G' \subseteq G$ genau dann *stark zusammenhängende Komponente* von G, wenn es für alle Knotenpaare $x, y \in V(G')$ einen Weg von x nach y innerhalb von G' gibt. G' heißt *maximal*, wenn alle Teilgraphen $G'' \subseteq G$ mit $G' \subset G''$ keine stark zusammenhängenden Komponenten von G sind.

Satz 3.2 *Es sei $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$ ein verzweigtes lösbares Stapelproblem, dabei sei k die kleinste Anzahl an Hilfsstapelplätzen, für die Q lösbar ist. c sei die Anzahl der Kreise in $G_\mathrm{A} \cup G_\mathrm{Z}$, dc die Anzahl knotendisjunkter Kreise in $G_\mathrm{A} \cup G_\mathrm{Z}$ und $mscc$ sei die Anzahl maximaler stark zusammenhängender Komponenten in $G_\mathrm{A} \cup G_\mathrm{Z}$. Dann gilt*

$$c \geq k \geq dc \geq mscc.$$

Beweis: Da es in einer stark zusammenhängenden Komponente mehrere knotendisjunkte Kreise geben kann, jedoch ein Kreis nicht in mehreren stark zusammenhängenden Komponenten liegen kann, gilt $dc \geq mscc$.

Angenommen, es gilt $k < dc$. Nach Lemma 2.1 kann durch das Entfernen von k Knoten aus G_A $G_\mathrm{A} \cup G_\mathrm{Z}$ kreisfrei gemacht werden. Durch das Entfernen von k Knoten können aber nur k knotendisjunkte Kreise aufgetrennt werden. Somit gilt $k \geq dc$.

Angenommen, es sei $c < k$. Dann könnte durch das Ausstapeln von c Knoten das Stapelproblem Q gelöst werden – k sollte aber die kleinste Anzahl an auszustapelnden Elementen sein, mit denen Q lösbar ist. Daher gilt $c \geq k$. □

Aus der Argumentation zum Beweis von Satz 3.2 läßt sich sofort schlußfolgern, daß es ausreicht, die maximal stark zusammenhängenden Komponenten eines Stapelproblems Q zu betrachten, um dessen Lösbarkeit zu bestimmen. Für einen Lösungsalgorithmus ist es daher sinnvoll, ein Stapelproblem Q in Teilprobleme $Q^1, \ldots, Q^l$ zu zerlegen, die sich bei der Bestimmung der maximalen stark zusammenhängenden Komponenten ergeben.

Definition 3.2 Es sei $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$ ein verzweigtes Stapelproblem. Die Menge der Stapelprobleme $\mathcal{Q} := \{Q^i = (G^i_\mathrm{A}, G^i_\mathrm{Z}, k^i) : i = 1, \ldots, l\}$ heißt genau dann *scc-Dekomposition von* Q, wenn $G^i_\mathrm{A} \cup G^i_\mathrm{Z}$ für alle $i \in \{1, \ldots, l\}$ maximal stark zusammenhängende Komponente von $G_\mathrm{A} \cup G_\mathrm{Z}$ ist.

Lemma 3.3 Es sei $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$ ein verzweigtes Stapelproblem und $\mathcal{Q} := \{Q^i = (G^i_\mathrm{A}, G^i_\mathrm{Z}, k^i) : i = 1, \ldots, l\}$ eine scc-Dekomposition von Q so daß für alle $Q^i = (G^i_\mathrm{A}, G^i_\mathrm{Z}, k^i)$ gilt: k^i ist minimal, d. h. Q^i ist lösbar, $Q'^i = (G^i_\mathrm{A}, G^i_\mathrm{Z}, k^i - 1)$ jedoch nicht. Dann gilt:

$$Q \text{ ist genau dann lösbar, wenn } k \geq \sum_{i=1}^{l} k^i \text{ gilt.}$$

