

# Algorithmen für Matching-Märkte

Hui Jin  
Matrikel-Nr: 2123850

November 2005

Lehrstuhl  
Mathematische Grundlagen der Informatik  
an der Brandenburgischen Technischen Universität  
Cottbus

# Inhaltsverzeichnis

<b>Vorwort und Danksagung</b>	<b>1</b>
<b>1 Einleitung</b>	<b>2</b>
2.1 Definition eines Graphen .....	5
2.2 Matchings in bipartiten Graphen .....	6
2.3 Breitensuche .....	8
2.4 Gewichtete Matchings in bipartiten Graphen .....	9
2.5 Kooperative Spiele .....	11
2.6 Lineare Programmierung und Dualitätstheorie .....	11
2.7 Polynomialzeit .....	12
<b>3 Das Heiratsproblem</b>	<b>13</b>
3.1 Modelldarstellung .....	13
3.2 Die Spielregeln .....	14
3.3 Zulässigkeit und Stabilität von Outcomes .....	14
3.4 Der „men propose women dispose“ Algorithmus .....	15
<b>4 Das Zuweisungsspiel</b>	<b>19</b>
4.1 Modelldarstellung .....	19
4.2 Die Spielregeln .....	20
4.3 Zulässigkeit und Stabilität von Outcomes .....	20
4.4 Die Ungarische Methode .....	21
<b>5 Das verallgemeinerte Modell</b>	<b>27</b>

5.1	Modelldarstellung .....	27
5.2	Vorbereitungen .....	28
5.3	Ein Algorithmus für stabile Outcomes .....	33
5.3.1	Suche nach dem Weg $\mathcal{P}$ .....	33
5.3.2	Modifikation von Payoffs .....	34
5.3.3	Augmentierung .....	35
5.4	Korrektheits- und Komplexitätsanalyse .....	38
5.5	Spezialfälle .....	42
5.6	Beispiele .....	42
	<b>Notation</b>	<b>60</b>
	<b>Literaturverzeichnis</b>	<b>61</b>

## Zusammenfassung

Wir beschäftigen uns mit der Existenz von stabilen Matchings für sowohl diskrete als auch kontinuierliche Märkte, nämlich das Heiratsproblem und das Zuweisungsspiel. Dann entwickeln wir für das verallgemeinerte Modell von Eriksson und Karlander [5] einen polynomiellen ( $\mathcal{O}(n^4)$ ) Algorithmus nach den Schlüssellemmas von Sotomayor [8].

# Vorwort und Danksagung

Die vorliegende Arbeit entstand während meines Studiums an der Fakultät Mathematik, Naturwissenschaften und Informatik der Brandenburgischen Technischen Universität Cottbus. Die Arbeit wurde betreut von Herrn Prof. Dr. Winfried Hochstättler, ehemaliger Inhaber des Lehrstuhls für Mathematische Grundlagen der Informatik der Brandenburgischen Technischen Universität Cottbus.

Ich bedanke mich herzlich bei allen, die zum Gelingen dieser Arbeit beigetragen haben.

Herrn Prof. Dr. Winfried Hochstättler danke ich für die Betreuung dieser Arbeit, seine Nachsicht, Anregungen und Hinweise und die vielen Konsultationen in zwangloser Atmosphäre.

Herrn Prof. Dr. Heinz-Uwe Künenle möchte ich für die Erstellung des zweiten Gutachtens danken.

Besonders hervorheben möchte ich Herrn Robert Nickel, der mir intensive Konsultationen gewährte und die Arbeiten kritisch begleitete. Ohne ihn hätte ich diese Arbeit nicht schaffen können. Ihm gilt mein Dank für seine riesige Geduld, ständige Hilfsbereitschaft und uneigennützig und wirksame Hilfe.

Frau Fang Ji danke ich für die Freundschaft und ihre Begleitung während meines ganzen Studiums in Deutschland.

Bei Herrn Qiyong Cui und René Cabanis bedanke ich mich für das zeitaufwendige Korrekturlesen der Arbeit.

Besonderer Dank gilt meinen Eltern und meinem Freund für ihre moralische und finanzielle Unterstützung.

Ein spezieller Dank geht an all jene, die mir geholfen haben.

# Kapitel 1

## Einleitung

In der letzten Zeit sind beträchtliche Fortschritte mit dem Studium der zweiseitigen Matching Probleme aus spieltheoretischer Sicht gemacht worden, sowohl mit der Entwicklung der grundlegenden Theorie (siehe z.B. Roth und Sotomayor [10]) als auch mit dem Verstehen der praktischen Probleme, die im Zusammenhang mit der Implementierung des Matching Algorithmus stehen (siehe z.B. Roth [13] und Roth [14]).

Der Ausdruck *zweiseitig* bezieht sich auf die Tatsache, dass die Agenten in solchen Märkten von Anfang an zu einer von zwei disjunkten Mengen gehören - z.B. Firmen und Arbeiter. Dies steht im Kontrast zu Warenmärkten, in denen der Marktpreis darüber entscheiden kann, ob ein Agent ein Käufer oder ein Verkäufer ist. Z.B. hat der Markt für Gold sowohl Käufer als auch Verkäufer. Irgendein bestimmter Agent kann als ein Käufer zu einem Preis und als ein Verkäufer zu einem anderen Preis angesehen werden. Damit ist dieser Markt im entsprechenden Sinne nicht zweiseitig. Ein Arbeitsmarkt ist häufig ein zweiseitiger Markt, da die Mengen von Firmen und von Arbeitern disjunkt sind.

Der Ausdruck *Matching* bezieht sich auf die bilaterale Tauschart in diesen Märkten - z.B. eine Firma stellt genau einen Arbeiter ein. Diese Firma beschäftigt nur diesen einen Arbeiter, und dieser Arbeiter arbeitet dann nur in dieser Firma. Im Gegensatz dazu kann man mit einem Lastwagen Weizen in den Warenmarkt kommen, und mit einem neuen Traktor nach Hause zurück, obwohl der Weizenkäufer keine Traktoren verkauft und der Traktorenverkäufer keinen Weizen kauft.

An den zweiseitigen Märkten gibt es zwei fundamentale Modelle —das Heiratsproblem von Gale und Shapley [3] und das Zuweisungsspiel von Shapley und Shubik [11]. Das Heiratsproblem besagt: In einem Dorf gibt es  $m$  Frauen und

$n$  Männer im heiratsfähigen Alter. Jeder Mensch besitzt eine Präferenzliste von seinen akzeptablen Partnern/Partnerinnen. Unser Ziel ist, eine Hochzeit zu finden, in der es kein solches Paar  $(i, j)$  gibt, dass  $i$  und  $j$  lieber miteinander heiratet wären, als mit ihren momentanen Partnern. Solch eine Hochzeit nennen wir *stabil*. Den betreffenden Markt nennen wir diskret. Gale und Shapley [3] bewiesen mit einem Algorithmus namens „men propose-women dispose“, dass eine *stabile Hochzeit* immer existiert.

Das Zuweisungsspiel ist nichts anderes als das Problem der Bestimmung eines maximalen gewichteten Matchings in bipartiten Graphen. Im Zuweisungsspiel modelliert man Geld als eine kontinuierliche Variable, so heißt der diesbezügliche Markt kontinuierlich. Eine Lösung des Zuweisungsspiels, Outcome genannt, besteht aus einem Matching und einer Zuteilung von Gewichten. Man spricht von einem stabilen Outcome, wenn es kein Paar gibt, dessen Gewichtszuteilungssumme kleiner als das entsprechende Gewicht ist. Shapley und Shubik [11] zeigten mittels der Dualitätstheorie der Linearen Programmierung, dass die Menge der stabilen Outcomes nicht leer ist. Der klassische Algorithmus für gewichtete Matchings in bipartiten Graphen ist Kuhn's Ungarische Methode [7], welche in diesem Jahr 50 Jahre alt geworden ist.

Die empirische Bedeutung beider Modelle kommt aus der Variation des Arbeitsmarktes. Stellen wir uns vor, einige neue Absolventen einer juristischen Fakultät gehen in den Markt hinein, um eine gehobene Position in privaten juristischen Firmen zu finden, aus der sie flexibles Gehalt beziehen können, oder sie bewerben sich um eine Anstellung im öffentlichen Dienst als Verwaltungschef im Bundesgerichtshof mit festem Gehalt nach Tarif. Normalerweise wird der erstere Markt als ein Zuweisungsspiel modelliert, in dem das Gehalt ausgehandelt werden kann und sich kontinuierlich in der Menge der reellen Zahlen verändert. Dahingegen wird der letztere als ein Heiratsmarkt modelliert.

Für die beiden Modelle wurden ähnliche Ergebnisse erzielt, z.B. die Gleichheit des Kerns und der Menge der stabilen Outcomes, sowie die Verbandstruktur des Kerns. Roth und Sotomayor [10] fragten nach einer Erklärung für diese Ähnlichkeit. Dazu legten Eriksson und Karlander [5] einen einzelnen Markt dar, der die diskrete und kontinuierliche Märkte beinhaltet. Das Interesse des Modells ist zu zeigen, dass ein Agent gleichzeitig in beiden Märkten Geschäfte machen kann. Eriksson und Karlander [5] lieferten einen algorithmischen Beweis für die Existenz von stabilen Outcomes. Der entsprechende Algorithmus ist nicht polynomiell aber pseudopolynomiell.

Sotomayor [8] zeigte, dass ein stabiles Outcome für das verallgemeinerte Modell

von Eriksson und Karlander [5] immer existiert. Fujishige und Tamura [2] waren der Meinung, dass der Beweis von Sotomayor [8] nicht-konstruktiv ist. Hingegen wird in dieser Arbeit ein polynomieller ( $\mathcal{O}(n^4)$ ) Algorithmus aus den Schlüssellemmas von Sotomayor [8] gewonnen.

Im Kapitel 2 führe ich mehrere Grundbegriffe und Ergebnisse von Graphentheorie, Linearer Programmierung und Spieltheorie ein, um die drei Modelle vorzustellen. In den Kapiteln 3, 4 und 5 werden das Heiratsproblem, das Zuweisungsspiel und das verallgemeinerte Modell gesondert beschrieben. Als Lösungsmethode bespreche ich für das Heiratsproblem den Algorithmus „men propose-women dispose“, für das Zuweisungsspiel die Ungarische Methode in der Version nach A. Frank [1] und für das verallgemeinerte Modell einen neuen Algorithmus nach den Schlüssellemmas von Sotomayor [8].



# Kapitel 2

## Grundbegriffe

In diesem Kapitel führen wir Grundbegriffe ein, die die Beschreibung unserer Modelle ermöglichen. Als erstes besprechen wir Definition eines Graphen.

### 2.1 Definition eines Graphen

Sei  $V$  eine endliche Menge (von Knoten) und  $A$  eine endliche Menge (von Bögen). Wir haben zwei Abbildungen für  $A$ :

$$\begin{array}{ll} \textit{head} : A \rightarrow V & \textit{tail} : A \rightarrow V \\ a \rightarrow \textit{head}(a) & a \rightarrow \textit{tail}(a) \end{array}$$

die jedem Bogen einen eindeutigen Anfang  $\textit{head}$  und ein Ende  $\textit{tail}$  zuordnen. Das Quadruple  $D = (V, A, \textit{head}, \textit{tail})$  nennen wir einen **gerichteten Graphen** oder kurz **Digraphen**. Spielt die Orientierung eines Bogens keine Rolle, nennen wir ihn *Kante* und die Menge der Kanten bezeichnen wir mit  $E$ . Wir erhalten dann ein Triple  $G = (V, E, \varphi)$ , wobei  $\varphi$  die *Inzidenzfunktion* ist, die jede Kante einen oder zwei Knoten zuordnet (die Endknoten der Kante). Ein solches Triple nennen wir einen **Graphen**.

Für diese gesamte Arbeit sei  $G = (P \dot{\cup} Q, E)$  ein fester bipartiter Graph mit Knotenpartition  $\{P, Q\}$ , also liegen die Endknoten jeder Kante in verschiedenen Klassen der Partition. Als  $i$  usw. bezeichnete Knoten von  $G$  liegen generell in  $P$ , als  $j$  usw. bezeichnete in  $Q$ .

## 2.2 Matchings in bipartiten Graphen

**Definition 2.2.1.** Eine Menge  $\mu$  von paarweise nicht benachbarten Kanten in einem (nicht notwendig bipartiten) Graphen  $G = (V, E)$  nennt man ein **Matching** oder eine **Paarung**.  $\mu$  ist ein **Matching von**  $U \subseteq V$ , wenn jeder Knoten aus  $U$  mit einer Kante aus  $\mu$  inzident ist.

Wir sagen  $i$  ist mit  $j$  **gematcht**, wenn  $(u, v) \in \mu$ ; mit keiner Kante aus  $\mu$  inzidente Knoten sind **ungematcht**.

Man schreibt  $\mu(i) = j$  oder  $\mu(j) = i$ , wenn  $i$  und  $j$  miteinander gematcht sind.

**Definition 2.2.2.** Zu jedem Matching  $\mu$  kann man eine Matrix  $x(\mu) = x = (x_{ij})$  von Nullen und Einsen definieren durch

$$x_{ij} = \begin{cases} 1 & , \text{ falls } i \text{ und } j \text{ miteinander gematcht sind;} \\ 0 & , \text{ sonst.} \end{cases}$$

Diese Matrix nennen wir die **Matrix von**  $\mu$ .

**Bemerkung:** Daher haben wir

$$\sum_{i \in P} x_{ij} \leq 1, \sum_{j \in Q} x_{ij} \leq 1;$$

$$\sum_{i \in P} x_{ij} = 0 \text{ (bzw. } \sum_{j \in Q} x_{ij} = 0), \text{ wenn } j \text{ (bzw. } i) \text{ ungematcht ist.}$$

Die Aufgabe, in einem bipartiten Graphen ein Matching mit möglichst vielen Kanten zu finden, ist als „*Heiratsproblem*“ bekannt, welches wir im Kapitel 3 beschreiben werden.

