Matching Markets

Ausarbeitung zum Seminar
"Aktuelle Themen der Theoretischen Informatik - Algorithmen"

Marius Hagemann
5732742

14. Oktober 2019

Universität Frankfurt am Main
Fachbereich Informatik

Dr. Annamaria Kovacs

Inhaltsverzeichnis

1 Einleitung

In dieser Ausarbeitung wird untersucht, wann bipartite Graphen ein perfektes Matching besitzen. Weiterhin wird eine Art Markt mit Preisen und Angeboten modelliert, wofür der Graph Begriff leicht verändert wird. Dabei wird untersucht, wann es in diesem Markt eine optimale Zuteilung von "Gütern" zu "Käufern" gibt und wie die Preise der Güter eine solche optimale Zuteilung beeinflussen. Ein wichtiger Begriff dabei ist der markträumende Preis, welcher in Kapitel 4 behandelt wird. Auch wird ein Algorithmus angegeben, welcher markträumende Preise berechnet.

Die Ausarbeitung orientiert sich am Kapitel 10 des Buchs **Networks, Crowds, and Markets** von David Easley und Jon Kleinberg [1].

2 Allgemeine bipartite Graphen

Zu Beginn definieren wir die grundlegenden mathematischen Begriffe. Als wichtigste Grundlage dient hierbei das Konstrukt des Graphen.

Definition 2.1 (Graph).
Ein (ungerichteter) **Graph** $G = (V, E)$ ist ein Tupel bestehend aus einer Knotenmenge V und einer Kantenmenge E. Eine Kante verbindet zwei Knoten miteinander und ist damit eine Menge, aus zwei Knoten. Es gilt: $E \subseteq \{\{u, v\} | u, v \in V, u \neq v\}$.

In dieser Arbeit spielt eine spezielle Klasse von Graphen, bipartite Graphen, eine zentrale Rolle. Bei einem bipartiten Graphen kann man die Knoten in zwei Mengen teilen, sodass alle Kanten nur zwischen den beiden Mengen verlaufen und nicht innerhalb einer Menge. Formal bedeutet dies:

Definition 2.2 (bipartiter Graph).
Ein Graph $G = (V, E)$ heißt **bipartit**, wenn es Teilmengen $V_1 \subset V$ und $V_2 \subset V$ gibt, für die $V_1 \cup V_2 = V$ und $V_1 \cap V_2 = \emptyset$ gilt, sodass für jede Kante $e \in E$ ein $u \in V_1$ und ein $v \in V_2$ existiert, sodass $e = \{u, v\}$ gilt. Die Knotenmengen V_1 und V_2 werden auch als Partitionen bezeichnet.

Ein Beispiel für einen bipartiten Graphen sieht man in Abbildung 2.1. Dabei gilt für die Partitionen: $V_1 = \{v_1, v_2, v_3, v_4\}$ und $V_2 = \{v_5, v_6, v_7, v_8\}$. Man sieht deutlich, dass alle Kanten die Partitionen V_1 und V_2 "kreuzen".

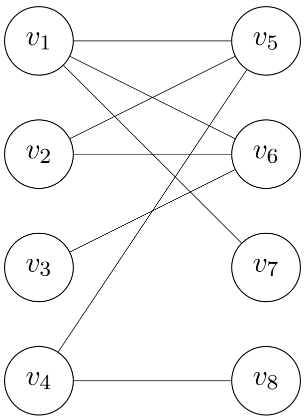

Abbildung 2.1: Beispiel eines bipartiten Graphen

Weiterhin soll in dieser Arbeit der Begriff eines **Matchings** näher behandelt werden.

Definition 2.3 (Matching).
Ein **Matching** $M \subseteq E$ ist eine Kantenmenge, wobei kein Knoten mit mehr als einer Kante aus M inzident ist.