Beweis: („⇒“) $G_\mathrm{A} \cup G_\mathrm{Z}$ besteht aus den maximal stark zusammenhängenden Komponenten $G^1_\mathrm{A} \cup G^1_\mathrm{Z}, \ldots, G^l_\mathrm{A} \cup G^l_\mathrm{Z}$, wobei zur Lösung von Q aus der Komponente $G^i_\mathrm{A} \cup G^i_\mathrm{Z}$ genau c^i Knoten ausgestapelt werden ($i = 1, \ldots, l$). Insgesamt werden k Elemente ausgestapelt, d. h. $\sum_{i=1}^{l} c^i = k$. Angenommen, Q ist lösbar und es gilt $k < \sum_{i=1}^{l} k^i$. Somit gilt: $\sum_{i=1}^{l} c^i < \sum_{i=1}^{l} k^i$. Deshalb existiert wenigstens eine Komponente $G^j_\mathrm{A} \cup G^j_\mathrm{Z}$, für die c^j Knoten $v_1, \ldots, v_{c_j}$ ausgestapelt werden und für die $Q^j = (G^j_\mathrm{A}, G^j_\mathrm{Z}, c^j)$ mit $c^j < k^j$ gilt und die lösbar ist, da das gesamte Problem Q lösbar ist. Das widerspricht jedoch der Voraussetzung, daß k^j minimal ist. Aus diesem Grund muß die Annahme falsch sein.

(„⇐“) Es gelte $k \geq \sum_{i=1}^{l} k^i$. Dann ist Q lösbar, denn wenn jede maximal stark zusammenhängende Komponente für sich gelöst wird, wird auch das gesamte Problem gelöst, da es keine Kreise geben kann, die durch mehrere maximal stark zusammenhängende Komponenten verlaufen. Damit werden zur Lösung nur $\sum_{i=1}^{l} k^i \leq k$ Hilfsstapelplätze zur Lösung von Q benötigt.□

Es ergeben sich zwei Ansätze, die zur algorithmischen Bearbeitung des Problems herangezogen werden können. Einerseits kann das Problem exakt durch vollständige Enumeration gelöst werden, was aufgrund der NP-Vollständigkeit sehr lange dauern wird. Andererseits ist die Bearbeitung durch ein heuristisches Verfahren denkbar, welches wesentlich schneller – insbesondere in Polynomialzeit – arbeiten kann, jedoch im allgemeinen nur eine Näherungslösung liefert.

In den folgenden Abschnitten sollen Algorithmen zu diesen Lösungsansätzen erläutert und die Ergebnisse ihrer Implementierungen vorgestellt werden.

3.2 Branching & Bounding

Um algorithmisch die kleinste notwendige Anzahl an Hilfsstapelplätzen zu bestimmen, bedienen wir uns einer simplen Technik: Wir simulieren einfach das schrittweise Umstapeln von Elementen des Ausgangsstapels auf den Zielstapelplatz. Immer dann, wenn kein Element der Spitze des Ausgangsstapelplatzes auf den Zielstapelplatz gelegt werden kann, benötigen wir einen Hilfsstapelplatz. Nach Lemma 3.3 reicht es aus, alle Stapelprobleme Q^i der scc-Dekomposition von Q zu betrachten. Die kleinste nötige Anzahl an Hilfsstapelplätzen, die zur Lösung von Q benötigt werden, ergibt sich dann aus der Summe der Lösungen der Q^i. Wir zerlegen demnach das Problem $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$ in die Teilprobleme $Q^i = (G^i_\mathrm{A}, G^i_\mathrm{Z}, k^i)$ indem wir die stark zusammenhängenden Komponenten der Vereinigung $G_\mathrm{A} \cup G_\mathrm{Z}$ bilden.

Nachdem wir getestet haben, daß die zu untersuchende Komponente Q^i nicht trivial ist – nicht nur aus einem isolierten Knoten besteht – überprüfen wir, ob sich für Q^i ein Kreis finden läßt, der ein Element v der Spitze des Ausgangsstapelplatzes enthält, so daß Lemma 3.1 erfüllt ist. Sollte das der Fall sein, so stapeln wir v auf einen Hilfsstapelplatz aus. Durch das Entfernen von v kann es passieren, das die stark zusammenhängende Komponente zerfällt. An dieser Stelle haben wir nun die Möglichkeit die zerfallene Komponente wieder in stark zusammenhängende Komponenten zu zerlegen oder aber nach weiteren Knoten v zu suchen, die Lemma 3.1 erfüllen. Für unsere Untersuchung wurde die zweite Möglichkeit gewählt: Durch das Ausstapeln von v entsteht nun auf dem Ausgangsstapelplatz eine neue Spitze. Wir können nun nach Knoten auf dieser Spitze suchen, die sich direkt auf den Zielstapelplatz einstapeln lassen oder Lemma 3.1 erfüllen. Diese Schritte wiederholen wir solange, bis sich keine solchen Knoten mehr finden lassen. An diesem Punkt angelangt, müssen wir einen Knoten der Spitze auf einen Hilfsstapelplatz bringen – da wir aber nicht wissen welchen, bleibt uns nur, nach und nach jeden dieser Knoten auszustapeln, den verbleibenden Rest von Q^i rekursiv zu lösen und uns die entsprechende Lösung – Anzahl der benötigten Hilfsstapelplätze – zu merken. Dieses einfache Backtracking-Verfahren wird zum Branch-and-Bound Algorithmus, indem wir uns das jeweils beste erzielte Ergebnis merken und die Berechnung für einen ausgestapelten Knoten sofort abbrechen, sobald die in dieser Verzweigung benötigte Anzahl an Hilfsstapelplätzen größer wird, als dieses bisher beste Ergebnis. Als Resultat erhalten wir die minimal notwendige Anzahl an Hilfsstapelplätzen, die zur Lösung von Q^i benötigt wird.