*Alternierende Wege* spielen bei der praktischen Suche nach möglichst großen Matchings eine wesentliche Rolle. Betrachten wir ein beliebiges Matching  $\mu$  im Graphen  $G$ .

**Definition 2.2.3.** Ein Weg in  $G$ , der in einem ungematchten Knoten aus  $P$  beginnt und dann abwechselnd Kanten aus  $E \setminus \mu$  und aus  $\mu$  enthält, ist ein **alternierender Weg** (bezüglich  $\mu$ ).

Einen alternierenden Weg  $\mathcal{P}$ , der in einem ungematchten Knoten aus  $Q$  endet, nennt man einen **augmentierenden Weg**.

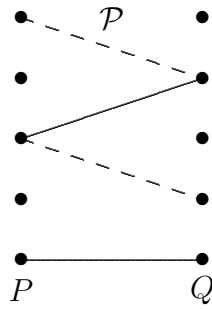


Abbildung 2.1: Beispiel für den augmentierenden Weg  $\mathcal{P}$

Im Folgenden wollen wir einige Sätze für die maximale Kardinalität eines Matchings zitieren. Es hilft beim Verstehen der Algorithmen, die in den folgenden Kapiteln dargestellt werden.

**Satz 2.2.4 (Augmentierender Weg Satz).** *Sei  $G = (V, E)$  ein (nicht notwendig bipartiter) Graph und  $\mu$  ein Matching in  $G$ . Dann ist  $\mu$  genau dann kardinalitätsmaximal, wenn es keinen augmentierenden Weg gibt.*

Der folgende Satz charakterisiert die maximale Kardinalität eines Matchings  $\mu$  durch eine Art Dualitätsbedingung.

**Definition 2.2.5.** *Sei  $G = (V, E)$  ein Graph und  $U \subseteq V$ . Dann heißt  $U$  eine **Überdeckung von  $G$**  (kantenüberdeckende Knotenmenge), falls jede Kante mindestens einen Knoten in  $U$  hat.*

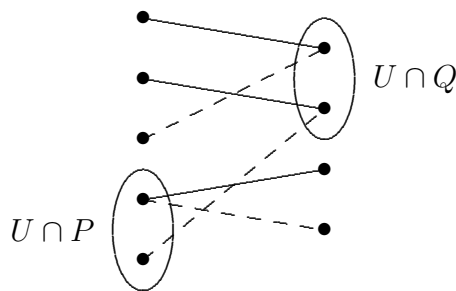


Abbildung 2.2: Beispiel für die Überdeckung  $U$

**Satz 2.2.6 (Satz von König).** *Sei  $G = (P \dot{\cup} Q, E)$  ein bipartiter Graph. Dann gilt*

$$\max\{|\mu| \mid \mu \text{ ist Matching von } G\} = \min\{|U| \mid U \text{ ist Überdeckung von } G\}.$$

## 2.3 Breitensuche

Einen augmentierenden Weg bestimmen wir algorithmisch mit Suchverfahren, in dieser Arbeit mit Breitensuche (BFS). Es wird eine Queue verwendet, um die Knoten zu speichern. Vom Startknoten aus besucht der Algorithmus alle an den Startknoten angeschlossenen Knoten, markiert sie als besucht und legt sie auf die Queue. Sind alle angeschlossenen Knoten besucht, entfernt er einen Knoten von der Queue und besucht alle unbesuchten, an diesen Knoten angeschlossenen Knoten und legt sie ebenfalls auf die Queue. Dies macht er so lange, bis ein Abbruchkriterium erfüllt wird oder die Queue leer ist.

Wir erhalten folgenden Code:

---

**Algorithmus 1** Breitensuche

---

```
1: procedure BFS( $(G, s)$ )           ▷  $s$  ist Startknoten und  $G$  ein (Di-)Graph
2:    $Q \leftarrow \emptyset$ ;
3:   Vorgänger[ $s$ ]  $\leftarrow -1$ ;           ▷ Hier beginnt der Pfad
4:    $Q.append(s)$ ;                       ▷ Lege  $s$  auf den Queue  $Q$ 
5:   while  $Q \neq \emptyset$  do
6:      $k=Q.top()$ ;                         ▷ Entnimm und entferne  $k$  aus  $Q$ 
7:     for  $w \in k.Nachbarschaft$  do       ▷ Es gibt einen Pfeil von  $k$  nach  $w$ 
8:       if  $Vorgänger[w] = \emptyset$  then
9:          $Vorgänger[w] \leftarrow k$ ;
10:      if  $w$  erfüllt Abbruchkriterium then
11:         $\mathcal{P} \leftarrow \langle s, \dots, w \rangle$ ;
12:      else
13:         $Q.append(w)$ ;
14:      end if
15:    end if
16:  end for
17: end while
18: end procedure
```

---

**Satz 2.3.1.** *BFS berechnet die Komponenten eines Graphen  $G = (V, E)$  in  $\mathcal{O}(|V| + |E|)$ , wenn kein Abbruchkriterium festgelegt ist.*

## 2.4 Gewichtete Matchings in bipartiten Graphen

Im vorigen Abschnitt haben wir die maximale Kardinalität eines Matchings diskutiert. In diesem Abschnitt behandeln wir gewichtete Matchings. Das Problem, in einem bipartiten Graphen ein Matching mit maximalem Gewicht zu finden, ist äquivalent zum „Zuweisungsspiel“.

**Definition 2.4.1.** *Es sei  $G = (P \dot{\cup} Q, E)$  ein bipartiter Graph mit Gewichtsfunktion  $\alpha : E \rightarrow \mathbb{R}$ . Das **Gewicht**  $\alpha(\mu)$  eines Matchings  $\mu$  von  $G$  wird definiert durch  $\alpha(\mu) := \sum_{e \in \mu} \alpha(e)$ . Man spricht von einem **maximalen gewichteten Matching**  $\mu$ , wenn  $\alpha(\mu) \geq \alpha(\mu')$  für alle Matchings  $\mu'$  von  $G$  gilt.*

**Bemerkung:** In unseren Modellen betrachten wir nur die Matchings mit nicht negativer Gewichtsfunktion, also die *individuell rationalen* Matchings.

Wenn wir der kleineren Knotenmenge geeignet viele Dummy-Knoten hinzufügen und die entsprechenden Kantengewichte auf 0 setzen, erhalten wir  $|P| = |Q|$ . Wir betrachten also nur die Matchings von Maximalgewicht in einem Graphen mit der Knotenpartition  $|P| = |Q|$ .

In verschiedenen Modellen werden die *Zulässigkeit* und die *Stabilität* von Outcomes (bzw. Payoffs) unterschiedlich definiert. Das Heiratsproblem und das Zuweisungsspiel sind Spezialfälle vom verallgemeinerten Modell. Deswegen benutzen wir die Definitionen aus dem verallgemeinerten Modell.

Im verallgemeinerten Modell wird die Knotenmenge in zwei Klassen  $F$  und  $R$  aufgeteilt, d.h.  $F \dot{\cup} R = P \dot{\cup} Q$ ; und die Gewichtsfunktion  $\alpha$  in zwei Matrizen  $A = (a_{ij})$  und  $B = (b_{ij})$ , d.h.  $\alpha_{ij} = a_{ij} + b_{ij}$ . Den Sinn der Aufteilung besprechen wir im Kapitel 5 auf Seite 27.

**Definition 2.4.2.** *Die Mengen  $F^*$  und  $R^*$  werden definiert durch*

$$F^* := \{(i, j) \in P \times Q \mid i \in F \text{ und } j \in F\}$$

$$R^* := \{(i, j) \in P \times Q \mid i \in R \text{ oder } j \in R\}.$$

*Daher ist  $P \times Q = F^* \cup R^*$ .*

**Definition 2.4.3.** *Ein **Outcome** ist das Matching  $\mu$  und ein Paar von Vektoren  $(u, v) \in \mathbb{R}^n \times \mathbb{R}^n$ , das **Payoff** genannt wird. Mit  $(u, v; \mu)$  bezeichnet man ein *Outcome*.*

**Definition 2.4.4.** Ein Outcome  $(u, v; \mu)$  heißt **zulässig**, wenn

1.  $u_i \geq 0, v_j \geq 0$  für alle  $(i, j) \in P \times Q$  und  
 $u_i = 0$  (bzw.  $v_j = 0$ , falls  $i$  (bzw.  $j$ ) ungematcht ist;
2.  $u_i + v_j = a_{ij} + b_{ij}$ , falls  $(i, j) \in \mu$ ;
3.  $u_i = a_{ij}$  und  $v_j = b_{ij}$ , falls  $(i, j) \in \mu \cap R^*$ .

Ein Payoff  $(u, v) \in \mathbb{R}^n \times \mathbb{R}^n$  heißt **zulässig**, wenn ein Matching  $\mu$  existiert, so dass das Outcome  $(u, v; \mu)$  zulässig ist.

**Definition 2.4.5.** Ein Outcome  $(u, v; \mu)$  heißt **stabil**, wenn  $(u, v; \mu)$  zulässig ist, und für alle  $(i, j) \in P \times Q$  gelten:

1.  $u_i + v_j \geq a_{ij} + b_{ij}$ , falls  $(i, j) \in F^*$ ;
2.  $u_i \geq a_{ij}$  oder  $v_j \geq b_{ij}$ , falls  $(i, j) \in R^*$ .

Ein Payoff  $(u, v)$  mit 1 und 2 heißt **stabil**.

**Definition 2.4.6.** Ein **blockierendes Paar** von einem zulässigen Outcome  $(u, v; \mu)$  (bzw. von einem Payoff  $(u, v)$ ) ist ein Paar  $(i, j) \in P \times Q$  mit

$$\begin{aligned} u_i + v_j &< a_{ij} + b_{ij} && , \quad \text{falls } (i, j) \in F^* \text{ oder} \\ u_i < a_{ij} \quad \text{und} \quad v_j < b_{ij} && , \quad \text{falls } (i, j) \in R^* \end{aligned}$$

Hierzu sagt man auch, dass  $i$  und  $j$  blockierende Partner sind.

**Bemerkung:** Im Vergleich mit Definition 2.4.5 sehen wir, dass ein zulässiges Outcome genau dann stabil ist, wenn es keine blockierenden Paare gibt.

**Definition 2.4.7.** Für ein zulässiges Outcome  $(u, v; \mu)$  sei  $Y_i$  die Menge aller blockierenden Partner von  $i$ .

Für jedes  $j \in Y_i$  wird ein zulässiges Outcome  $(u^j, v^j; \mu^j)$  definiert, so dass

1.  $i$  und  $j$  miteinander gematcht sind, also  $(i, j) \in \mu^j$ , die Matching-Partner von  $i$  und  $j$  im Matching  $\mu$  ungematcht sind, wenn sie existieren, und andere Knoten unverändert bleiben;
2.  $u_i^j = a_{ij}$ , falls  $(i, j) \in R^*$ , und
3.  $u_i^j = a_{ij} + b_{ij} - v_j$ , falls  $(i, j) \in F^*$ .

Wenn  $u_i^j \geq u_i^k$  für alle  $k \in Y_i$  gilt, dann ist  $j$  ein **blockierender Traumpartner** von  $i$ .

**Bemerkung:**  $u_i^j$  ist der größtmögliche Payoff von  $i$ , wenn  $i$  und  $j$  ihre gegenwärtige Partnerschaft brechen und zusammenarbeiten. Es deutet an, dass  $i$  in  $(u^j, v^j; \mu^j)$  keine blockierenden Paare erzeugen kann, wenn  $j$  ein blockierender Traumpartner von  $i$  ist.

## 2.5 Kooperative Spiele

In spieltheoretischer Sprache sind unsere Modelle *kooperative Spiele*. Nach Neumann und Morgenstern [21] wird ein kooperatives Spiel wie folgt definiert.

**Definition 2.5.1.** Ein **kooperatives Spiel** ist ein Paar  $(N, \nu)$ , wobei  $N$  eine endliche Menge von Spielern ist und  $\nu : 2^N \rightarrow \mathbb{R}$  eine super-additive, normierte Funktion ist, d.h.,

1.  $\nu(\emptyset) = 0$  und
2.  $S \cap T = \emptyset \Rightarrow \nu(S \cup T) \geq \nu(S) + \nu(T)$ .

Eine Teilmenge  $S \subseteq N$  wird als eine **Koalition** bezeichnet,  $N$  als **Gesamtkoalition**, und  $\nu$  als die **charakteristische Funktion**.

**Definition 2.5.2.** Sei  $(N, \nu)$  ein kooperatives Spiel. Der **Kern** des Spiels ist ein Polyeder  $\text{core}(\nu) \subseteq \mathbb{R}^N$ , welches durch folgendes System definiert wird:

$$\begin{aligned} x(N) &:= \sum_{i \in N} x_i = \nu(N); \\ x(S) &:= \sum_{i \in S} x_i \geq \nu(S) \quad \forall S \subset N. \end{aligned}$$

## 2.6 Lineare Programmierung und Dualitätstheorie

Im Kapitel 4 wird ein berühmter Algorithmus für gewichtete Matchings in bipartiten Graphen, die sogenannte Ungarische Methode, beschrieben. Zur besseren Erklärung, wieso dieser Algorithmus funktioniert, wiederholen wir einige Ergebnisse der Dualitätstheorie.

**Definition 2.6.1.** Sei  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $b \geq 0$  und  $c \in \mathbb{R}^n$ . Ferner sei  $m \leq n$  und  $A$  habe vollen Rang, also  $\text{rang}(A) = m$ . Die Aufgabenstellung

$$\begin{aligned} \max \quad & c^\top x \\ \text{unter} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

nennen wir Lineares Optimierungsproblem in Standardform. Ist  $x \geq 0$  mit  $Ax = b$ , so sagen wir,  $x$  ist **zulässig** für das Problem.

**Definition 2.6.2.** *Das Lineare Programm*

$$\begin{array}{ll} \min & y^\top b \\ \text{unter} & y^\top A \geq c^\top \end{array}$$

heißt das **duale Programm** zum Linearen Programm in Standardform. Das Ausgangsproblem bezeichnen wir als das **Primale Programm**.

