Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

Datum: 20.6.2019

Übungsblatt mit Lösungen 10

Aufgabe T35

Manche Paare von Städten bilden Partnerschaften mit anderen Städten (Aachen zum Beispiel mit Liège, Toledo und weiteren). Diese Beziehungen lassen sich gut durch einen Graphen modellieren.

Nun soll im *Internationalen Jahr der Städtefreundschaften* je eine Feier zwischen allen befreundeten Städtepaaren durchgeführt werden. Wir gehen der Einfachheit halber davon aus, dass alle Feiern dasselbe kosten. Eine Feier zwischen zwei Städten kann entweder von der einen oder der anderen Stadt finanziert werden. Jede Stadt hat ein Budget, welches besagt für wie viele Feiern sie Geld spenden kann. Dies soll stets eine ganze Zahl sein.

Ihr Problem ist es nun, einen Plan zu finden, der besagt welche Stadt für welche Feiern zahlen soll, ohne dass Budgets überschritten werden. Können Sie dies durch Lösen eines Flussproblems lösen?

Lösungsvorschlag

Wir konstruieren folgendes Flussnetzwerk. Jede Stadt ist ein Knoten. Gibt es eine Partnerschaft zwischen Städten A und B, dann fügen wir einen neuen Knoten P_{AB} hinzu, sowie je eine gerichtete Kante von A bzw. B zu P_{AB} mit einer Kapazität von 1. Von P_{AB} zur Senke fügen wir eine weitere Kante mit Kapazität 1 ein. Nun folgen noch Kanten von der Quelle zu jedem Knoten, der eine Stadt darstellt. Die Kapazität dieser Kanten sind einfach so groß wie das Budget der Stadt.

Es gibt jetzt eine einfache Bijektion zwischen ganzzahligen Flüssen in diesem Netzwerk und einem gültigen Plan für die Verteilung der Festivitätenfinanzierung: Stadt A ist für die Finanzierung der Party mit B genau dann verantwortlich, wenn ein Fluss von 1 von A nach P_{AB} fließt. Es ist leicht einzusehen, dass so einem Fluss eine gültige Finanzierung entspricht und umgekehrt. Insbesondere entspricht die maximale Anzahl finanzierbarer Feiern und die dazu nötigen Finanzierungsverhältnisse dem maximalen Fluss durch das Netzwerk.

Aufgabe T36

Die internationale Raumstation ISS steht auch Weltraumtouristen offen. Sie sollen nun entscheiden, welche Touristen Sie mitnehmen wollen, um möglichst viel Geld zu verdienen.

Gegeben sind Kandidaten $K_1, \ldots, K_n$, welche jeweils bereit sind, $k_1, \ldots, k_n$ US-Dollar zu zahlen. Allerdings sind sie anspruchsvoll und erwarten auf der ISS auch ein Unterhaltungsprogramm (der Erstbesucher Cameron wollte zum Beispiel einen Weltraumspaziergang machen). Zu diesem Zweck stehen eine Menge „Spielzeuge“ $Z_1, \ldots, Z_m$ zur Verfügung. Bei der Mitnahme eines Spielzeugs zur ISS entstehen allerdings jeweils Kosten $z_1, \ldots, z_m$. Der Kandidat K_i ist nur bereit zu zahlen, wenn die Spielzeuge $R_i \subseteq \{Z_1, \ldots, Z_m\}$ mitgenommen werden.

Entwerfen Sie einen effizienten Algorithmus, der eine Menge von Kandidaten auswählt, um die Einnahmen (also die gezahlten Gebühren der Touristen minus die Kosten für die Spielzeuge) zu maximieren. Jedes Spielzeug muss nur einmal mitgenommen werden, selbst wenn mehrere es benutzen wollen.

Lösungsvorschlag

Wir konstruieren auf der Knotenmenge $\{s,t\} \cup \{K_1,\dots,K_n\} \cup \{Z_1,\dots,Z_m\}$ das folgende Flussproblem:

1. Von der Quelle s läuft zu jedem Knoten K_i eine Kante mit Kapazität k_i.
2. Für jedes $Z_j \in R_i$ läuft eine Kante mit unendlicher Kapazität von K_i nach Z_j.
3. Von jedem Knoten Z_i läuft eine Kante mit Kapazität z_i in die Senke t.