In Abbildung 2.2 sind verschiedene Matchings zu sehen. Die Kanten, die im Matching enthalten sind, sind jeweils hervorgehoben.
Ein Matching ist **nicht-erweiterbar**, wenn – wie der Name schon sagt – man keine weitere Kante zum Matching hinzufügen kann, da sonst die Matching-Eigenschaft verletzt werden würde. Ein **größtmögliches** Matching hat die größtmögliche Anzahl an Kanten, die ein Matching auf dem

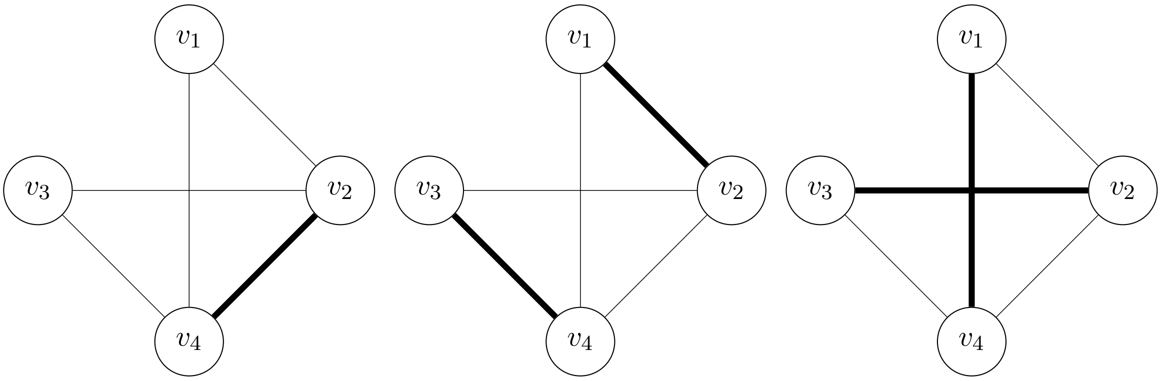

Abbildung 2.2: Verschiedene Matchings auf einem Beispielgraphen

gegebenen Graphen haben kann, und ein **perfektes** Matching ist ein Matching, welches alle Knoten überdeckt. Im obigen Beispiel sind alle drei Matchings nicht erweiterbar, jedoch sind nur die letzten beiden perfekt (und somit auch größtmöglich). Mit anderen Worten:

Definition 2.4 (perfektes Matching)**.**
Ein Matching M heißt **perfekt**, wenn $2 \cdot |M| = |V|$ gilt.

Betrachtet man perfekte Matchings auf bipartiten Graphen, muss offensichtlich $|V_1| = |V_2|$ gelten. Ferner gilt für bipartite Graphen, dass in einem solchen Graphen nur dann ein perfektes Matching existiert, wenn es eine eins-zu-eins Zuordnung der Knoten aus V_1 und V_2 gibt. Mathematisch bedeutet dies, dass es eine bijektive Abbildung $f : V_1 \to V_2$ geben muss, sodass für alle $v \in V_1$: $\{v, f(v)\} \in E$ gilt. Das perfekte Matching ist dann: $M = \{\{v, f(v)\} \in E \mid v \in V_1\}$.

Nun geht es um die Frage, ob ein gegebener bipartiter Graph ein perfektes Matching enthält oder nicht. Gibt es ein perfektes Matching, ist dies leicht zu zeigen: Man muss lediglich eine Kantenmenge angeben und nachweisen, dass diese die Eigenschaft eines perfekten Matchings erfüllt. Gibt es hingegen kein perfektes Matching, ist es auf den ersten Blick nicht einfach, dies zu zeigen. Dabei hilft Satz 2.7, welcher Matching-Theorem, Satz von Hall und König oder auch Heiratssatz [3] genannt wird.
Um diesen Satz zu formulieren werden noch zwei weitere Definitionen benötigt. Zuerst wird die Nachbarschaft einer Knotenmenge definiert.

Definition 2.5 (Nachbarschaft)**.**
Die **Nachbarschaft** $N(S)$ einer Knotenmenge S ist gegeben als:

$$N(S) = \Big\{v \in V \setminus S \mid \exists u \in S : \{v, u\} \in E\Big\}$$

Für bipartite Graphen gilt offensichtlich: Wenn $S \subseteq V_1$, dann ist $N(S) \subseteq V_2$. Die zweite Definition bezieht sich auf das Größenverhältnis zwischen einer Menge und ihrer Nachbarschaft.

Definition 2.6 (blockierende Menge)**.**
Eine Knotenmenge $S \subseteq V_1$ (bzw. $S \subseteq V_2$) heißt **blockierend**, wenn $|S| > |N(S)|$ gilt.