Das folgende Lemma zeigt, dass das duale und primale Programm die gleiche Zielfunktion haben.

**Satz 2.6.3 (Starke Dualität).** *Ist  $x \geq 0$  zulässig für das primale Programm und  $y$  für das duale Programm, so gilt  $c^\top x \leq y^\top b$ . Das Gleichheitszeichen gilt genau dann, wenn  $x$  und  $y$  die optimalen Lösungen vom primalen Programm und dualen Programm sind.*

## 2.7 Polynomialzeit

Um die Aufwandsabschätzung für einen Algorithmus zu vereinfachen, ist es nützlich, die  $\mathcal{O}$ -Notation zu verwenden.

Seien  $k$  eine natürliche Zahl,  $X, Y \subset \mathbb{N}^k$  und  $f : X \rightarrow \mathbb{R}$ ,  $g : Y \rightarrow \mathbb{R}$  Funktionen. Man schreibt  $f = \mathcal{O}(g)$ , falls es positive reelle Zahlen  $B$  und  $C$  gibt, derart, dass für alle  $(n_1, \dots, n_k) \in \mathbb{N}^k$  mit  $n_i > B$ ,  $1 \leq i \leq k$ , folgendes gilt:

1.  $(n_1, \dots, n_k) \in X \cap Y$  d.h.  $f(n_1, \dots, n_k)$  und  $g(n_1, \dots, n_k)$  sind definiert,
2.  $f(n_1, \dots, n_k) \leq Cg(n_1, \dots, n_k)$ .

Das bedeutet, dass fast überall  $f(n_1, \dots, n_k) \leq Cg(n_1, \dots, n_k)$  gilt. Ist  $g$  eine Konstante, so schreibt man  $f = \mathcal{O}(1)$ .

**Definition 2.7.1.** *Angenommen, ein Algorithmus bekommt als Eingabe ganze Zahlen  $z_1, \dots, z_n$ . Man sagt, dass dieser Algorithmus **polynomielle Laufzeit** hat, wenn es nicht negative ganze Zahlen  $e_1, \dots, e_n$  gibt, so dass der Algorithmus die Laufzeit*

$$\mathcal{O}((\text{size } z_1)^{e_1}, \text{size } z_2)^{e_2} \cdots \text{size } z_n)^{e_n})$$

*hat.*



# Kapitel 3

## Das Heiratsproblem

In diesem Kapitel wird das diskrete Modell, das sogenannte Heiratsproblem, eingeführt. Gale und Shapley [3] formulierten dieses Modell, und bewiesen algorithmisch die Existenz von stabilen Matchings, wenn die Präferenzlisten strikt sind. Traditionell werden die beiden Mengen der Agenten für Männer und Frauen gehalten, statt Firmen und Arbeiter, Käufer und Verkäufer, usw. Es ist vielleicht nützlich, dass man die Agenten als die Männer und Frauen im heiratsfähigen Alter in einem Dorf sieht.

### 3.1 Modelldarstellung

Wir geben zuerst das Heiratsmodell von Gale und Shapley [3] wieder.

Es gibt zwei endliche disjunkte Mengen  $W$  und  $M$ :  $W = \{w_1, w_2, \dots, w_m\}$  ist die Menge der Frauen, und  $M = \{m_1, \dots, m_n\}$  ist die Menge der Männer. Jeder Mensch gibt in einer Liste seine akzeptablen Heiratspartner bekannt. Die Präferenzliste eines Agenten repräsentiert, wie er oder sie sich für eine von mehreren verschiedenen Alternativen entscheidet, wenn er oder sie eine Wahl trifft.

Die Präferenzliste  $L(w)$  von  $w$  sieht z.B. so aus:  $L(w) = h, [j, k], \dots, w, p$ . Das bedeutet,  $h$  ist der Lieblingspartner von  $w$ ; für  $w$  sind  $j$  und  $k$  gleichwertig; und  $p$  ist für  $w$  inakzeptabel, also würde  $w$  lieber ledig bleiben, wenn nur  $p$  verfügbar ist.

- Die Schreibweise  $w >_m w'$  bedeutet, dass der Mann  $m$  die Frau  $w$  der Frau  $w'$  vorzieht, also würde  $m$  die Frau  $w$ , aber nicht  $w'$  wählen, wenn er eine Wahl zwischen  $w$  und  $w'$  trifft. Die Bedeutung von  $m >_w m'$  ist analog.
- Wenn man sagt, dass  $w$  und  $w'$  für einen Spieler  $m$  gleichwertig sind (im Zeichen  $w =_m w'$ ), meint er, dass, wenn der Spieler eine Wahl zwischen  $w$  und  $w'$  trifft, er dann irgendeine von beiden wählen wird.
- Wenn entweder ein Mann  $m$  die Alternative  $w$  der Alternative  $w'$  vorzieht oder  $w$  und  $w'$  für ihn gleichwertig sind, so sagt man, er mag  $w$  mindestens wie  $w'$ .

## 3.2 Die Spielregeln

Wie in einer spieltheoretischen Analyse sind die Regeln des Spiels ganz wichtig, denn diese beeinflussen jeden Aspekt der Analyse. Im ganzen Modell nehmen wir an:

1. Ein Mann und eine Frau, die ihr Einverständnis zur Heirat miteinander geben, können heiraten.
2. Jeder Agent kann sein Einverständnis frei versagen und ledig bleiben.

Nach den Spielregeln wird kein Agent zur Heirat gezwungen, also betrachten wir nicht die Matchings, die auf den Zwang eines Agenten hinauslaufen. Stellen wir uns vor, in einem Matching  $\mu$  ist ein solches irreguläres Paar  $(m, w)$  gematcht. So würde mindestens einer davon lieber ledig bleiben als den anderen zu heiraten.

Z.B. die Frau  $w$  würde lieber ledig bleiben als den Mann  $m$  zu heiraten. D.h., sie wird nie den Mann  $m$  heiraten, solange Ledigbleiben eine Alternative für sie ist. Die Spielregel stellt sicher, dass die Option Ledigbleiben immer verfügbar ist.

## 3.3 Zulässigkeit und Stabilität von Outcomes

Wir identifizieren die Zulässigkeit und die Stabilität mit Definition 2.4.4 und Definition 2.4.5.

Hier ist  $F = \emptyset$ . Mit  $a_{ij}$  spielt  $i$  auf die Akzeptabilität von  $j$ .  $j$  ist von  $i$  akzeptabel, wenn  $a_{ij} > 0$ . Analog spielt  $j$  mit  $b_{ij}$  auf die Akzeptabilität von  $i$ .  $i$  ist von  $j$  akzeptabel, wenn  $b_{ij} > 0$ . Die Präferenzliste  $L(w)$  auf Seite 13 bedeutet  $a_{eh} > a_{wj} = a_{wk} > 0 = a_{wp}$ .

Ein Outcome  $(u, v; \mu)$  ist in diesem Modell vom Matching  $\mu$  eindeutig bestimmt. Wenn  $i$  und  $j$  miteinander gematcht sind, dann kriegt  $i$  (bzw.  $j$ ) den Payoff  $u_i = a_{ij}$  (bzw.  $v_j = b_{ij}$ ). Ein solches Matching heißt *zulässig* (bzw. *stabil*), wenn das entsprechende Outcome zulässig (bzw. stabil) ist, also wenn jedes gematchte Paar voneinander akzeptabel ist.

Sehen wir uns ein Matching  $\mu$  an, in dem ein Mann  $i$  und eine Frau  $j$  nicht miteinander gematcht sind, aber sie bevorzugen einander vor ihren momentanen Partnern, nämlich  $u_i = a_{i,\mu(i)} < a_{ij}$  und  $v_j = b_{\mu(j),j} < b_{ij}$ . Der Mann und die Frau  $(i, j)$  werden *blockierendes Paar* genannt, oder man sagt, dass der Mann und die Frau  $(i, j)$  das matching  $\mu$  blockieren. Mann  $i$  und Frau  $j$  würden gerne ihre momentanen Beziehungen beenden und miteinander heiraten. Die Spielregeln erlauben ihnen, so zu tun. Damit ist das Matching  $\mu$  in diesem Sinne unstabil. Also ist ein Matching  $\mu$  stabil, wenn es keine blockierenden Paare gibt.

**Beispiel:** Es gibt drei Männer und drei Frauen mit folgenden Präferenzlisten:

$$\begin{aligned} L(m_1) &= w_1, w_3, w_2 & L(w_1) &= m_2, m_1, m_3 \\ L(m_2) &= w_3, w_1, w_2 & L(w_2) &= m_1, m_3, m_2 \\ L(m_3) &= w_1, w_3, w_2 & L(w_3) &= m_1, m_2, m_3 \end{aligned}$$

Alle möglichen Matchings sind individuell rational, weil alle Paare  $(m, w)$  voneinander akzeptierbar sind. Das Matching  $\mu = \{(m_1, w_1), (m_2, w_2), (m_3, w_3)\}$  ist wegen des blockierenden Paares  $(m_2, w_1)$  unstabil. Doch ist das Matching  $\mu' = \{(m_1, w_1), (m_2, w_3), (m_3, w_2)\}$  stabil.

### 3.4 Der „men propose women dispose“ Algorithmus

Wenn die Präferenzlisten **strikt** sind, ist es eben das in Gale und Shapley [3] vorgestellte „Marriage Model“. Im folgenden erläutern wir den bekannten „men propose-women dispose“ Algorithmus, mit dem man ein stabiles Matching finden kann, wenn die Präferenzlisten strikt sind.

---

**Algorithmus 2** Men Propose Women Dispose

---

```
1: while ein mann  $m \in M$  nicht verlobt do
2:    $m$  macht der ersten Frau  $w$  auf seiner Liste einen Antrag;
3:   if  $w$  nicht verlobt then
4:      $m$  und  $w$  verloben sich:  $(m, w) \in \mu$ ;
5:   else if bisher  $(\hat{m}, w) \in \mu$ , aber  $m >_w \hat{m}$  then
6:     Verlobung  $(\hat{m}, w)$  wird gelöst;
7:      $m$  und  $w$  verloben sich:  $(m, w) \in \mu$ ;
8:      $\hat{m}$  streicht  $w$  von seiner Liste;
9:   else ▷  $w$  ist glücklicher mit  $\hat{m}$  als mit  $m$ 
10:     $m$  streicht  $w$  von seiner Liste;
11:   end if
12: end while
```

---

Das Matching, welches man mit diesem Algorithmus erhält, ist für die Männer optimal, d.h., jeder Mann heiratet die für ihn bestmöglich akzeptable Frau. Mit einem Beispiel erklären wir den Algorithmus.

**Beispiel 3.5.** Die Präferenzlisten sehen so aus:

$$\begin{aligned}L(m_1) &= w_2, w_1, w_4, w_3 \\L(m_2) &= w_3, w_1, w_4, w_2 \\L(m_3) &= w_3, w_4, w_2, w_1 \\L(m_4) &= w_2, w_3, w_4, w_1 \\L(m_5) &= w_2, w_1, w_3\end{aligned}$$

$$\begin{aligned}L(w_1) &= m_3, m_1, m_2, m_4, m_5 \\L(w_2) &= m_2, m_3, m_1, m_4, m_5 \\L(w_3) &= m_1, m_4, m_5, m_2, m_3 \\L(w_4) &= m_5, m_4, m_1, m_2, m_3\end{aligned}$$

• *Erster Schritt:*

$m_1, m_4$  und  $m_5$  machen  $w_2$  einen Antrag;  
 $m_2$  und  $m_3$  machen  $w_3$  einen Antrag;  
 $w_2$  lehnt  $m_4$  und  $m_5$  ab und verlobt sich mit  $m_1$ ;  
 $w_3$  lehnt  $m_3$  ab und verlobt sich mit  $m_2$ .

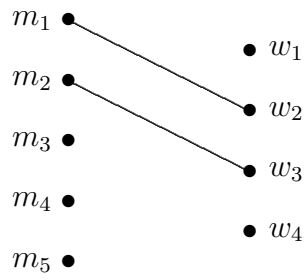


Abbildung 3.1: Erster Schritt

- *Zweiter Schritt:*

$m_3, m_4$  und  $m_5$  machen ihrer zweiten Wahl ( $w_4, w_3$  und  $w_1$ ) einen Antrag;  
Die Verlobung  $(m_2, w_3)$  wird gelöst und  $w_3$  verlobt sich mit  $m_4$ .

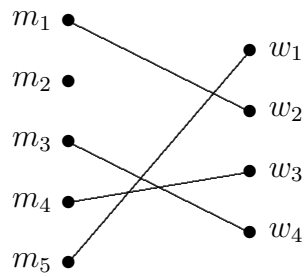


Abbildung 3.2: Zweiter Schritt

- *Dritter Schritt:*

$m_2$  macht seiner zweiten Wahl  $w_1$  einen Antrag;  
Die Verlobung  $(m_5, w_1)$  wird gelöst und  $w_1$  verlobt sich mit  $m_2$ .

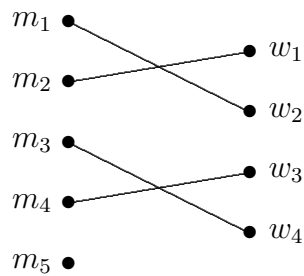


Abbildung 3.3: Dritter Schritt

- *Vierter Schritt:*

$m_5$  macht seiner dritten Wahl  $w_3$  einen Antrag und wird abgelehnt. So ist  $m_5$  von allen seinen akzeptierbaren Frauen abgelehnt worden, bleibt also ledig. Dann haben wir ein stabiles Matching  $\mu$ :

$$\mu = \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \\ m_2 & m_1 & m_4 & m_3 \end{array}$$

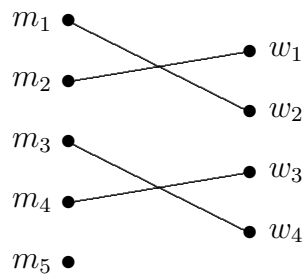


Abbildung 3.4: Output

# Kapitel 4

## Das Zuweisungsspiel

In diesem Kapitel wird das Zuweisungsspiel eingeführt. Es ist ein kooperatives Spiel, in dem Geld eine auffällige Rolle spielt. Man modelliert Geld als eine kontinuierliche Variable anstatt einer diskreten. Shapley und Shubik [11] zeigten, dass die stabilen Payoffs des Zuweisungsspiels die Lösung des dualen Programmes zu dem Linearen Programm für gewichtete bipartite Matchings sind. Kuhn [7] entwickelte die populäre Ungarische Methode, mit der man ein maximales gewichtetes Matching in bipartiten Graphen finden kann. A. Frank [1] hat diese Methode gut erklärt. Als Lösungsmethode des Zuweisungsspiels besprechen wir im Abschnitt 4.4 die Version der Ungarischen Methode von A. Frank [1].