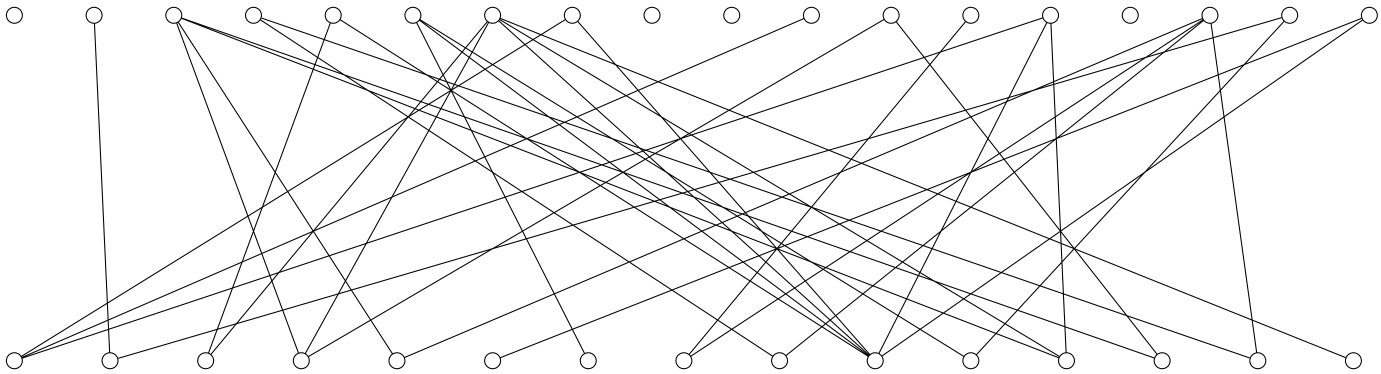
Jeder (nicht unendliche) Schnitt im oben beschriebenen Netzwerk stellt eine Lösung dar: Wir kaufen alle Spielzeuge deren Kante geschnitten wird und nehmen alle Kandidaten mit deren Kante *nicht* geschnitten wird. Da unendliche Kanten nicht im Schnitt liegen, sind alle Wünsche erfüllt.

Sei $M = \sum_{i=1}^{n} k_i$ der maximal mögliche Gewinn. Die gestrichenen Kanten entsprechen jeweils der Differenz zu M (durch Verzicht auf einen Touristen oder Verzicht auf die Ersparnis durch Nichtmitnahme eines Spielzeugs).

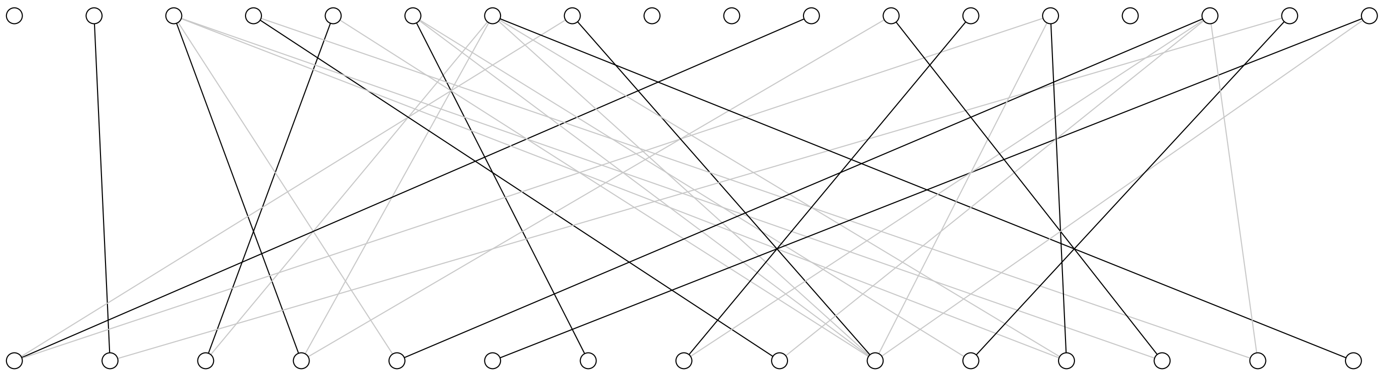
Ein Min-Cut entspricht somit einer Lösung, mit maximalem Gewinn. Ein Min-Cut für dieses Netzwerk ist natürlich endlich, da es endliche Schnitte gibt (beispielsweise die Menge aller Kanten, die in s beginnen).