Eine Menge ist also blockierend, wenn sie mehr Elemente hat als ihre Nachbarschaft. Intuitiv ist sofort klar, dass es in dem Fall, wenn eine blockierende Menge existiert, kein perfektes Matching geben kann. Dies gilt, da es bei einem perfekten Matching eine eins-zu-eins Zuordnung der Knoten aus der blockierenden Menge zu den Knoten aus der Nachbarschaft geben muss. Wenn in der Nachbarschaft aber echt weniger Knoten vorhanden sind, ist dies nicht möglich. Somit ist die Nicht-Existenz einer blockierenden Menge notwendig für die Existenz eines perfektes Matchings. Das Matching-Theorem besagt nun, dass die Nicht-Existenz sogar hinreichend ist.

Satz 2.7 (Matching-Theorem).
Hat ein bipartiter Graph kein perfektes Matching, dann muss es eine blockierende Menge geben.

Der Beweis dieses Satzes wird an dieser Stelle ausgelassen, kann aber in Kapitel 10.6 von "Networks, Crowds, and Markets" [1] nachgelesen werden.

Weiterhin lässt sich auch zeigen, dass in dem Fall, wenn ein bipartiter Graph kein perfektes Matching hat, es in beiden Partitionen eine blockierende Menge gibt. Mit Hilfe des Matching-Theorems ist es nun einfacher zu zeigen, dass es auf einem bipartiten Graphen kein perfektes Matching gibt. Es reicht demnach also aus, eine blockierende Menge zu finden. Hat man eine blockierende Menge gefunden, kann der Graph kein perfektes Matching haben.

3 Preise und Bewertungen

Die bisher behandelten bipartiten Graphen sind statisch. Entweder es gibt ein perfektes Matching in einem gegebenen Graphen oder es gibt kein perfektes Matching. Um einen Markt zu simulieren wird jedoch ein dynamischeres Modell benötigt. Dies wird durch einige Veränderungen umgesetzt: Die ”festen” Kanten zwischen den Partitionen werden abgeschafft. Dafür wählt jeder Knoten der Partition V_2 (ab jetzt ”**Käufer**” genannt) zu jedem Knoten aus V_1 eine nichtnegative ganzzahlige **Bewertung**. Dabei drückt die Bewertung aus, wie sehr eine Verbindung ”gewollt” ist. Wenn die Bewertung hoch ist, ist diese Verbindung beliebt, bei niedriger Bewertung ist die Verbindung unbeliebter.
Damit können auch die vorher beschriebenen bipartiten Graphen simuliert werden. Dazu setzt man die Bewertung zu Knoten mit denen im bipartiten Graphen eine Kante besteht auf den Wert 1 und die Bewertung zu allen anderen Knoten auf den Wert 0. Ein Beispiel für diese Methode kann in Abbildung 3.1 betrachtet werden.

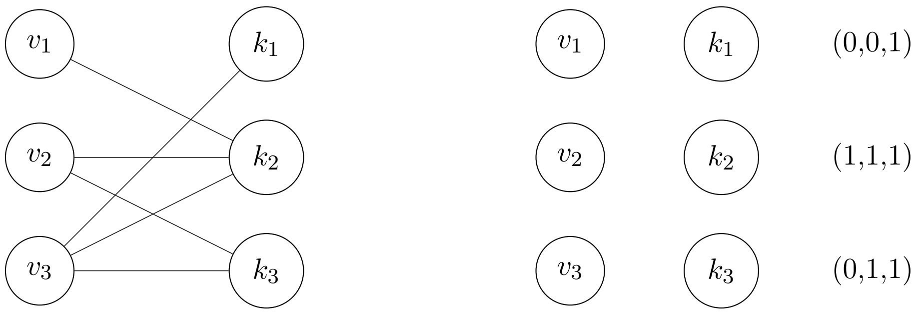

Abbildung 3.1: Bipartiter Graph und die entsprechende Repräsentation mit Bewertungen

Als nächste Erweiterung werden den Knoten $i \in V_1$ (ab jetzt ”**Verkäufer**” genannt) ganzzahlige **Preise** $p_i \geq 0$. zugeordnet. Die bevorzugten Verbindungen hängen jetzt von den Bewertungen und den Preisen ab. Hierfür definieren wir den Begriff **Payoff**, mit dessen Hilfe man die einzelnen Zuordnungen beurteilen kann.