### 4.1 Modelldarstellung

Am Markt gibt es  $m$  Verkäufer und  $n$  Käufer. Verkäufer  $i$  schätzt den Wert seines Objektes auf  $c_i$  Euro. Doch schätzt Käufer  $j$  den Wert des gleichen Objektes auf  $r_{ij}$  Euro. Wenn es zum Preis  $p$  gehandelt wird, ist das Payoff von  $j$   $v_j = r_{ij} - p$  und das von  $i$   $u_i = p - c_i$ .  $\alpha_{ij} = \max\{u_i + v_j, 0\} = \max\{r_{ij} - c_i, 0\}$  ist der potentielle Gewinn des Geschäftes zwischen  $i$  und  $j$ . Wie ist der Gesamtgewinn zu maximieren?

Formal gibt es zwei endliche disjunkte Mengen  $P$  und  $Q$ :  $P = \{1, \dots, i, \dots, m\}$  ist die Menge der Verkäufer und  $Q = \{1, \dots, j, \dots, n\}$  ist die Menge der Käufer. Manchmal nennt man die Mitglieder von  $P$   $P$ -Agenten und die von  $Q$   $Q$ -Agenten. Im Zusammenhang mit jeder möglichen Partnerschaft  $(i, j)$  in  $P \times Q$  steht eine nicht-negative reelle Zahl  $\alpha_{ij}$ , der Wert der Koalition  $\{i, j\}$  von einem  $P$ -Agenten und einem  $Q$ -Agenten. Die charakteristische Funktion  $\nu$  ist gegeben durch:

$\nu(S) = \alpha_{ij}$ , falls  $S = \{i, j\}$  für  $i \in P$  und  $j \in Q$ ;

$\nu(S) = 0$ , falls  $S$  nur  $P$ -Agenten oder  $Q$ -Agenten enthält;

$\nu(S) = \max\{\nu(i_1, j_1) + \nu(i_2, j_2) + \dots + \nu(i_k, j_k)\}$ , falls  $\{(i_1, j_1), (i_2, j_2), \dots, (i_k, j_k)\} \subseteq S_P \times S_Q$ , wo  $S_P$  (bzw.  $S_Q$ ) die Menge der  $P$ -Agenten (bzw.  $Q$ -Agenten) in  $S$  ist, und  $k \leq \min(|S_P|, |S_Q|)$ .

Das Problem, für eine gegebene Matrix  $\alpha = (\alpha_{ij})$  den Wert der charakteristischen Funktion  $\nu(N)$  zu berechnen, nennt man das *optimale Zuweisungsproblem*, oder einfach das Zuweisungsproblem. Damit werden die Spiele in entsprechender Form Zuweisungsspiele genannt.

## 4.2 Die Spielregeln

Wir beschäftigen uns mit einem Markt, der aus vielen Verkäufern und vielen Käufern besteht, und jeder Verkäufer besitzt und jeder Käufer benötigt genau ein einzelnes Objekt.

Es handelt sich bei den Spielregeln darum, dass jedes Paar  $(i, j) \in P \times Q$  gemeinsam  $\alpha_{ij}$  bekommt, und jede größere Koalition nur wertvoll ist, wenn sie sich in solche Paare organisieren kann. Die Elemente jeder Koalition können ihren gemeinsamen Wert auf beliebigem Weg aufteilen.

Eine Zuteilung des Spiels ist ein nicht-negativer Vektor  $(u, v) \in \mathbb{R}^m \times \mathbb{R}^n$ , so dass  $\sum_{i \in P} u_i + \sum_{j \in Q} v_j = \nu(P \cup Q)$  gilt. Der einfachste Weg, dies zu interpretieren, ist, dass man die Quantitäten  $\alpha_{ij}$  als Menge von Geld betrachtet und annimmt, dass die Präferenzen der Agenten sich nur nach monetären Payoffs richten. Demnach setzen wir voraus, dass die Präferenzen unstrikt sein können.

## 4.3 Zulässigkeit und Stabilität von Outcomes

Hier ist  $R = \emptyset$ . In der Praxis wird  $\alpha_{ij} \geq 0$  erfordert. So definieren wir  $\alpha = \max\{a_{ij} + b_{ij}, 0\}$ .

Das Zuweisungsspiel ist nichts anderes als das gewichtete bipartite Matching. So betrachten wir folgende Lineare Programmierung (LP) für das gewichtete



bipartite Matching  $\mu$  mit der Gewichtsfunktion  $\alpha$ , wobei  $x$  die Matrix von  $\mu$  ist ( $x_{ij} = 1$ , wenn  $i$  und  $j$  miteinander gematcht sind; und  $x_{ij} = 0$ , sonst):

$$(LP) \quad \max \quad \sum_{i,j} \alpha_{ij} \cdot x_{ij}$$

$$\text{unter} \quad (a) \quad \sum_i x_{ij} \leq 1$$

$$(b) \quad \sum_j x_{ij} \leq 1$$

$$(c) \quad x_{ij} \geq 0$$

und das duale Programm zu (LP):

$$(DLP) \quad \min \quad \sum_{i \in P} u_i + \sum_{j \in Q} v_j$$

$$\text{unter} \quad (a') \quad u_i \geq 0, \quad v_j \geq 0$$

$$(b') \quad u_i + v_j \geq \alpha_{ij}$$

Beim (DLP) geht es um Geld, also um die Payoffs.  $u$  und  $v$  sind die Payoffs der Verkäufer bzw. Käufer. Da (LP) eine Lösung hat, muss also (DLP) auch eine optimale Lösung haben. Lemma 2.6.3 liefert, dass beide Zielfunktionen den gleichen Wert haben müssen. Das bedeutet, dass

$$\sum_{i \in P} u_i + \sum_{j \in Q} v_j = \sum_{P \times Q} \alpha_{ij} \cdot x_{ij}. \quad (*)$$

gilt, wenn  $\mu$  ein optimales Matching ist und  $(u, v)$  eine optimale Lösung von (DLP) ist.

Ein Payoff  $(u, v)$  mit  $(*)$  heißt *zulässig*. Ein zulässiges Outcome ist *stabil*, wenn  $u_i + v_j \geq \alpha_{ij}$  für alle  $(i, j) \in P \times Q$  gilt. Es ist genau die duale Restriktion von (LP) auf Seite 21.

## 4.4 Die Ungarische Methode

Als Lösungsmethode für Zuweisungsspiele stellen wir einen populären und wesentlichen kombinatorischen Algorithmus, die sogenannten *Ungarische Methode*, vor. Der Algorithmus stammt von Kuhn [7] und beruht auf der Idee von König und Egerváry, demnach nannte Kuhn ihn Ungarische Methode. Dieser Algorithmus erzeugt ein optimales gewichtetes Matching für einen bipartiten Graphen  $G = (P \dot{\cup} Q, E)$  in  $\mathcal{O}(n^3)$  Zeit, wobei  $n = \max(|P|, |Q|)$ . Um die Ungarische Methode wiederzugeben, fügen wir außerdem einige Begriffe und Ergebnisse von gewichteten Matchings hinzu.

Wenn wir der kleineren Knotenmenge geeignet viele Dummy-Knoten hinzufügen und die entsprechenden Kantengewichte auf 0 setzen, erhalten wir  $|P| = |Q|$ . So

sprechen wir nur über den bipartiten Graphen  $G$  mit Knotenmenge  $V = P \dot{\cup} Q$ , wobei  $P = \{1, \dots, i, \dots, n\}$  und  $Q = \{1, \dots, j, \dots, n\}$ . Die nicht-negative, quadratische  $(n \times n)$ -Matrix  $\alpha = (\alpha_{ij})$  wird als Gewicht auf  $G$  interpretiert.

Sei  $\beta$  das Gewicht eines optimalen Matchings in  $G$ ,  $\mu$  ein perfektes Matching von  $G$  und  $(u, v)$  zulässig. (Mit  $Z$  bezeichnet man den dualen Zulässigkeitsbereich.) Durch Addieren über alle Kanten des Matchings  $\mu$  haben wir:

$$\alpha(\mu) \leq \beta \leq \sum_{i=1}^n (u_i + v_i) \quad (\star).$$

Wenn wir ein Payoff  $(u, v) \in Z$  und ein Matching  $\mu$  finden, so dass das Gleichheitszeichen in  $(\star)$  gilt, dann muss dieses Matching optimal sein. Deshalb definieren wir einen Teilgraphen  $H_{u,v}$  bezüglich Gleichheit in der Summe.  $H_{u,v}$  benutzt Gleichheit der Summanden.

**Definition 4.4.1.** Sei  $(u, v) \in Z$ . Orientieren wir alle Kanten  $(i, j)$  mit  $u_i + v_j = \alpha_{ij}$  wie folgt:

- von  $Q$  nach  $P$ , wenn  $i$  und  $j$  gematcht sind;
- von  $P$  nach  $Q$ , sonst.

Dann erhalten wir einen Teilgraphen  $H_{u,v}$ , der nur die obigen dichten Kanten besitzt. Man nennt  $H_{u,v}$  den **Digraphen der dichten Kanten** für das Outcome  $(u, v; \mu)$ .

Ohne Beweis (siehe [15]) führen wir ein Lemma ein, um die Idee der Ungarischen Methode zu erklären.

**Lemma 4.4.2.** Sei  $H = H_{u,v}$  der Graph der dichten Kanten für das Payoff  $(u, v) \in D$ .  $\sum_{i=1}^n (u_i + v_i) = \beta$  gilt genau dann, wenn  $H$  ein perfektes Matching hat.

In diesem Falle ist jedes perfekte Matching von  $H$  ein optimales Matching von  $G$ .

### Idee der Ungarischen Methode

Die Ungarische Methode beginnt mit einem Payoff  $(u, v) \in D$  mit  $u_i = \max_j \alpha_{i,j}$ ,  $i = 1, \dots, n$ , und  $v_1 = \dots = v_n = 0$ . Wenn der entsprechende Graph der dichten Kanten ein perfektes Matching enthält, ist unser Problem gelöst. Sonst verbessert

der Algorithmus mit  $\text{ALTERNATE}(\mathcal{P})$  das Matching durch einen alternierenden Weg  $\mathcal{P}$  in  $H_{u,v}$  oder bestimmt durch eine Breitensuche  $\text{BFS}(H_{u,v}, i)$  ohne Abbruchkriterium einen alternierenden Baum mit Knotenmenge  $\bar{P} \dot{\cup} \bar{Q}$ , dessen Wurzel ein ungematchter Knoten  $i \in P$  ist.

Sei nun  $\delta = \min\{u_i + v_j - \alpha_{ij} \mid i \in \bar{P}, j \notin \bar{Q}\}$ . Dann modifiziert der Algorithmus

$$u'_i = \begin{cases} u_i - \delta & , \text{ falls } i \in \bar{P} \\ u_i & , \text{ falls } i \notin \bar{P} \end{cases}$$

$$v'_j = \begin{cases} v_j + \delta & , \text{ falls } j \in \bar{Q} \\ v_j & , \text{ falls } j \notin \bar{Q}. \end{cases}$$

Es wird die Summe  $\sum(u_i + v_j)$  reduziert oder mindestens eine Kante  $(i, j)$  mit  $i \in \bar{P}$  und  $j \notin \bar{Q}$  erzeugt, welche in dem neuen Digraphen der dichten Kanten  $H_{u',v'}$  steht. Das Verfahren wird wiederholt, bis wir einen Graphen  $H$  erhalten, der ein perfektes Matching besitzt.  $\square$

---

**Algorithmus 3** Die Ungarische Methode (Version nach A. Frank)

---

```

1: procedure WEIGHTEDBIPARTITEMATCHING
2:   for all  $i \in P$  do
3:      $u_i \leftarrow \max\{\alpha_{ij} \mid j \in Q\}$ 
4:   end for
5:   while  $(u, v; x)$  unstabil do
6:     if  $\exists$  augmentierender Weg  $\mathcal{P}$  in  $H_{u,v}$  then
7:        $\text{ALTERNATE}(\mathcal{P})$ 
8:     else
9:        $i \leftarrow$  ungematchter Knoten aus  $P$ 
10:       $\text{HUNGARIANUPDATE}(i)$ 
11:    end if
12:  end while
13: end procedure

14: procedure HUNGARIANUPDATE( $i$ )
15:   $\bar{P} \dot{\cup} \bar{Q} \leftarrow \text{BFS}(H_{u,v}, i)$ 
16:   $\delta \leftarrow \min\{u_i + v_j - \alpha_{ij} \mid i \in \bar{P}, j \notin \bar{Q}\} > 0$ 
17:  for all  $i \in \bar{P}$  do
18:     $u_i \leftarrow u_i - \delta$ 
19:  end for
20:  for all  $j \in \bar{Q}$  do
21:     $v_j \leftarrow v_j + \delta$ 
22:  end for
23: end procedure

```

---

**Beispiel 4.5.** Die Gewichtsfunktion  $\alpha$  ist gegeben durch

$$\alpha = \begin{pmatrix} 3 & 8 & 9 & 1 & 6 \\ 1 & 4 & 1 & 5 & 5 \\ 7 & 2 & 7 & 9 & 2 \\ 3 & 1 & 6 & 8 & 8 \\ 2 & 6 & 3 & 6 & 2 \end{pmatrix}$$

Starte mit  $u = (9, 5, 9, 8, 6)$ ,  $v = (0, 0, 0, 0, 0)$  und  $\mu = \emptyset$ .