Aufgabe H28 (8 Punkte)

Finden Sie ein Matching mit maximaler Kardinalität in diesem Graphen. Aus wie vielen Kanten besteht es?



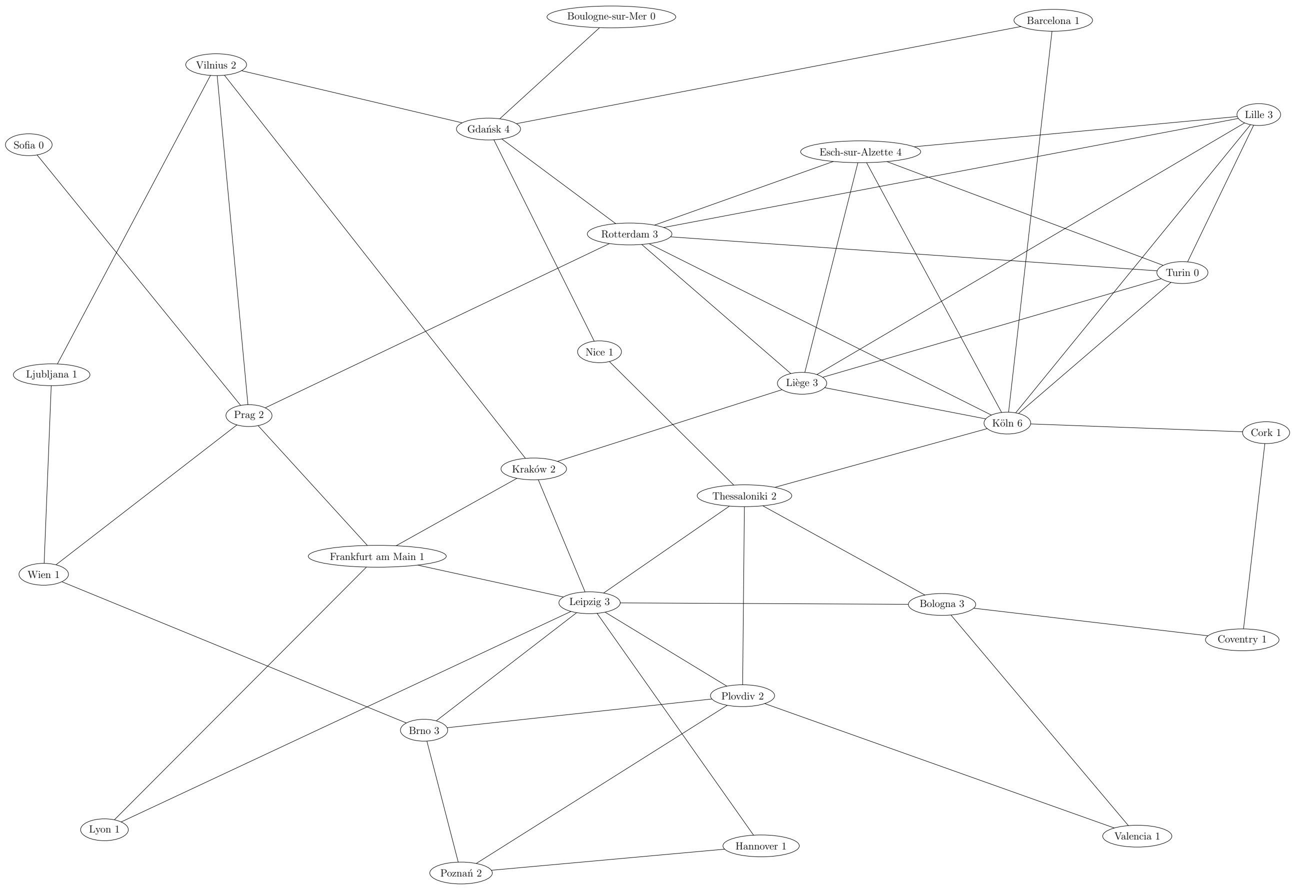
Lösungsvorschlag



Das Matching besteht aus 14 Kanten

Aufgabe H29 (10 Punkte)