Definition 3.1 (Payoff)**.**
Ein Verkäufer i habe einen Preis p_i und ein Käufer j habe für i eine Bewertung von $b_{i,j}$. Der **Payoff** beträgt dann: $P_{i,j} = b_{i,j} - p_i$

Der Payoff ist also die Differenz der Bewertung und des Preises. Die Käufer versuchen ihren eigenen Payoff zu maximieren. Ein Verkäufer v, der den Payoff eines bestimmten Käufers k maximiert, wird demnach als **bevorzugter Verkäufer** von k bezeichnet. Dies führt uns zu folgender Definition:

Definition 3.2 (preferred-seller-Graph)**.**
Der **preferred-seller-Graph** (kurz **p-s-Graph**) entsteht, indem zwischen jedem Käufer und seinem bevorzugten Verkäufer eine Kante gezogen wird.

In Abbildung 3.2 ist ein Beispiel dazu gegeben. Dabei ist auf der linken Seite ein Graph mit Bewertungen und Preisen und auf der rechten Seite der dazugehörige p-s-Graph. Um dies zu erstellen wurde für jeden Käufer k_i der Verkäufer v_j gesucht, welcher den Payoff maximiert. Für k_1 ist das beispielsweise v_1 mit dem Payoff $P = 1$. In diesem Beispiel wird deutlich, dass ein Käufer durchaus mehrere bevorzugte Verkäufer haben kann und dass ein Verkäufer von mehreren Käufern als bevorzugt bestimmt werden kann.

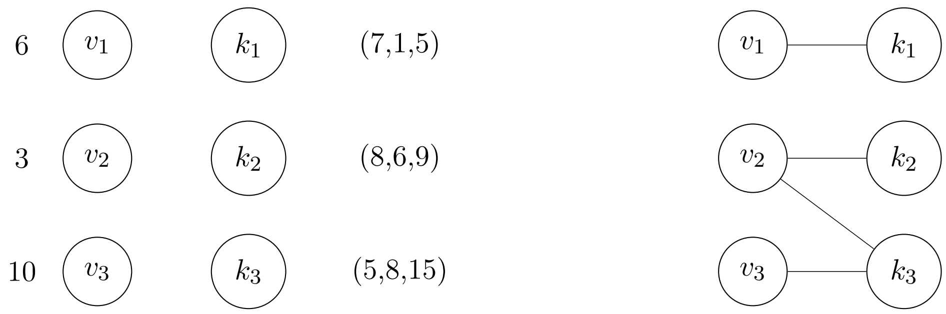

Abbildung 3.2: Beispiel für einen p-s-Graph

In einem weiteren Beispiel (Abbildung 3.3) ist der gleiche Graph mit denselben Gewichten wie zuvor gegeben, lediglich die Preise wurden verändert. Dabei sieht man, dass man bei unterschiedlichen Preisen auch unterschiedliche p-s-Graphen erhält.

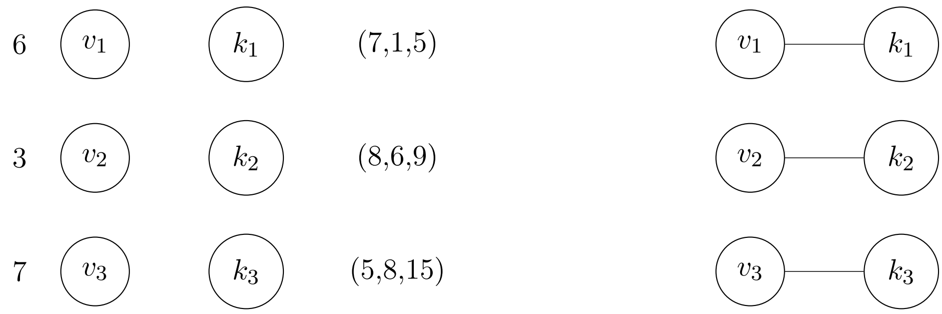

Abbildung 3.3: Analoges Beispiel zu Abbildung 3.2 mit anderen Preisen

4 Markträumende Preise

Im vorherigen Kapitel haben wir gesehen, wie die Preise den p-s-Graphen beeinflussen. Dies lässt den Schluss zu, dass man die Preise auf irgendeine Art und Weise anpassen kann, sodass man am Ende einen p-s-Graphen erhält, auf dem es ein perfektes Matching gibt.