- *Erster Schritt:*

Wähle einen augmentierenden Weg  $\mathcal{P} = (1, 3')$ . Durch Aufruf von  $\text{ALTERNATE}(\mathcal{P})$  werden die Matching- und Nicht-Matching-Kanten ausgetauscht. Dann haben wir das Outcome:  $u = (9, 5, 9, 8, 6)$ ,  $v = (0, 0, 0, 0, 0)$  und  $\mu = \{(1, 3')\}$ .

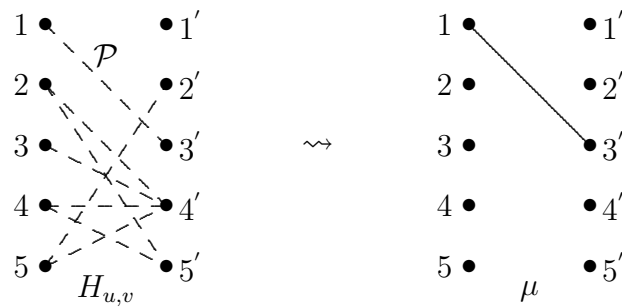


Abbildung 4.1: Erster Schritt

- *Zweiter-Vierter Schritt:*

Da ein augmentierender Weg existiert, verbessert man das Outcome wie im ersten Schritt. Nach der Augmentierung haben wir folgendes Outcome:  $u = (9, 5, 9, 8, 6)$ ,  $v = (0, 0, 0, 0, 0)$ ,  $\mu = \{(1, 3'), (2, 5'), (3, 4'), (5, 2')\}$ .

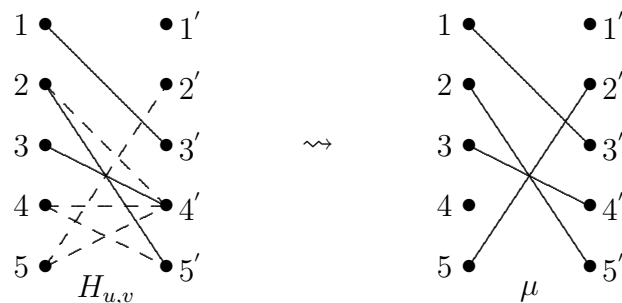


Abbildung 4.2: Vierter Schritt

- *Fünfter Schritt:*

Es gibt keinen augmentierenden Weg mehr. Setze  $i$  auf 4. HUNGARIANUPDATE wird aufgerufen. Danach haben wir  $\bar{P} = \{2, 3, 4\}$ ,  $\bar{Q} = \{4', 5'\}$  und  $\delta = 1$ . Eine neue Kante  $(2, 2')$  entsteht. Das Outcome wird bestimmt durch:  $u = (9, 4, 8, 7, 6)$ ,  $v = (0, 0, 0, 1, 1)$  und  $\mu = \{(1, 3'), (2, 5'), (3, 4'), (5, 2')\}$ .

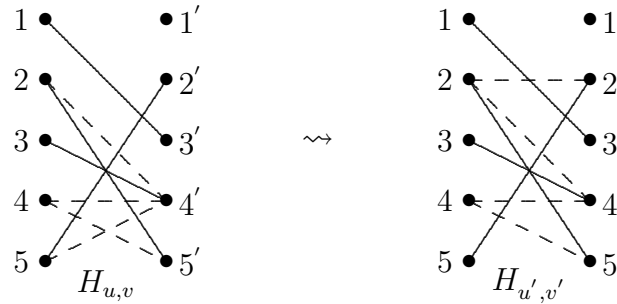


Abbildung 4.3: Fünfter Schritt

- *Sechster Schritt:*

In diesem Schritt wird das Outcome wieder durch Aufruf von HUNGARIANUPDATE augmentiert.  $\bar{P} = \{2, 3, 4, 5\}$ ,  $\bar{Q} = \{2', 4', 5'\}$  und  $\delta = 1$ . Somit erhalten wir die neuen Kanten  $(3, 1')$ ,  $(3, 3')$  und  $(4, 3')$ . Die gematchten Paare bleiben so wie im fünften Schritt. Was sich verändert hat, ist nur das Payoff. Das Outcome sieht dann so aus:  $u = (9, 3, 7, 6, 5)$ ,  $v = (0, 1, 0, 2, 2)$  und  $\mu = \{(1, 3), (2, 5), (3, 4), (5, 2)\}$ .

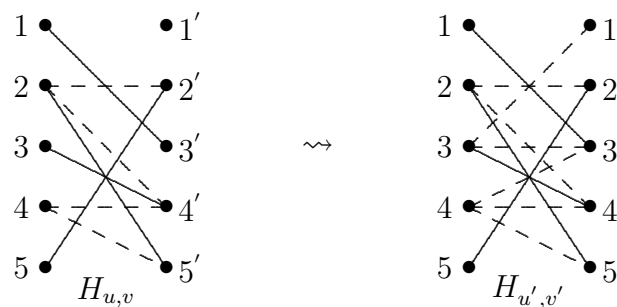


Abbildung 4.4: Sechster Schritt

- *Siebter Schritt:*

Es gibt einen augmentierenden Weg  $\mathcal{P} = (4, 4', 3, 1')$ . So wird ALTERNATE( $\mathcal{P}$ ) aufgerufen.

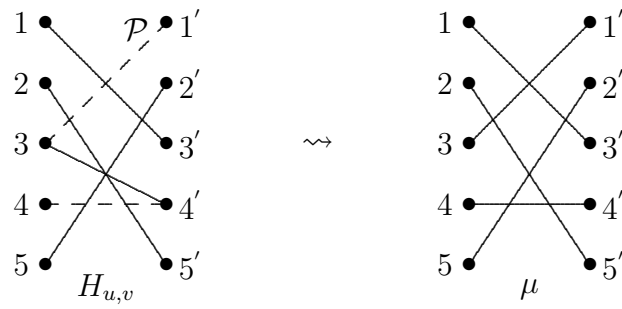


Abbildung 4.5: Siebter Schritt

*Output:*

$$\begin{aligned}
 u &= (9, 3, 7, 6, 5) \\
 v &= (0, 1, 0, 2, 2) \\
 \mu &= \{(1, 3), (2, 5), (3, 1), (4, 4), (5, 2)\}
 \end{aligned}$$

# Kapitel 5

## Das verallgemeinerte Modell

Eriksson und Karlander [5] führten das Heiratsproblem und das Zuweisungsspiel zu einem verallgemeinerten Modell zusammen und bewiesen mit einem pseudopolynomiellen Algorithmus, dass ein stabiles Outcome immer existiert. Sotomayor [8] lieferte einen anderen Beweis für die Existenz von stabilen Outcomes für das gleiche Modell von Eriksson und Karlander, also für das im folgenden zu beschreibende verallgemeinerte Modell. Fujishige und Tamura [2] waren der Ansicht, dass der Beweis nicht konstruktiv ist. Dagegen haben wir aus den Schlüssellemmas von Sotomayor [8] einen polynomiellen Algorithmus entwickelt.

### 5.1 Modelldarstellung

Es ist ein sogenanntes *One-to-One Matching*, an dem zwei endliche disjunkte Mengen  $P = \{1, \dots, i, \dots, m\}$  und  $Q = \{1, \dots, j, \dots, n\}$  von Spielern beteiligt sind, die wir Firmen und Arbeiter nennen. Sie können auch Assistenzärzte und Krankenhäuser, Verkäufer und Käufer, u.ä. sein. Jede Firma stellt genau einen Arbeiter ein.

Die Spieler werden in zwei Klassen  $F$  und  $R$  aufgeteilt, also ist  $F \dot{\cup} R = P \dot{\cup} Q$ . Mit den Spielern aus  $F$  wird ein flexibles Gehalt verbunden. Ihre Gehälter können ausgehandelt werden. Die Spieler aus  $R$  arbeiten nach Tarif, also geht es dabei um festes Gehalt. Wie im Abschnitt 2.4 erläutert, bezeichnet man mit  $F^*$  die Menge aller flexiblen Paare  $(i, j) \in P \times Q$  und mit  $R^*$  die Menge aller Paare  $(i, j) \in P \times Q$ , wobei mindestens ein Spieler von beiden tariflich gebunden ist.

Für jedes Paar  $(i, j) \in P \times Q$  gibt es ein Paar von nicht-negativen reellen Zahlen  $(a_{ij}, b_{ij})$ , wo  $a_{ij} + b_{ij}$  als Ertragskraft vom Paar  $(i, j)$  gesehen werden kann.

Die Produktivität in einer flexiblen Partnerschaft, d.h.,  $(i, j) \in F^*$ , wird nach  $u_i + v_j = a_{ij} + b_{ij}$  aufgeteilt, und in einer Tarifpartnerschaft, also  $(i, j) \in R^*$ , nach  $u_i = a_{ij}$  und  $v_j = b_{ij}$ .

Durch Ergänzung von Dummy-Knoten und -Kanten sieht das Problem dann wie folgt aus:  $G = (P \dot{\cup} Q, E)$  ist also ein bipartiter Graph mit zwei nicht-negativen Gewichtsfunktionen  $a, b : E \rightarrow \mathbb{R}$  und  $F \dot{\cup} R = P \dot{\cup} Q$  eine andere Partition der Knoten. Die Zulässigkeit und Stabilität von Outcomes (bzw. Payoffs) haben wir im Abschnitt 2.4 definiert.

## 5.2 Vorbereitungen

In den folgenden Abschnitten wird ein Algorithmus, den wir aus der Beweisidee von Sotomayor [8] konstruiert haben, beschrieben. Der Algorithmus erzeugt ein stabiles Outcome für das verallgemeinerte Modell in Zeit  $\mathcal{O}(n^4)$ . Dabei soll das Outcome in jedem Schritt des Algorithmus spezielle Eigenschaften erfüllen.

**Definition 5.2.1.** Die Menge  $S$  wird definiert durch

$$S := \{(u, v; \mu) \mid (u, v; \mu) \text{ ist ein zulässiges Outcome, in dem } i \text{ ungematcht sein muss, wenn } (i, j) \text{ } (u, v; \mu) \text{ blockiert}\}.$$

**Bemerkung:** Für die Outcomes in  $S$  gibt es keine gematchte Firma, die blockierende Paare erzeugt. Die Menge  $S$  ist nicht leer, denn das Outcome  $(0, 0; \emptyset)$ , in dem jeder Spieler ungematcht ist, liegt in  $S$ .

**Lemma 5.2.2.** Für festes Matching  $\mu$  ist die Menge  $V_\mu$  definiert durch

$$V_\mu := \{v \in \mathbb{R}^n \mid \exists u \in \mathbb{R}^m : (u, v; \mu) \in S\}.$$

Wenn für dieses Matching  $\mu$  ein Payoff  $(u, v)$  mit  $(u, v; \mu) \in S$  existiert, dann ist die Menge  $V_\mu$  kompakt.

**Beweis:** Für ein zulässiges Outcome ist entweder  $v_j = 0$ , (wenn  $j$  ungematcht ist), oder  $u_i + v_j = a_{ij} + b_{ij}$ , (wenn  $j$  und  $i$  miteinander gematcht sind). Daher gilt  $|v_j| \leq \max_{i,j} (a_{ij} + b_{ij})$ . Da  $n$  endlich ist, ist  $v$  beschränkt.

Wir beweisen die Abgeschlossenheit.



1. Fall  $F^* \cap \mu = \emptyset$ , d.h., alle gematchten Paare  $(i, j)$  liegen in  $R^*$ .  
Aus der Zulässigkeit des Outcomes folgt:

$$\begin{aligned} u_i &= 0 & , & \text{ falls } i \text{ ungematcht ist;} \\ v_j &= 0 & , & \text{ falls } j \text{ ungematcht ist;} \\ u_i &= a_{ij} \quad \text{und} \quad v_j = b_{ij} & , & \text{ falls } (i, j) \in \mu \end{aligned}$$

Damit ist das Outcome  $(u, v; \mu)$  eindeutig bestimmt. Für  $\mu$  ist  $(u, v; \mu)$  offenbar das einzige Outcome, das in  $S$  liegt. Also ist  $V_\mu$  in diesem Fall kompakt.

2. Fall  $F^* \cap \mu \neq \emptyset$ .

Es seien  $I = \{i \in P \mid \exists j \in Q : (i, j) \in F^* \cap \mu\}$  und  
 $J = \{j \in Q \mid \exists i \in P : (i, j) \in F^* \cap \mu\}$ .

Sei  $\{v^k\}_{k \in \mathbb{N}} \subseteq V_\mu$  eine Folge, die gegen ein  $v \in \mathbb{R}^n$  konvergiert. Es ist zu zeigen, dass  $v$  wieder in  $V_\mu$  liegt.