3.3 Heuristische Lösungsverfahren

Das untersuchte heuristische Verfahren verläuft wie der Branch-and-Bound Algorithmus. Jedoch an der Stelle, an der weder Knoten der Spitze existieren, die Lemma 3.1 erfüllen, noch Knoten direkt auf den Zielstapelplatz gebracht werden können, muß hier genau ein Knoten ausgewählt werden, der auf den Hilfsstapelplatz gebracht wird. Da diese Entscheidung welcher Knoten ausgestapelt wird, nicht wieder korrigiert werden kann, müssen wir diese Wahl von einem definierten Kriterium abhängig machen. Dazu bewerten wir alle in Frage kommenden Knoten (die Knoten der Spitze des Ausgangsstapelplatzes) durch eine Funktion f:

Definition 3.3 Es sei $Q = (G_\mathrm{A}, G_\mathrm{Z}, k)$ ein verzweigtes Stapelproblem und $Q^i = (G^i_\mathrm{A}, G^i_\mathrm{Z}, k^i)$ ein Stapelproblem aus der scc-Dekomposition von Q. Weiterhin sei $\mathrm{SPITZE}(G^i_\mathrm{A}) = \{v_1, \ldots, v_l\}$. Für jeden Knoten v_t der Spitze von G^i_A seien $C_{t,1}, \ldots, C_{t,r_t}$ alle diejenigen Kreise in $G^i_\mathrm{A} \cup G^i_\mathrm{Z}$, für die $v_t \in V(C_{t,j})$ für $j \in \{1, \ldots, r_t\}$ gelte. Dann ist f wie folgt definiert:

$$f(v_t) := |\{v_1, \ldots, v_l\} \cap \bigcup_{j=1}^{r_t} V(C_{t,j})|.$$

Demnach bestimmt $f(v_t)$ die Anzahl derjenigen Knoten der Spitze, die in Kreisen in $G^i_\mathrm{A} \cup G^i_\mathrm{Z}$ enthalten sind in denen auch v_t auftritt. Derjenige Knoten mit dem größten Funktionswert wird schließlich auf einen Hilfsstapelplatz gebracht. Der Rest der betrachteten stark zusammenhängenden Komponente Q^i wird nun separat (rekursiv) untersucht. Die Qualität und Geschwindigkeit eines solchen heuristischen Verfahrens hängt offensichtlich von der Güte der gewählten Bewertungsfunktion f ab.

3.4 Erfahrungen mit den Algorithmen

Zur Durchführung von Experimenten sind zunächst Beispielgraphen zu generieren, die Ausgangs- und Zielstapel eines Stapelproblems repräsentieren. Die Knoten eines gerichteten, kreisfreien Graphen können in gewisse Ebenen eingeordnet werden, zwei Knoten liegen genau dann in einer Ebene, wenn die Länge der kürzesten Pfade, die von der Spitze des Graphen zum Knoten führen, gleich ist. Die *durchschnittliche Breite* eines solchen Graphen ergibt sich aus dem arithmetischen Mittel der Knotenanzahl in diesen Ebenen des Graphen. Die *Höhe* eines gerichteten, kreisfreien Graphen G ist die Anzahl dieser Ebenen – die Länge des längsten Pfades in G.

Obwohl das Generieren solcher zufälligen Graphen kein triviales Problem ist, soll hier nicht näher darauf eingegangen werden. Im wesentlichen verwenden wir die Methode, die Hempel und Vasilkov in [2] erläutern. Dabei lassen sich unter anderem die durchschnittliche Breite und Höhe eines zu generierenden Graphen angeben sowie dessen Kantendichte – d. h. die Wahrscheinlichkeit, mit der mögliche Kanten im Graphen existieren.