Definition 4.1 (Marktäumende Preise)**.**
Preise werden **markträumend** genannt, wenn es auf dem resultierenden p-s-Graphen ein perfektes Matching gibt.

Hat man also markträumende Preise, kann jeder Käufer einen seiner bevorzugten Verkäufer auswählen, sodass keine Konflikte entstehen und damit jeder Verkäufer zu genau einem Käufer zugeordnet ist.

Satz 4.2 (Existenz markträumender Preise)**.**
Für beliebige Bewertungen der verschiedenen Käufer existieren markträumende Preise.

Um diesen Satz zu beweisen wird ein Algorithmus beschrieben, der zu einem gegebenen Graphen mit Bewertungen markträumende Preise für die Verkäufer berechnet. Dieser Algorithmus wird mit "`Auktion`" bezeichnet.

Am Anfang hat jeder Verkäufer den Preis 0. Der Algorithmus läuft solange in Runden, bis er ein "`BREAK`" erreicht und damit abbricht. Zu Beginn jeder Runde hat jeder Verkäufer einen Preis, wobei der kleinste Preis den Wert 0 hat. (Hat man Preise, deren kleinster Wert nicht 0 ist, muss man lediglich von jedem Preis den Wert des kleinsten Preises abziehen)

Der Algorithmus sieht wie folgt aus:

```
Erstelle den p-s-Graph
IF (der p-s-Graph hat ein perfektes Matching)
    THEN: BREAK, markträumende Preise gefunden.
    ELSE: Finde eine blockierende Menge S an Käufern
          jeder Verkäufer aus N(S) erhöht den Preis um 1
          reduziere die Preise, sodass der kleinste Preis 0 beträgt
          starte nächste Runde
```

Ein Beispiel für diesen Algorithmus wird in Abbildung 4.1 gegeben.

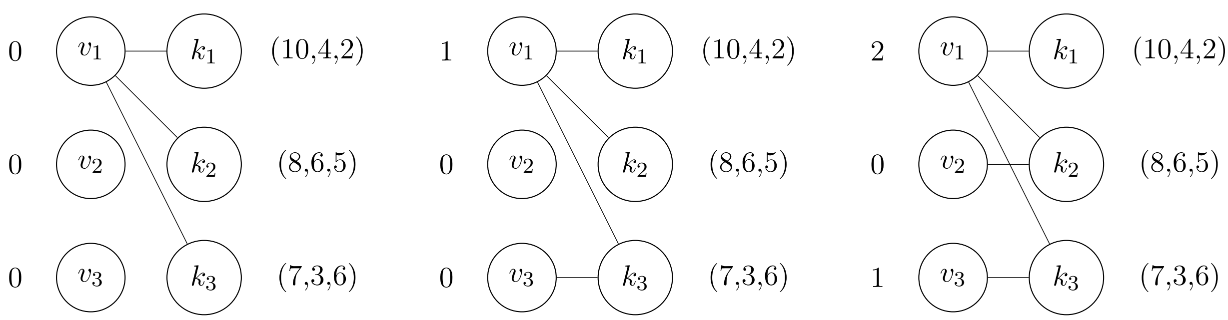

Abbildung 4.1: Beispiel für den `Auktion`-Algorithmus

`Auktion` bricht offensichtlich nur dann ab, wenn der p-s-Graph ein perfektes Matching besitzt. Dann sind die Preise per Definition markträumend. Dies wird im folgenden Lemma festgehalten.

Lemma 4.3. *Wenn `Auktion` terminiert, dann sind die berechneten Preise markträumend.*

Nun bleibt nur noch zu zeigen, dass `Auktion` für jede Eingabe terminiert.

Lemma 4.4. *`Auktion` terminiert für jede Eingabe.*

Beweis zu Lemma 4.4. Hierfür definieren wir für gegebene Preise p_i und Bewertungen $b_{i,j}$ die potentielle Energie $E(v)$ für jeden Knoten v des Graphen. Die potentielle Energie eines Käufers ist dabei sein maximaler Payoff. Für einen Verkäufer ist die potentielle Energie sein aktueller Preis. Die potentielle Energie des gesamten Graphen ergibt sich dann als Summe der potentiellen Energien aller Knoten:

$$E(G) = \sum_{v \in V} E(v)$$

Zu Beginn des `Auktion` Algorithmus hat jeder Verkäufer den Preis 0. Damit gilt $E(v) = 0$ für alle Verkäufer v. Dementsprechend ist die potentielle Energie eines Käufers gleich dem Wert seiner größten Bewertung. Für die gesamte potentielle Energie gilt zu Beginn also: $E(G) \geq 0$.