$v_k \rightarrow v$ , d.h., es gibt zu jedem  $\varepsilon > 0$  einen Index  $k_0(\varepsilon)$ , so dass  $\|v_k - v\| < \varepsilon$  bleibt für alle  $k \geq k_0(\varepsilon)$ . Mit Begründung, dass alle Normen auf einem endlichdimensionalen Vektorraum äquivalent sind, betrachten wir die Norm  $\|\cdot\|$  im Sinne der euklidischen Norm, also

$$\|v_k - v\| = \sqrt{\sum_{j=1}^n (v_j^k - v_j)^2} < \varepsilon.$$

Demnach haben wir

$$|v_j^k - v_j| < \varepsilon.$$

Betrachten wir nun das Outcome  $(u, v; \mu)$  mit

$$u_i = \begin{cases} 0 & , \text{ falls } i \text{ ungematcht ist;} \\ a_{ij} + b_{ij} - v_j & , \text{ falls } \exists j \in Q : (i, j) \in \mu. \end{cases}$$

Wie im 1. Fall gezeigt, sind die Payoffs von  $i \in P \setminus I$  und von  $j \in Q \setminus J$  für festes Matching eindeutig festgelegt, d.h.,

$$u_i^1 = u_i^2 = \dots = u_i^k = \dots = \text{konst}, \quad \forall i \in P \setminus I \quad \text{und}$$

$$v_j^1 = v_j^2 = \dots = v_j^k = \dots = \text{konst}, \quad \forall j \in Q \setminus J.$$

Dann gilt  $u_i = 0 = u_i^k$ , wenn  $i$  nicht gematcht ist. Wenn  $i$  gematcht ist, d.h.,  $\exists j \in Q : (i, j) \in \mu$ , dann haben wir für  $i \in P \setminus I$

$$\begin{aligned} u_i + v_j &= a_{ij} + b_{ij} = u_i^k + v_j^k \\ \implies |u_i - u_i^k| &= |v_j^k - v_j| < \varepsilon \\ \implies u_i &= u_i^k \end{aligned}$$

Zusammenfassung

$$\begin{aligned} u_i &= u_i^k & , & \text{ falls } i \in P \setminus I \\ |u_i - u_i^k| &< \varepsilon & , & \text{ falls } i \in I \\ v_j &= v_j^k & , & \text{ falls } j \in Q \setminus J \\ |v_j - v_j^k| &< \varepsilon & , & \text{ falls } j \in J \end{aligned}$$

Daraus folgt

- (a) Wegen der Zulässigkeit von  $(u^k, v^k; \mu)$  ist  $(u, v; \mu)$  auch zulässig.  
 (b) Es sei  $i$  eine gematchte Firma. Wenn für  $(u^k, v^k; \mu)$   $i$  keine blockierenden Paare erzeugt, d.h.,  $\forall j \in Q$  :

$$\begin{aligned} u_i^k + v_j^k &\geq a_{ij} + b_{ij} & , & \text{ falls } (i, j) \in F^* \\ u_i^k &\geq a_{ij} \quad \text{oder} \quad v_j^k &\geq b_{ij} & , \text{ falls } (i, j) \in R^* , \end{aligned}$$

dann erfüllt das Outcome  $(u, v; \mu)$  obige Ungleichungen auch. Damit erzeugt  $i$  in  $(u, v; x)$  keine blockierenden Paare.

Daher ist  $(u, v; \mu) \in S$ , also ist  $v \in V_\mu$ . Folglich ist  $V_\mu$  kompakt.  $\square$

**Bemerkung:**

- (a) Nach dem Extremalsatz existiert für festes Matching  $\mu$  ein Outcome  $(\hat{u}, \hat{v}; \mu) \in S$  mit

$$\sum_{j \in Q} \hat{v}_j \geq \sum_{j \in Q} v_j, \quad \forall (u, v; \mu) \in S,$$

da  $V_\mu$  kompakt ist und die Funktion

$$f : V_\mu \rightarrow \mathbb{R} \quad \text{mit} \quad f(v) = \sum_{j \in Q} v_j = \sum_{j=1}^n v_j$$

stetig ist.

- (b) Die Menge

$$\begin{aligned} V &= \{v \in \mathbb{R}^n \mid (u, v; \mu) \in S \text{ für ein } u \in \mathbb{R}^m \text{ und ein Matching } \mu\} \\ &= \bigcup \{V_\mu \mid \mu \text{ ist ein Matching}\} \end{aligned}$$

ist als endliche Vereinigung aller  $V_\mu$  wieder kompakt.

Somit gibt es ein Outcome  $(\tilde{u}, \tilde{v}; \mu)$  mit dem maximalen gesamten Payoff  $\sum_{j \in Q} \tilde{v}_j$ , d.h.

$$\sum_{j \in Q} \tilde{v}_j \geq \sum_{j \in Q} v_j \quad \text{für alle } (u, v; \mu) \in S.$$

**Definition 5.2.3.** Für ein zulässiges Outcome  $(u, v; \mu)$  konstruieren wir einen **Augmentierungsgraphen**  $G_{(u, v; \mu)}$  als einen Teilgraphen von  $G$  mit der Kantenmenge

$$E_{(u, v; \mu)} := \{(j, i) | (i, j) \in \mu\} \cup \{(i, j) | j \in D_i(u, v, \mu)\},$$

wobei

$$\begin{aligned} D_i(u, v, \mu) &:= \{\text{die blockierenden Traumpartner von } i\} \\ &\cup \{j \in Q | (i, j) \in F^* \text{ und } (i, j) \notin \mu \text{ und } u_i + v_j = a_{ij} + b_{ij}\} \\ &\cup \{j \in Q | (i, j) \in R^* \text{ und } (i, j) \notin \mu \text{ und } u_i = a_{ij} \text{ und } v_j < b_{ij}\}. \end{aligned}$$

**Bemerkung:** Wenn  $j \in D_i(u, v, \mu)$  und  $(i, j) \in R^*$  kein blockierendes Paar ist, dann können wir ein zulässiges Outcome konstruieren, in dem  $i$  und  $j$  mit Payoff  $u_i = a_{ij}$  und  $v_j = b_{ij}$  miteinander gematcht sind, die alten Matchingpartner von  $i$  und  $j$  ungematcht sind, wenn sie existieren, und die anderen sich nicht ändern. Somit verbessert sich das Payoff  $v_j$  im neuen Outcome.

Das folgende Lemma ist eine kleine Modifikation von Sotomayor [[8], Lemma 1].

**Lemma 5.2.4.** Sei  $(u, v; \mu) \in S$ . Gebe es eine ungematchte Firma  $i_1 \in P$  mit  $D_i(u, v, \mu) \neq \emptyset$ . Dann existiert für das Matching  $\mu$  ein Payoff  $(u', v')$  mit  $(u', v'; \mu) \in S$ , so dass es im Augmentierungsdigraphen  $G_{(u', v'; \mu)}$  einen gerichteten Weg  $\mathcal{P}$  von  $i_1$  zu einem Spieler aus  $R$ , einem ungematchten flexiblen Arbeiter oder einer flexiblen Firma mit Payoff Null gibt.

**Beweis:** Wenn es im Augmentierungsdigraphen  $G_{(u, v; \mu)}$  vom Outcome  $(u, v; \mu)$  einen solchen Weg gibt, dann ist der Satz bewiesen. Sonst modifizieren wir den Payoff  $(u, v)$ , bis im Augmentierungsdigraphen  $G_{(u', v'; \mu)}$  vom neuen Outcome  $(u', v'; \mu)$  ein solcher Weg entsteht.

Sei  $j_1 \in D_i(u, v, \mu)$  und  $\bar{P}$  (bzw.  $\bar{Q}$ ) die Menge aller Firmen (bzw. Arbeiter), die von  $j_1$  aus erreichbar sind. Dann ist  $\bar{Q}$  wegen  $j_1 \in Q$  nicht leer.

Wenn kein solcher Weg im Augmentierungsdigraphen  $G_{(u, v; \mu)}$  existiert, dann gibt es in  $\bar{P} \cup \bar{Q}$  keinen Spieler aus  $R$ . Alle Elemente aus  $\bar{Q}$  sind gematcht und  $u_i > 0$  gilt für alle  $i \in \bar{P}$ . Nun betrachten wir ein Paar  $(i, j) \in P \times Q$  mit  $i \in \bar{P}$  und  $j \in Q \setminus \bar{Q}$ , dann haben wir:

1.  $(i, j) \notin \mu$  und  $j \notin D_i(u, v, \mu)$ ;  
(Falls  $(i, j) \in \mu$ , dann gibt es eine Kante von  $j$  nach  $i$ ; falls  $j \in D_i(u, v, \mu)$ , dann gibt es eine Kante von  $i$  nach  $j$ . In beiden Fällen sind  $i$  und  $j$  gleichzeitig von  $j_1$  aus erreichbar oder nicht.)

2.  $(i, j)$  ist kein blockierendes Paar, (da  $(u, v, \mu) \in S$  und  $i$  gematcht ist);  
Es bedeutet  $u_i \geq a_{ij} + b_{ij} - v_j, \forall j \in Q$  und, wenn  $j \in R$ , ist  $u_i \geq a_{ij}$  oder  $v_j \geq b_{ij}$ .
3.  $u_i > a_{ij} + b_{ij} - v_j$  und, wenn  $j \in R$  und  $v_j < b_{ij}$ , ist  $u_i > a_{ij}$  (nach 1, 2 und der Definition von  $D_i(u, v, \mu)$ )

Seien

$$\begin{aligned}\tilde{F} &:= (\bar{P} \times (Q \setminus \bar{Q})) \cap F^* \text{ und} \\ \tilde{R} &:= \{(i, j) \in \bar{P} \times ((Q \setminus \bar{Q}) \cap R) \mid u_i > a_{ij} \text{ und } v_j < b_{ij}\}.\end{aligned}$$

Für alle  $i \in \bar{P}$  gibt es dann ein  $\lambda > 0$  mit  $u_i - \lambda \geq 0$  und

$$\begin{aligned}u_i - \lambda &\geq a_{ij} && \text{für alle } (i, j) \in \tilde{R} \text{ und} \\ u_i + v_j - \lambda &\geq a_{ij} + b_{ij} && \text{für alle } (i, j) \in \tilde{F}.\end{aligned}$$

Nun konstruieren wir ein neues Outcome, das wieder in  $S$  liegt, durch

$$\begin{aligned}u'_i &= \begin{cases} u_i - \delta & , \text{ falls } i \in \bar{P} \\ u_i & , \text{ falls } i \notin \bar{P} \end{cases} \\ v'_j &= \begin{cases} v_j + \delta & , \text{ falls } j \in \bar{Q} \\ v_j & , \text{ falls } j \notin \bar{Q}, \end{cases}\end{aligned}$$

wobei

$$\begin{aligned}\delta &:= \max\{\lambda \mid u_i - \lambda \geq 0 && , \forall i \in \bar{P} && (1) \\ &u_i - \lambda \geq a_{ij} && , \forall (i, j) \in \tilde{R} && (2) \\ &u_i + v_j - \lambda \geq a_{ij} + b_{ij} && , \forall (i, j) \in \tilde{F} && (3) \quad \}.\end{aligned}$$

$\delta$  ist so gewählt worden, dass in einer von (1)-(3) das Gleichheitszeichen mindestens für eine Kante  $(i, j)$  gilt. Wenn das Gleichheitszeichen in (1) gilt, dann erhalten wir nach der Reduktion von  $u_i$  eine Firma mit Payoff Null; wenn das Gleichheitszeichen in (2) gilt, dann gelten im neuen Outcome  $u'_i = a_{ij}$  und  $v'_j < b_{ij}$ , damit ist  $j \in D_i(u', v', \mu)$ , so bekommen wir einen Arbeiter aus  $R$ . Sonst erreichen wir einen anderen flexiblen Arbeiter aus  $Q \setminus \bar{Q}$  und  $\bar{Q}$  wird vergrößert. Wenn noch kein solcher Weg existiert, dann wiederholen wir den Prozess. Nach der Bemerkung von Lemma 5.2.4 und dem Auswahlprinzip von  $\delta$  kann das Payoff  $v$  nicht immer verbessert werden. Tatsächlich wird diese Wiederholung höchstens  $|Q|$  mal getroffen. Somit ist das Verfahren endlich und endet mit einem solchen Weg.  $\square$

## 5.3 Ein Algorithmus für stabile Outcomes

In den nachfolgenden Abschnitten erklären wir Schritt für Schritt den Algorithmus.

Zuerst fügen wir der kleineren Knotenmenge geeignet viele Dummy-Knoten als flexiblen Knoten hinzu, und setzen 0 auf die entsprechenden Kantengewichte, um  $|P| = |Q|$  zu erhalten. Dann starten wir mit einem zulässigen Outcome  $(u, v; \mu) = (0, 0; \emptyset)$  und einem virtuellen Payoff  $(\bar{u}, v)$  mit

$$\bar{u}_i = \max(\{a_{ij} + b_{ij} \mid (i, j) \in F^*\} \cup \{a_{ij} \mid (i, j) \in R^*\}), \quad i \in P,$$

so dass keine Kante  $(i, j)$  ein blockierendes Paar im (unzulässigen) Outcome  $(\bar{u}, v; \mu)$  erzeugt. Also ist  $u$  die Zuteilung vom gegenwärtigen Matching und  $\bar{u}_i$  eine obere Schranke des größtmöglichen Nutzens von  $i$ . Solange das Outcome nicht stabil ist, es also ein blockierendes Paar  $(i_1, j)$  gibt, dann setzt man einen blockierenden Traumpartner von  $i_1$  auf  $j_1$ . Danach sucht man im Graphen  $G_{(u, v; \mu)}$  einen Weg  $\mathcal{P} = (i_1, j_1, i_2, j_2, \dots)$  zu einem Spieler aus  $R$ , einem ungematchten flexiblen Arbeiter, oder einer flexiblen Firma mit Payoff Null. Wenn obiger Weg nicht existiert, dann modifiziert man das Payoff  $(u, v)$  wie im Beweis für Lemma 5.2.4, bis ein solcher Weg  $\mathcal{P}$  entsteht. Danach alterniert man das Matching durch den Weg  $\mathcal{P}$  (siehe Algorithmus 6). Wenn für alle  $i \in P$   $\bar{u}_i > 0$  gilt, dass  $i$  gematcht ist, erhalten wir ein stabiles Outcome  $(u, v; \mu)$ .