Bei den ersten Experimenten fiel auf, daß ein sehr hoher Anteil der untersuchten Beispiele in Polynomialzeit lösbar ist. Algorithmisch bedeutet das für diese Beispiele: Im Branch-and-Bound Verfahren ist keine Verzweigung nötig, im heuristischen Verfahren müssen keine Knoten durch f bewertet werden. Schritt für Schritt finden sich immer wieder Knoten, die entweder den Voraussetzungen von Lemma 3.1 genügen oder sich direkt auf den Zielstapelplatz einstapeln lassen, solange, bis der Ausgangsstapel vollständig abgearbeitet wurde. Das Branch-und-Bound Verfahren und das heuristische Verfahren liefern in diesem Fall die gleichen Ergebnisse. In Tabelle 1 wird der Anteil derjenigen Beispiele (gemessen an allen untersuchten Beispielen) angegeben, welche ohne Verzweigung lösbar sind. Die betrachteten Graphen haben maximal 100 Knoten, die Kantendichte beträgt 25 %. Es wurde die Abhängigkeit von der Struktur der Graphen – Verhältnis von Höhe zu Breite – untersucht.

Kantendichte: 25 %, $|V| \leq 100$
r_A=Höhe/Breite im Ausgangsstapel
r_Z=Höhe/Breite im Zielstapel

	$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$
$r_A = 10$	97%	94%	91%
$r_A = 1$	96%	90%	83%
$r_A = 0.1$	90%	80%	63%

Tabelle 1: Anteil der Beispiele, die in Polynomialzeit lösbar sind, gemessen an allen untersuchten Stapelproblemen mit den angegebenen Parametern.

Dabei scheint die Struktur der Stapel eine wesentliche Rolle zu spielen, so daß auch in den folgenden Experimenten diese Struktur als Parameter variiert wurden. Dabei muß erwähnt werden, das im weiteren Text nur noch über Beispiele berichtet wird, für deren Lösung im Branch-and-Bound Algorithmus wirklich verzweigt werden muß.

3.4.1 Qualität der Heuristik

In zwei Versuchsreihen wurden die Ergebnisse des heuristischen Verfahrens bewertet: Einmal betrug die Kantendichte wieder 25 %, das andere Mal nur 7 %. Durch diese geringere Kantendichte mußte aus Rechenzeitgründen die Größe der Probleme auf maximal 50 Knoten beschränkt werden. In Tabelle 2 wird angegeben, in wieviel Prozent der untersuchten Fälle die Heuristik das Optimum (O) und eine „gute“ (G) Lösung (Abweichung vom Optimum um maximal einen Knoten) liefert.

3.4.2 Rechenzeit der Algorithmen im Vergleich

Der Vergleich der benötigten Rechenzeiten zwischen dem Branch-and-Bound und dem heuristischen Verfahren fällt recht eigentümlich aus: Die betrachteten Beispiele lassen sich in zwei Gruppen einteilen: In der ersten Gruppe – der überwiegenden Anzahl der Beispiele – gibt es kaum einen Unterschied zwischen den beiden Verfahren, die Rechenzeit betreffend. In der zweiten – deutlich kleineren – Gruppe steigt jedoch die vom Branch-and-Bound Verfahren benötigte Rechenzeit derart stark an, daß die Heuristik um ein Vielfaches schneller ist.

Tabelle 3 zeigt den Anteil „schwieriger“ Beispiele an den untersuchten Stapelproblemen. Schwierig soll in diesem Fall bedeuten, daß die zur Lösung eines Stapelproblems Q benötigte Zeit $T_{\mathrm{Branch-and-Bound}}(Q)$ mehr als zehnmal größer ist, als die Zeit $T_{\mathrm{Heuristik}}(Q)$, die der heuristische Algorithmus benötigt:

$$T_{\mathrm{Branch-and-Bound}}(Q) > 10 \cdot T_{\mathrm{Heuristik}}(Q).$$

Kantendichte: 25 %, $|V| \leq 100$
r_A=Höhe/Breite im Ausgangsstapel
r_Z=Höhe/Breite im Zielstapel

	$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$
$r_A = 10$	O: 66% G: 93%	O: 73% G: 97%	O: 71% G: 94%
$r_A = 1$	O: 68% G: 94%	O: 60% G: 89%	O: 66% G: 91%
$r_A = 0.1$	O: 61% G: 93%	O: 53% G: 85%	O: 52% G: 79%