Da für die Preise p_v per Definition $p_v \geq 0$ gilt, ist die potentielle Energie der Verkäufer immer positiv. Zu Beginn jeder Runde von `Auktion` hat der kleinste Preis immer den Wert 0. Auch für alle Bewertungen $b_{i,j}$ gilt laut Definition $b_{i,j} \geq 0$. Damit ist der maximale Payoff eines jeden Käufers auch nicht negativ. Also gilt für das gesamte Potential vor jeder Runde: $E(G) \geq 0$.

Im Verlauf des Algorithmus werden die Bewertungen nicht verändert. Demnach ändert sich die potentielle Energie nur, wenn die Preise verändert werden. Dies passiert lediglich in den Zeilen 5 und 6.

- In Zeile 5 werden die Preise von jedem Knoten der Nachbarschaft $N(S)$ einer blockierenden Menge S um 1 erhöht. Somit steigt die potentielle Energie aller Verkäufer aus $N(S)$ jeweils um den Wert 1. Die potentielle Energie der Käufer aus S sinkt jedoch um jeweils 1, da der bevorzugte Verkäufer jedes Käufers aus S in $N(S)$ liegt. Da S eine blockierende Menge ist, gilt $|S| > |N(S)|$. Damit wird der gesamten potentiellen Energie in diesem Schritt mindestens eine Einheit mehr abgezogen als addiert. $E(G)$ wird also um mindestens 1 kleiner.

- In Zeile 6 werden die Preise reduziert, sodass der kleinste Preis den Wert 0 hat. Dabei wird von jedem Preis der Wert p des kleinsten Preises abgezogen. Somit sinkt die potentielle Energie eines jeden Verkäufers um p. Dabei steigt jedoch der Payoff (und damit auch die potentielle Energie) eines jeden Käufers genau um den Wert p. Da die Anzahl der Verkäufer gleich der Anzahl der Käufer ist, ändert sich die gesamte potentielle Energie nicht.

Damit wird in jeder Runde von `Auktion` die gesamte potentielle Energie um mindestens 1 verringert. Da zu Beginn $E(G) \geq 0$ gilt und die potentielle Energie nicht negativ werden kann, terminiert `Auktion` also nach einer endlichen Anzahl an Schritten. $\square$

Mit diesem Wissen kann nun der Beweis zum vorherigen Satz formuliert werden:

Beweis von Satz 4.2. Lemma 4.3 besagt, dass die von `Auktion` erstellten Preise markträumend sind und Lemma 4.4 besagt, dass `Auktion` auf jeder Eingabe terminiert. Folglich erstellt `Auktion` zu jeder Eingabe markträumende Preise. Dadurch ist die Existenz von markträumenden Preisen gezeigt. $\square$

Nun stellt sich die Frage, ob diese Preise auch dazu führen, dass die Gemeinschaft der Käufer mit dieser Zuteilung "zufrieden" ist. In diesem Zusammenhang nennt man eine Zuteilung sozial optimal, wenn die Summe der Bewertungen, welche ausgewählt wurden, maximiert wird.

Satz 4.5 (Optimalität markträumender Preise)**.**
Für beliebige markträumende Preise p_i und ein beliebiges perfektes Matching auf dem p-s-Graphen ist die Summe aller ausgewählten Bewertungen maximal.