---

### Algorithmus 4 Konstruktion eines stabilen Outcomes

---

```

while blockierendes Paar  $(i_1, j)$  existiert do
     $j_1 \leftarrow$  ein blockierender Traumpartner von  $i_1$ ;
     $\mathcal{P} \leftarrow$  FINDPATH( $i_1, j_1$ );
    while kein wie im Lemma 5.2.4 gezeigter Weg  $\mathcal{P} = (i_1, j_1, i_2, j_2, \dots)$  existiert
    do
        HUNGARIANUPDATE( $j_1$ );
         $\mathcal{P} \leftarrow$  BFS( $G_{(u, v; \mu)}, j_1$ );
    end while
    PATHUPDATE( $\mathcal{P}$ );
end while

```

---

### 5.3.1 Suche nach dem Weg $\mathcal{P}$

Durch Aufruf von FINDPATH( $i_1, j_1$ ) oder BFS( $G_{(u, v; \mu)}, j_1$ ) wird überprüft, ob ein wie im Lemma 5.2.4 gezeigter Weg im Graphen  $G_{(u, v; \mu)}$  existiert. Das Verfahren

$\text{BFS}(G_{(u,v;\mu)}, j_1)$  ist die Breitensuche mit dem Abbruchkriterium  $w \in F \cap P$  mit  $u_w = 0$ ,  $w \in F \cap Q$  ungematcht oder  $w \in R$ .

---

**Algorithmus 5** Suche nach dem zu augmentierenden Pfad

---

```

1: procedure FINDPATH( $i, j$ )
2:   if  $i \in R$  oder  $j \in R$  then
3:      $\mathcal{P} \leftarrow \langle i, j \rangle$ ;
4:   else
5:      $\mathcal{P} \leftarrow \text{BFS}(G_{(u,v;\mu)}, j)$ ;
6:     if  $\mathcal{P} \neq 0$  then
7:        $\mathcal{P} \leftarrow i\mathcal{P}$ ;
8:     end if
9:   end if
10: end procedure

```

---

### 5.3.2 Modifikation von Payoffs

Die Modifikation von Payoffs ist ähnlich dem Update der dualen Variablen in der Ungarischen Methode. Der Hauptunterschied ist, dass die Ungarische Methode immer die duale Zulässigkeit gewährleistet, während es sich bei Algorithmus 4 um ein virtuelles Payoff handelt.

In  $\text{HUNGARIANUPDATE}(j_1)$  wählen wir  $\delta$  wie in (1)-(3), vom Beweis für Lemma 5.2.4, und modifizieren  $\bar{u}$  für alle Knoten in  $\bar{P}$ . Wir haben also folgendes:

---

```

1: procedure HUNGARIANUPDATE( $j_1$ )
2:    $\bar{P} \dot{\cup} \bar{Q} \leftarrow \text{BFS}(G_{(u,v;\mu)}, j_1)$ 
3:    $\delta \leftarrow \min(\{u_i | i \in \bar{P}\} \cup \{u_i - a_{ij} | (i, j) \in \tilde{R}\} \cup \{u_i + v_j - a_{ij} - b_{ij} | (i, j) \in \tilde{F}\})$ 
4:   for all  $i \in \bar{P}$  do
5:      $u_i \leftarrow u_i - \delta$ 
6:      $\bar{u}_i \leftarrow u_i - \delta$ 
7:   end for
8:   for all  $j \in \bar{Q}$  do
9:      $v_j \leftarrow v_j + \delta$ 
10:  end for
11: end procedure

```

---

### 5.3.3 Augmentierung

In diesem Abschnitt erklären wir im Detail, wie ein Outcome  $(u, v; \mu)$  durch den Weg  $\mathcal{P}$  augmentiert wird. Dabei werden folgende Verfahren benutzt:

ALTERNATE verändert die Partnerschaft zwischen den Knoten entlang eines alternierenden Wegs  $\mathcal{P} = (i_1, j_1, i_2, j_2, \dots)$ , also ist jede zweite Kante in  $\mathcal{P}$  eine Matchingkante, oder erzeugt einen alternierenden Kreis (im Fall 3.3). Matching- und Nicht-Matchingkanten werden vertauscht, so dass die ehemaligen Matchingkanten Nicht-Matchingkanten werden und umgekehrt. Somit verändert sich die Anzahl der Matchingkanten nicht, wenn  $\mathcal{P}$  in einer ungematchten Firma beginnt und in einer Firma endet oder  $\mathcal{P}$  ein alternierender Kreis ist; und vergrößert sich um 1, wenn  $\mathcal{P}$  in einer ungematchten Firma beginnt und in einem ungematchten Arbeiter endet.

Für den Weg  $\mathcal{P} = (i_1, j_1, i_2, j_2, \dots)$  bedeutet die Schreibweise  $\mathcal{P}_{[i_t, j_t]}$  einen Teilweg von  $i_t$  nach  $j_t$ .

UPDATE setzt

$$\begin{aligned} \bar{u}_i \leftarrow u_i \leftarrow a_{ij} + b_{ij} - v_j, & \quad \text{falls } (i, j) \in F^* \\ \bar{u}_i \leftarrow u_i \leftarrow a_{ij} \text{ und } v_j \leftarrow b_{ij}, & \quad \text{falls } (i, j) \in R^*. \end{aligned}$$

UNMATCH( $j$ ) entfernt eine Matchingkante  $(i, j)$  und setzt  $u_i \leftarrow 0$

DISCARD( $i$ ) setzt  $\bar{u}_i \leftarrow u_i \leftarrow 0$

Die Augmentierung behandelt folgende Fälle:

Fall 1:  $\mathcal{P}$  erreicht einen ungematchten flexiblen Arbeiter  $j_s \in F$ , und in  $\mathcal{P}$  gibt es keine Spieler aus  $R$ , also ist  $\mathcal{P} = (\overbrace{i_1, j_1, \dots, i_s, j_s}^F)$  und  $j_s$  ungematcht.

Fall 2:  $\mathcal{P}$  erreicht eine flexible Firma  $p \in F$  mit Payoff Null, und in  $\mathcal{P}$  gibt es keine Spieler aus  $R$ , also ist  $\mathcal{P} = (\overbrace{i_1, j_1, \dots, i_s, j_s, p}^F)$  und  $u_p = 0$ .

Fall 3:  $\mathcal{P}$  erreicht einen Spieler aus  $R$ . Es sei  $y$  der erste Knoten in  $\mathcal{P}$ , der zu  $R$  gehört,  $s$  die erste Position, in der  $y$  vorkommt.

Fall 3.1:  $y = i_1$  oder  $y = j_1$ , also ist  $\mathcal{P} = (i_1, j_1)$  und  $(i_1, j_1) \in R^*$ .

Fall 3.2:  $y = j_s$  für ein  $s > 1$  oder  $y = i_s$ ,  $s > 1$  und  $j_s \neq j_k, \forall k < s$ , also

$$\text{ist } \mathcal{P} = (\overbrace{i_1, j_1, \dots, i_{s-1}, j_{s-1}}^F, \overbrace{i_s, j_s}^{R^*}).$$

Fall 3.3:  $y = i_s$ ,  $s > 1$  und  $j_s = j_k$  für ein  $k < s$ , also ist

$$\mathcal{P} = (\overbrace{i_1, j_1, \dots, i_k, j_k = j_s, \dots, i_{s-1}, j_{s-1}}^F, \overset{R}{i_s}, \overset{F}{j_s}).$$

Fall 3.4:  $y = i_s$ ,  $s > 1$  und  $D_{i_s} = \emptyset$ , also ist  $\mathcal{P} = (\overbrace{i_1, j_1, \dots, i_{s-1}, j_{s-1}}^F, \overset{R}{i_s})$   
und  $D_{i_s} = \emptyset$ .

Es wird folgendermaßen augmentiert:



---

**Algorithmus 6** Augmentierung durch den Weg  $\mathcal{P}$ 

---

```
1: procedure PATHUPDATE( $\mathcal{P}$ )
2:   if  $\mathcal{P} = \overbrace{(i_1, j_1, \dots, i_s, j_s)}^F$  und  $j_s$  ungematcht then ▷ Fall 1
3:     ALTERNATE( $\mathcal{P}$ )
4:     UPDATE( $i_1, j_1$ )
5:   else if  $\mathcal{P} = \overbrace{(i_1, j_1, \dots, i_{s-1}, j_{s-1}, i_s)}^F$  und  $u_{i_s} = 0$  then ▷ Fall 2
6:     ALTERNATE( $\mathcal{P}$ )
7:     UPDATE( $i_1, j_1$ )
8:     DISCARD( $i_s$ )
9:   else if  $\mathcal{P} = \overbrace{(i_1, j_1)}^{R^*}$  then ▷ Fall 3.1
10:    UNMATCH( $j_1$ )
11:    ALTERNATE( $\mathcal{P}$ )
12:    UPDATE( $i_1, j_1$ )
13:   else if  $\mathcal{P} = \overbrace{(i_1, j_1, \dots, i_{s-1}, j_{s-1})}^F \overbrace{(i_s, j_s)}^{R^*}$  then ▷ Fall 3.2
14:    UNMATCH( $j_s$ )
15:    ALTERNATE( $\mathcal{P}$ )
16:    UPDATE( $i_1, j_1$ )
17:    UPDATE( $i_s, j_s$ )
18:   else if  $\mathcal{P} = \overbrace{(i_1, j_1, \dots, i_k, j_k = j_s, \dots, i_{s-1}, j_{s-1})}^F \overbrace{(i_s, j_s)}^{R \ F}$  then ▷ Fall 3.3
19:    ALTERNATE( $\mathcal{P}_{[i_{k+1}, j_s]}$ )
20:    UPDATE( $i_s, j_s$ )
21:   else if  $\mathcal{P} = \overbrace{(i_1, j_1, \dots, i_{s-1}, j_{s-1}, i_s)}^F$  und  $D_{i_s} = \emptyset$  then ▷ Fall 3.4
22:    UNMATCH( $j_{s-1}$ )
23:    ALTERNATE( $\mathcal{P}_{[i_1, j_{s-1}]}$ )
24:    UPDATE( $i_1, j_1$ )
25:    if  $i_s$  erzeugt blockierende Paare then
26:       $j_s \leftarrow$  ein blockierender Traumpartner von  $i_s$ 
27:      PATHUPDATE( $((i_s, j_s))$ )
28:    else
29:      DISCARD( $i_s$ )
30:    end if
31:  end if
32: end procedure
```

---

## 5.4 Korrektheits- und Komplexitätsanalyse

In diesem Abschnitt diskutieren wir die Korrektheit und die Komplexität von Algorithmus 4.

**Lemma 5.4.1.** *Nach jedem Aufruf von HUNGARIANUPDATE oder PATHUPDATE reduzieren sich das Payoff  $v$  von Arbeitern und die Anzahl  $|\mu|$  der Matchingkanten nicht.*

**Beweis:** In HUNGARIANUPDATE werden die Payoffs  $v_j$ ,  $j \in \bar{Q}$  nach Zeile 9 vergrößert und der Rest von den Payoffs  $v_j$ ,  $j \in Q \setminus \bar{Q}$  ändert sich nicht.

In PATHUPDATE wird das Payoff  $v$  nur verändert, wenn eine Kante  $(i, j) \in R^*$  in  $\mathcal{P}$  ist (Fälle 3.1, 3.2 und 3.3), also nur, wenn  $(i, j) \in R^*$  und  $j \in D_i(u, v, \mu)$ . Es bedeutet, dass entweder  $j$  ein blockierender Traumpartner von  $i$  ist oder  $u_i = a_{ij}$  und  $v_j < b_{ij}$  gelten. In beiden Fällen gilt  $v_j < b_{ij}$ , so wird  $v_j$  durch UPDATE( $i, j$ ) gesteigert. Der Rest der Payoffs der Arbeiter bleibt unverändert.

Somit nimmt das Payoff  $v$  der Arbeiter nicht ab.

Nun sprechen wir über die Größe des Matchings. Das Matching ändert sich nur in PATHUPDATE. Im 1. Fall wird  $|\mu|$  um 1 erhöht. In den Fällen 2 und 3.3 verändert sich  $|\mu|$  nicht. In den Fällen 3.1 und 3.2 wird  $|\mu|$  um 1 erhöht, wenn  $j_s$  ungematcht ist; ansonsten nicht verändert. Im Fall 3.4 bleibt  $|\mu|$  unverändert, wenn  $i_s$  kein blockierendes Paar erzeugt; ansonsten wird PATHUPDATE erneut aufgerufen. Demnach reduziert sich  $|\mu|$  nicht.  $\square$

**Lemma 5.4.2.** *Nach jedem Aufruf von HUNGARIANUPDATE oder PATHUPDATE gilt*

$$u_i = \bar{u}_i \iff i \text{ ist gematcht oder } u_i = \bar{u}_i = 0.$$

**Beweis:** Wir beginnen mit  $u = 0$  und  $\bar{u}_i = \max(\{a_{ij} + b_{ij} \mid (i, j) \in F^*\} \cup \{a_{ij} \mid (i, j) \in R^*\})$ , also gilt am Anfang  $\bar{u} \geq u$ . Ferner gilt  $\bar{u}_i > u_i$ , wenn  $i$  kein Dummy-Knoten ist. Die Werten von  $u_i$  und  $\bar{u}_i$  werden durch den gleichen Wert nur ersetzt, wenn UPDATE oder DISCARD aufgerufen wird. In allen anderen Fällen wird die (Un-)Gleichheitsbeziehung zwischen  $u_i$  und  $\bar{u}_i$  nicht geändert. Deshalb ist  $u_i = \bar{u}_i$  genau dann, wenn  $i$  gematcht ist oder  $u_i = \bar{u}_i = 0$ .  $\square$

**Lemma 5.4.3.** *Nach jedem Aufruf von HUNGARIANUPDATE oder PATHUPDATE befindet sich das neue Outcome wieder in  $S$ .*

**Beweis:** Das Verfahren HUNGARIANUPDATE modifiziert Outcomes nur in  $\bar{P} \cup \bar{Q}$ . Alle Firmen  $i \in \bar{P}$  sind von  $j_1$  aus erreichbar, so müssen alle  $i \in \bar{P}$  gematcht sein.

Alle Arbeiter  $j \in \bar{Q}$  sind auch gematcht, sonst haben wir einen gewünschten Weg zu einem ungematchten tariflichen Arbeiter  $j$ , also einem Spieler aus  $R$ , wenn  $j \in R$ . Ferner ist  $\bar{P} \cup \bar{Q} \subseteq F$ , ansonsten erhalten wir einen gewünschten Weg zu einem Spieler aus  $R$ .