Kantendichte: 7 %, $|V| \leq 50$
r_A=Höhe/Breite im Ausgangsstapel
r_Z=Höhe/Breite im Zielstapel

	$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$
$r_A = 10$	O: 43% G: 76%	O: 39% G: 80%	O: 66% G: 89%
$r_A = 1$	O: 47% G: 80%	O: 50% G: 70%	O: 68% G: 82%
$r_A = 0.1$	O: 61% G: 79%	O: 60% G: 85%	O: 72% G: 91%

Tabelle 2: Qualität der Heuristik für verschiedene Kantendichten

Unter diesen Beispielen ist jedoch auch der Faktor 1000 keine Seltenheit.

Kantendichte: 25 %, $\lvert V\rvert \leq 100$ r_{A}=Höhe/Breite im Ausgangsstapel r_{Z}=Höhe/Breite im Zielstapel			
	$r_{\mathrm{Z}} = 10$	$r_{\mathrm{Z}} = 1$	$r_{\mathrm{Z}} = 0.1$
$r_{\mathrm{A}} = 10$	0%	2%	0%
$r_{\mathrm{A}} = 1$	4%	2%	1%
$r_{\mathrm{A}} = 0.1$	9%	11%	13%

Kantendichte: 7 %, $\lvert V\rvert \leq 50$ r_{A}=Höhe/Breite im Ausgangsstapel r_{Z}=Höhe/Breite im Zielstapel			
	$r_{\mathrm{Z}} = 10$	$r_{\mathrm{Z}} = 1$	$r_{\mathrm{Z}} = 0.1$
$r_{\mathrm{A}} = 10$	10%	8%	9%
$r_{\mathrm{A}} = 1$	7%	11%	9%
$r_{\mathrm{A}} = 0.1$	9%	2%	2%

Tabelle 3: Anteil „schwieriger“ Stapelprobleme für verschiedene Kantendichten

4 Ausblick

Die Auswertung der Experimente ergab, daß Stapelprobleme mit sehr hoher Kantendichte (25%) leichter zu lösen sind als Probleme mit geringer Dichte. Hempel und Vasilkov [2] berichten ebenfalls von steigender Schwierigkeit bei abnehmender Kantendichte des Graphen. Hierfür ist die geringe Wahrscheinlichkeit für die Existenz von Kreisen verantwortlich zu machen, die Lemma 3.1 erfüllen – je geringer die Kantendichte, desto größer werden die Kreise in $G_{\mathrm{A}} \cup G_{\mathrm{Z}}$.

Gleichzeitig muß festgestellt werden, daß die Struktur der Graphen keinen großen Einfluß auf Güte und Geschwindigkeit der Heuristik haben, bzw. dieser Einfluß nicht konstant ist. Die bisherigen Versuche lassen vermuten, daß nicht die Struktur der Stapel, sondern die Kantendichte der Graphen wesentlichen Einfluß auf die Schwierigkeit des Problems hat. Aus [2] entnehmen wir, daß nach der Zerlegung des Problems in stark zusammenhängende Komponenten häufig nur eine Anzahl isolierter Knoten sowie eine sehr große Komponente vorliegen.

Eine weitere interessante Möglichkeit der Variation besteht in der Änderung der Bewertungsfunktion f. Hier könnte für ausgewählte Beispiele untersucht werden, inwieweit mit einfacheren (schnelleren) Kriterien ähnlich gute Resultate erzielt werden können. Abschließend können wir sagen, daß für Probleme mit den betrachteten Eigenschaften ($\lvert V\rvert < 100$, Kantendichte=25% bzw. $\lvert V\rvert < 50$, Kantendichte=7%) eine Kombination der beiden vorgestellten Verfahren sehr sinnvoll ist: Erst wenn das Branch-and-Bound Verfahren in einem vorgegebenen Zeitintervall keine Lösung liefert, sollte das heuristische Verfahren auf das Problem angesetzt werden.

Literatur

[1] L. Hempel: *Mathematische Modellierung von Stapelproblemen.* HAB-Weimar (1994), unveröffentlichter Vortrag.

[2] L. Hempel and D. Vasilkov: *On the Computational Aspects of the Pile Problem.* Preprint der Bauhaus-Universität Weimar, in Vorbereitung.

[3] R. Schmiedel: *persönliche Mitteilung.*