Beweis von Satz 4.5. Gegeben seien markträumende Preise und ein perfektes Matching M auf dem entsprechenden p-s-Graphen. Die Preise und Bewertungen seien fixiert.
Der gesamte Payoff P_{ges} sei die Summe der Payoffs über alle Käufer im Bezug zu den jeweiligen Verkäufern, die im Matching zugeordnet sind. Da im p-s-Graph jeder Käufer seinen eigenen Payoff maximiert, wird also auch P_{ges} über alle im p-s-Graph mögliche Zuteilungen maximiert.
Sei v die bijektive Funktion, welche jedem Käufer den nach M zugehörigen Verkäufer zuordnet. Weiterhin sei b_k die Bewertung von Käufer k zu $v(k)$ und p_i der Preis von Verkäufer i. Dann gilt:

$$\begin{aligned} P_{\text{ges}} &= \sum_{k \in \text{ Käufer}} P_{v(k),k} = \sum_{k \in \text{ Käufer}} (b_k - p_{v(k)}) = \sum_{k \in \text{ Käufer}} b_k - \sum_{k \in \text{ Käufer}} p_{v(k)} \\ &\overset{*}{=} \sum_{k \in \text{ Käufer}} b_k - \sum_{i \in \text{ Verkäufer}} p_i \end{aligned}$$

* Dies ist möglich, wegen der Bijektivität von v.

Demnach ist der gesamte Payoff gleich der Summe aller ausgewählten Bewertungen minus der Summe aller Preise. Die Summe aller Preise ist unabhängig vom Matching immer gleich, sie kann also als konstant angenommen werden. Damit folgt, dass wenn eine Auswahl den gesamten Payoff P_{ges} maximiert, dann maximiert sie auch die Summe aller ausgewählten Bewertungen.
Damit folgt die Behauptung. □

Zusammenfassend gibt es also zu beliebigen Bewertungen Preise, welche zu einer sozial optimalen Zuteilung führen. Dies sind genau die markträumenden Preise.

Mit Hilfe dieses Modells lässt sich nun auch ein Markt modellieren, bei dem es nur einen einzigen Verkäufer gibt. Dazu werden so viele virtuelle Verkäufer erstellt bis die Anzahl an Verkäufern gleich der Anzahl an Käufern ist. Zu diesen virtuellen Verkäufern hat jeder Käufer die Bewertung 0. Auf dieser Eingabe kann jetzt der `Auktion`-Algorithmus durchgeführt werden. Dabei wird der Käufer dem einzigen "realen" Verkäufer zugeordnet, welcher die höchste Bewertung für ihn hat. Dieser Käufer muss jedoch nicht den Wert seiner Bewertung als Preis bezahlen, sondern lediglich den Wert der zweithöchsten Bewertung. Somit entspricht der `Auktion`-Algorithmus in dieser Anwendung der so genannten Zweitpreisauktion. Ein Beispiel hierfür ist in Abbildung 4.2 gegeben. Dabei ist v_1 der einzige "reale" Verkäufer.

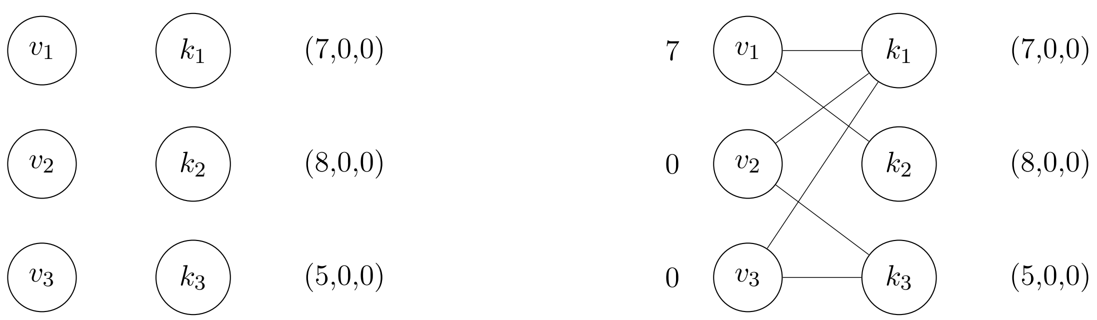

Abbildung 4.2: Beispiel für einen Markt mit einem einzelnem Verkäufer

Literaturverzeichnis

[1] `https://www.cs.cornell.edu/home/kleinber/networks-book/networks-book.pdf`, abgerufem am 27.11.2018

[2] `http://www.thi.cs.uni-frankfurt.de/lehre/dismod/ws1617/dismod_ws1617_skript.pdf`, abgerufen am 11.12.2018

[3] `https://www.math.uni-frankfurt.de/~dmst/teaching/lecture_notes/diskmathAlt/skriptTheobaldWS07.pdf`, abgerufen am 11.12.2018