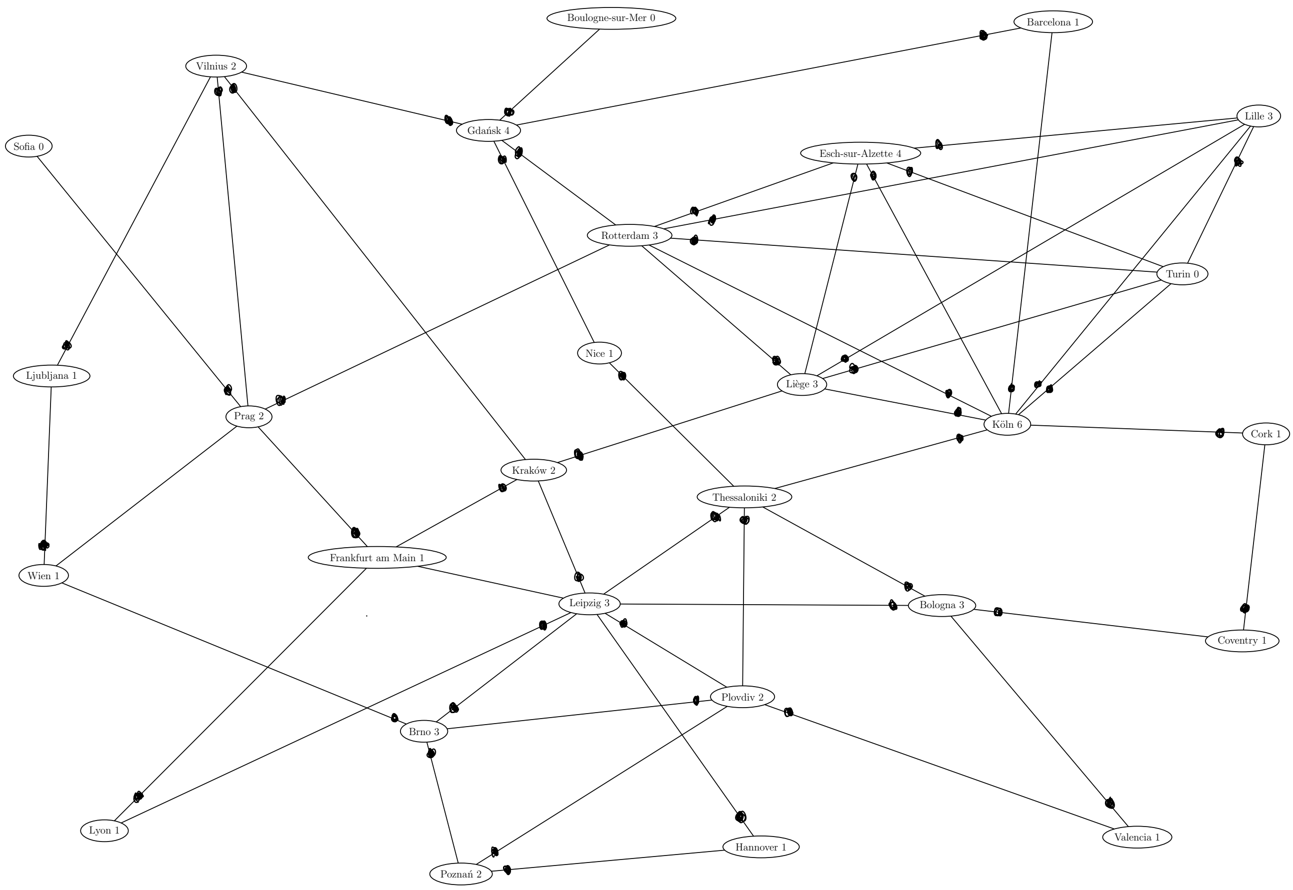
Finden Sie einen Plan, analog zur Aufgabe oben, für das kleinere Städtepartnerschaftsmodell unten. Die Zahlen geben das Budget für jede Stadt wieder. Zeichnen Sie die gefundene Lösung ein, indem Sie für jede Kante markieren, welche der beiden Städte für diese Feier aufkommt.



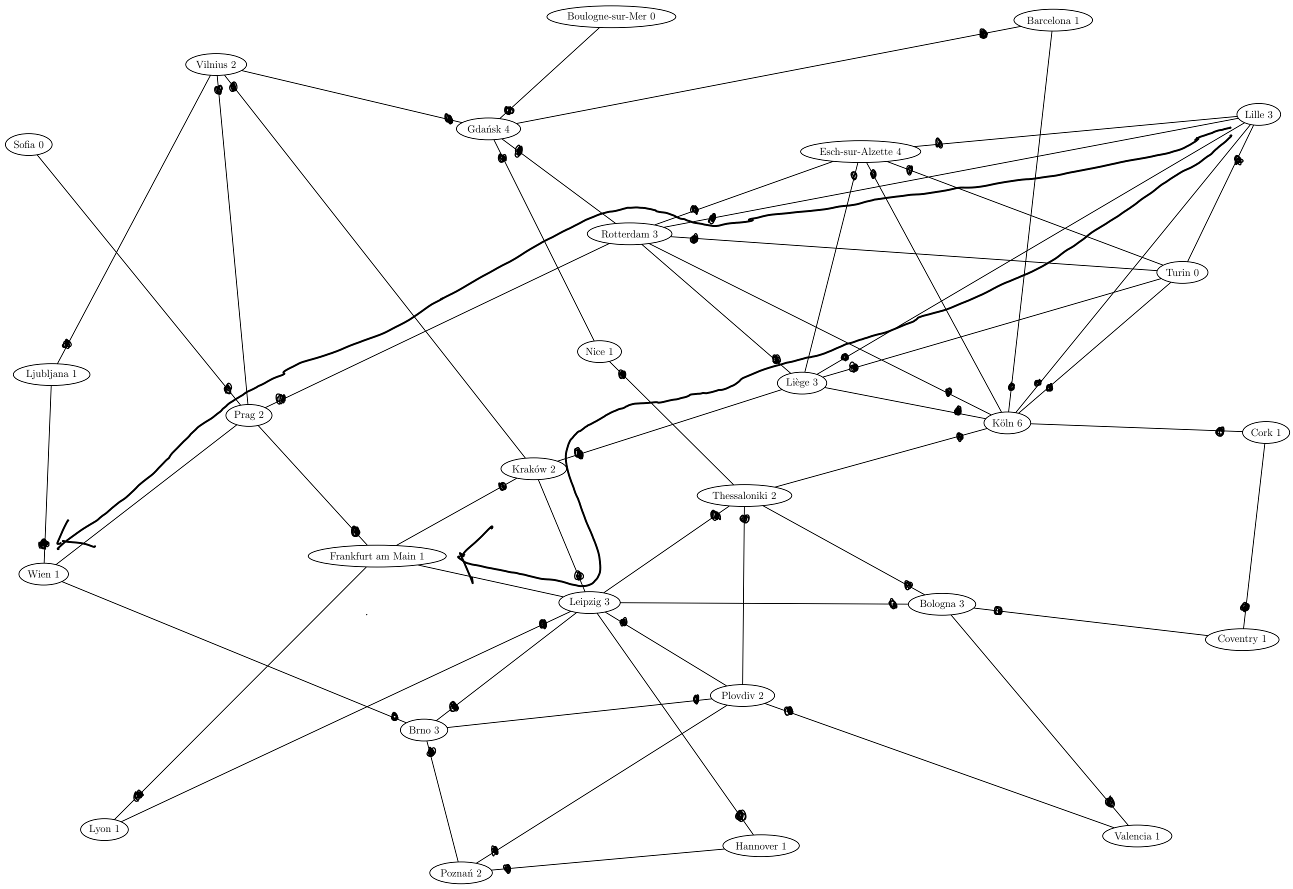
Lösungsvorschlag

Diese Aufgabe lässt sich recht leicht per Hand lösen. Man beginnt einfach die Finanzierung für Partys beliebig festzulegen (indem man mit Bleistift die verantwortliche Stadt an der Kante markiert) und achtet lediglich darauf, dass die Budgets nicht überzogen werden. Irgendwann kann man so nicht weitermachen und es verbleiben unfinanzierte Feste bestehen (aber wahrscheinlich nur sehr wenige).

In unserem Beispiel könnte der vorläufige Plan so aussehen.



Leider sind die Feiern zwischen Wien und Prag, sowie zwischen Frankfurt und Leipzig noch nicht finanziert. Jetzt stellen wir uns das zugrundeliegende Residualnetzwerk vor und finden einen augmentierenden Pfad. Dies entspricht in diesem Fall einem Pfad entlang dessen die Finanzierungsverantwortlichkeit „umgeschoben" wird. Dadurch wird der Fluss um eins erhöht.



Hier können augmentierende Pfade von Lille (das noch ein Budget von zwei übrig behalten hatte) zu Wien und Frankfurt finden. Nachdem wir den Fluss entlang dieser Pfade augmentiert haben, sind alle Partys finanziert.

Aufgabe H30 (12 Punkte)

Die Produktionsaufträge $J_1, J_2, \ldots, J_n$ eines Unternehmens benötigen verschiedene Maschinen $M_1, M_2, \ldots, M_m$, wobei ein Auftrag auch mehrere Maschinen belegen kann. Ein Auftrag bringt natürlich einen gewissen Geldbetrag ein. Die Maschinen können entweder gekauft werden — diese Kosten entstehen dann nur einmal und jede weitere Nutzung ist Kostenfrei — oder gemietet werden. Letzteres kostet pro Auftrag einen gewissen Betrag (dieser Betrag variiert also von Auftrag zu Auftrag!). Die dritte Tabelle enthält die Mietkosten der Maschinen für die jeweiligen Aufträge, eine leere Zelle bedeutet, dass diese Maschine für diesen Auftrag nicht benötigt wird, ansonsten benötigt man *alle* anderen Maschinen für diesen Auftrag. Zum Beispiel kostet Auftrag J_1 auf der Maschine M_1 30 Geldeinheiten (vorausgesetzt, M_1 wurde nicht gekauft).

Für so ein Szenario mit gegebenen Aufträgen, Maschinen und Kosten soll eine gewinnmaximierende Menge von Aufträgen mit entsprechender Zuteilung von Maschinen berechnet werden. Beschreiben Sie ein allgemeines Verfahren, um dieses Problem möglichst effizient zu lösen. Begründen Sie, wieso Ihr Verfahren korrekt funktioniert.

Benutzen Sie Ihr Verfahren, um eine optimale Lösung für das gegebene Szenario zu berechnen.

Auftrag	Zahlung
J_1	80
J_2	80
J_3	120

Maschine	Kaufpreis
M_1	62
M_2	80
M_3	100
M_4	32

	M_1	M_2	M_3	M_4
J_1	30		50	
J_2		60	22	
J_3	30	30	30	30

Lösungsvorschlag

Die Knotenmenge des Flussnetzwerks ist $\{s, t\} \cup \{J_1, \ldots, J_n\} \cup \{M_1, \ldots, M_m\}$. Die Kanten des Netzwerks sind wie folgt:

1. Von der Quelle s läuft zu jedem Knoten J_i eine Kante mit Kapazität j_i, wenn j_i die für einen angenommenen Auftrag J_i erhaltene Zahlung ist.

2. Von einem Auftrag J_i läuft eine Kante zur Maschine M_j, wenn M_j für den Auftrag J_i benötigt wird. Die Kapazität auf dieser Kante sind die Kosten, die für eine einmalige Anmietung der Maschine M_j für den Auftrag J_i entstehen (Tabelle 3).

3. Von jedem Knoten M_i läuft eine Kante mit Kapazität m_i in die Senke t, wenn m_i die Kosten sind, um die Maschine M_i zu kaufen.

Ein Min-Cut (S, T) für dieses Netzwerk ist natürlich endlich und ganzzahlig und kann effizient mit der Methode von Ford und Fulkerson berechnet werden. Wir wählen nun diejenigen Aufträge aus, die in S enthalten sind. Ebenso kaufen wir alle Maschinen in S.

Zur Begründung, warum dieses Verfahren korrekt ist: Die Kanten im Schnitt entsprechen anschaulich einem *Verzicht* auf einen Geldbetrag, welche der Kapazität dieser Kante entspricht. Wird etwa eine Kante $s \to J_i$ für einen Auftrag J_i vom Schnitt (S, T) geschnitten, dann verzichtet man auf die Einnahmen in Höhe von j_i. Der Verzicht auf die Einnahmen kann günstiger sein (und zu einem kleineren Schnitt führen), als wenn man den Job annimmt und dafür viele andere Kanten (für Aufträge oder Maschinen) schneiden muss. Schneidet man analog eine Kante $M_i \to t$, dann muss man den entsprechenden Kaufpreis für eine Maschine bezahlen. Auch dieses kann günstiger sein, als die Maschine für jeden angenommen Job zu mieten. Liegt andererseits eine Maschine M_j in T, dann ist es offenbar günstiger, M_j für jeden angenommenen Auftrag J_i anzumieten; andernfalls könnte man leicht einen kleineren Schnitt konstruieren, indem man M_j nach S verschiebt (und dem Fall entspricht, M_j zu kaufen), was im Widerspruch zur Minimalität des Schnittes steht.