Nach der Modifikation gilt für jede Matchingkante  $(i, j) \in \bar{P} \times \bar{Q}$

$$u'_i + v'_j = u_i + \delta + v_j - \delta = u_i + v_j = a_{ij} + b_{ij}.$$

Damit gilt die Zulässigkeit des neuen Outcomes.

Nach dem Wahlprinzip von  $\delta$  erzeugt jede gematchte Firma  $i \in \bar{P}$  keine blockierenden Paare. Für die gematchten Firmen  $i \in P \setminus \bar{P}$  bleibt  $u_i$  unverändert, und nach Lemma 5.4.1 reduziert sich  $v_j$  für alle  $j \in Q$  nicht. Somit erzeugt jede gematchte Firma  $i \in P \setminus \bar{P}$  auch keine blockierenden Paare. Deshalb liegt das neue Outcome wieder in  $S$ .

Fall 1/2:  $i_1, j_1 \in F$  und  $j_1 \in F$  sind im neuen Matching gematcht mit Payoff  $u_{i_1} + v_{j_1} = a_{i_1 j_1} + b_{i_1 j_1}$  und für  $t = 2, 3, \dots, s$  sind  $i_t$  im alten Outcome gematcht, daher gibt es keinen blockierenden Partner in  $D_{i_t}(u, v, \mu)$  und  $j_t \in D_{i_t}(u, v, \mu)$ , so muss  $u_{i_t} = a_{i_t j_t} + b_{i_t j_t} - v_{j_t}$  sein, da  $(i_t, j_t) \in F^*$ . Also ist das neue Outcome zulässig und erzeugt für keine Firma blockierende Paare. D.h., das neue Outcome liegt wieder in  $S$ .

Fall 3.1: Es gilt im neuen Outcome  $(i_1, j_1) \in \mu \cap R^*$ ,  $u_{i_1} = a_{i_1 j_1}$  und  $v_{j_1} = b_{i_1 j_1}$ . Daraus folgt die Zulässigkeit des neuen Outcomes. Da  $j_1$  ein blockierender Traumpartner von  $i_1$  ist, erzeugt  $i_1$  keine blockierenden Paare im neuen Outcome. Nämlich ist das neue Outcome ein Element von  $S$ .

Fall 3.2/3.3: Die Zulässigkeit ist offensichtlich.  $i_s$  ist im alten Outcome gematcht, also gibt es in  $D_{i_s}(u, v, \mu)$  keine blockierenden Partner. Folglich muss im alten Outcome  $u_{i_s} = a_{i_s j_s}$  und  $v_{j_s} < b_{i_s j_s}$  gelten, da  $(i_s, j_s) \in R^*$  und  $j_s \in D_{i_s}(u, v, \mu)$ . Somit ändert sich das Payoff  $u_{i_s}$  im neuen Outcome nicht. Deswegen erzeugt  $i_s$  im neuen Outcome keine blockierenden Paare. Dann bleibt das neue Outcome in  $S$ .

Fall 3.4: Das neue Outcome ist offenbar zulässig. Es ist wieder in  $S$ , wenn  $i_s$  keine blockierenden Paare erzeugt. Anderenfalls lässt man  $i_s$  mit seinem blockierenden Traumpartner gematcht, damit erzeugt  $i_s$  im neuen Outcome keine blockierenden Paare. Dann liegt das neue Outcome wieder in  $S$ .  $\square$

**Lemma 5.4.4.** *Nach jedem Aufruf von HUNGARIANUPDATE oder PATHUPDATE ist das Payoff  $(\bar{u}, v)$  stabil, und das virtuelle Payoff  $\bar{u}$  nimmt nicht zu.*

**Beweis:** In HUNGARIANUPDATE ist  $(\bar{u}, v)$  wegen der Definition von  $\delta$  stabil. Nach Zeile 6 im Verfahren HUNGARIANUPDATE( $j_1$ ) reduziert sich  $\bar{u}_i$  nur für einige Firmen und bleibt unverändert für die restlichen Firmen.

In PATHUPDATE wird das Payoff  $(\bar{u}, v)$  nur beim Aufruf von UPDATE( $i, j$ ) und DISCARD( $i_s$ ) geändert.

Beim Aufruf von DISCARD( $i_s$ ) erzeugt  $i_s$  nach Lemma 5.4.3 keine blockierenden Paare, weil  $i_s$  gematcht war und  $\bar{u}_{i_s} = u_{i_s} = 0$  bleibt. So ist  $(\bar{u}, v)$  stabil und  $\bar{u}$  steigt sich nicht.

UPDATE( $i, j$ ):

- $(i, j) = (i_1, j_1)$  (Fälle 1, 2, 3.1, 3.2, und 3.4)  
In diesem Fall wird  $i_1$  mit seinem blockierenden Traumpartner gematcht. Nach der Definition vom blockierenden Traumpartner erzeugt  $(i, j)$  keine blockierenden Paare mehr, wenn  $i$  und  $j$  miteinander gematcht werden. Also ist  $(\bar{u}, v)$  in diesem Fall stabil.
- $(i, j) = (i_s, j_s)$  (Fälle 3.2, 3.3, und 3.4)  
Wenn  $j_s$  ein blockierender Traumpartner von  $i_s$  ist (Fall 3.4), dann ist  $(\bar{u}, v)$ , wie oben gezeigt, stabil. Sonst gilt nach Lemma 5.4.2 und Definition von  $D_i(u, v, \mu)$

$$\begin{aligned} \bar{u}_i = u_i = a_{ij} + b_{ij} - v_j & \quad , \quad \text{falls } (i, j) \in F^* \\ \bar{u}_i = u_i = a_{ij}, \quad v_j < b_{ij} & \quad , \quad \text{falls } (i, j) \in R^*. \end{aligned}$$

Nach Lemma 5.4.3 hatte  $i$  keinen blockierenden Partner. Das Verfahren UPDATE( $i, j$ ) ändert  $\bar{u}_i$  nicht und erhöht  $v_j$ , wenn  $(i, j) \in R^*$ . Damit hat UPDATE( $i, j$ ) keinen Einfluß auf die Stabilität von  $(\bar{u}, v)$ . Vor dem Aufruf von UPDATE( $i, j$ ) war  $\bar{u}_i$  dual zulässig, und durch Aufruf von UPDATE( $i, j$ ) verändert sich  $\bar{u}_i$  zum minimalen Wert, der die duale Zulässigkeit garantiert. Demnach nimmt  $\bar{u}$  nicht zu.  $\square$

**Lemma 5.4.5.** *Nach dem Aufruf von DISCARD( $i_s$ ) wird  $i_s$  nie wieder gematcht.*

**Beweis:** Wenn DISCARD( $i_s$ ) im Fall 2 aufgerufen wird, dann war  $i_s$  gematcht mit  $\bar{u}_i = u_i = 0$ .  $i_s$  hat deswegen keinen blockierenden Partner nach Lemma 5.4.3. Und DISCARD( $i_s$ ) setzt  $i_s$  auf ungematcht. Im Fall 3.4 hat  $i_s$  mit  $\bar{u}_i = u_i = 0$  keinen blockierenden Partner und wird ungematcht, wenn DISCARD( $i_s$ ) aufgerufen wird. Wegen  $a_{ij} \geq 0$  und  $b_{ij} \geq 0$  und Lemma 5.4.1 wird  $i_s$  keinen blockierenden Partner haben. So wird  $i_s$  nie wieder im Weg  $\mathcal{P}$  auftreten.  $\square$

Wenn eine Tarifkante  $(i, j) \in R^*$  eine Matchingkante wird, wird diese Kante nach Lemma 5.4.3 und der Definition von  $D_i$  nie wieder als eine Nicht-Matchingkante

in einem Suchbaum auftreten. In diesem Sinne können wir sagen, dass die Kante *weggeworfen* ist. Also wird eine Tarifkante höchstens einmal gematcht.

**Proposition 5.4.6.** *Nach jedem Aufruf von PATHUPDATE gilt einer der folgenden Fakten:*

1.  $|\mu|$  ist vergrößert worden;
2. Eine Firma mit Payoff Null ist weggeworfen worden;
3. Eine Kante von  $R^*$  ist weggeworfen worden.

**Satz 5.4.7.** *Mit Algorithmus 4 erhält man ein stabiles Outcome.*

**Beweis:** Wenn das Outcome  $(u, v; \mu)$  nicht stabil ist, d.h., es gibt blockierende Paare, dann kann man ein blockierendes Paar  $(i_1, j_1)$  wählen, wobei  $j_1$  ein blockierender Traumpartner von  $i_1$  ist. Damit ist  $j_1 \in D_{i_1}(u, v, \mu)$ . Nach Lemma 5.4.3 muss  $i_1$  ungematcht sein, da  $(u, v; \mu) \in S$ . Lemma 5.2.4 garantiert die Existenz des gewünschten Wegs  $\mathcal{P} = (i_1, j_1, i_2, j_2, \dots)$ . Das Matching kann höchstens  $|Q|$  mal augmentiert werden. Es gibt nur  $|P|$  Firmen und höchstens  $|P| \times |Q|$  Kanten in  $R^*$  wegzuerwerfen. Somit kann das Verfahren weitergehen und endet nach endlich vielen Schritten mit einem stabilen Outcome.  $\square$

**Satz 5.4.8.** *Der Algorithmus 4 braucht  $\mathcal{O}(n^4)$  Zeit.*

**Beweis:** Nach dem Beweis für Lemma 5.4.7 kann das Verfahren PATHUPDATE  $\mathcal{O}(n^2)$  mal aufgerufen werden. Es ist noch zu zeigen, dass die While-Schleife in Zeile 3 vom Algorithmus 4  $\mathcal{O}(n^2)$  Zeit braucht. Hierzu benutzen wir eine Standardimplementierung der *Ungarischen Methode* (siehe z.B. Galil [4]).

Für ein bestimmtes  $j \in Q$  speichern wir einen Wert  $\delta_j$  entsprechend  $\delta$  in (1)-(3) auf Seite 32 als die „Distanz“ zwischen  $j$  und  $\bar{P} \cup \bar{Q}$ . Für jedes  $j$  muss  $\delta_j$  jedes Mal aktualisiert werden, wenn wir einen Knoten in  $\bar{P} \cup \bar{Q}$  hinzufügen, welches  $\mathcal{O}(n)$  mal passiert. Die Berechnung von  $\delta = \min\{\delta_j > 0\}$  braucht  $\mathcal{O}(n)$  Zeit. Nach einem Update der Payoffs setzen wir  $\delta_j := \max\{0, \delta_j - \delta\}$  und verwenden in  $\mathcal{O}(n^2)$  die BFS-Struktur für den nächsten Aufruf von HUNGARIANUPDATE wieder, bis wir einen ungematchten flexiblen Arbeiter, eine flexible Firma mit Payoff Null oder einen Spieler in  $R$  erhalten.  $\square$

## 5.5 Spezialfälle

Im Falle vom Heiratsproblem tritt immer Fall 3.1 ein, wenn das Outcome nicht stabil ist. In diesem Fall ist der Algorithmus 4 ganz ähnlich dem Algorithmus „men propose-women dispose“. Der Unterschied ist nur, bei „men propose-women dispose“ macht in jedem Schritt jeder nicht verlobte Mann seiner Lieblingsfrau, die ihn nicht abgelehnt hat, einen Antrag, während bei Algorithmus 4 in jedem Schritt nur ein nicht verlobter Mann seiner Lieblingsfrau, die ihn nicht abgelehnt hat, einen Antrag macht. (siehe Beispiel 5.9)

Im Falle vom Zuweisungsspiel treten nur die Fälle 1 und 2 ein. In diesem Falle ist Algorithmus 4 eine spezielle Implementierung der Ungarischen Methode. Er unterscheidet sich nur im Suchbaum von der

Ungarischer Methode. In der Ungarischen Methode ist die Wurzel des Suchbaumes eine ungematchte Firma, während im Algorithmus 4 die Wurzel des Suchbaumes ein blockierender Traumpartner einer ungematchten Firma ist. (siehe Beispiel 5.10)

## 5.6 Beispiele

In diesem Abschnitt führen wir vier Beispiele an, um zu veranschaulichen, wie der Algorithmus 4 funktioniert. Beim ersten Beispiel geht es um die Dummy-Knoten und beim zweiten um den Aufruf von HUNGARIANUPDATE. In den letzten beiden Beispielen verwenden wir Algorithmus 4 gesondert auf die Spezialfälle  $F = \emptyset$  und  $R = \emptyset$ , also auf das Heiratsproblem und das Zuweisungsspiel, an.

Für eine Firma  $i \in P$  bezeichnen wir einen blockierenden Traumpartner  $j$  von  $i$  mit dem Zeichen  $\bar{j}$ . In den Graphen bezeichnen wir mit  $\circ$  die Spieler von  $R$  und mit  $\bullet$  die Spieler von  $F$ .

**Beispiel 5.7.**

$$\begin{aligned} P &= \{p_1, p_2, p_3, p_4\} \\ Q &= \{q_1, q_2, q_3, q_4, q_5\} \\ R &= \{p_1, p_2, q_1, q_2\} \\ F &= \{p_3, p_4, q_3, q_4, q_5\} \end{aligned}$$

Die Gewichtsfunktionen  $a$  und  $b$  werden gegeben durch

	$q_1(R)$	$q_2(R)$	$q_3$	$q_4$	$q_5$
$p_1(R)$	11	8	6	7	8
$p_2(R)$	9	7	7	4	9
$p_3$	10	6	8	5	6
$p_4$	4	6	6	1	3

Tabelle 5.1: Gewichtsfunktion  $a$  für Beispiel 5.7

	$q_1(R)$	$q_2(R)$	$q_3$	$q_4$	$q_5$
$p_1(R)$	7	5	5	6	8
$p_2(R)$	9	5	7	6	7
$p_3$	7	5	5	4	9
$p_4$	4	6	4	5	6

Tabelle 5.2: Gewichtsfunktion  $b$  für Beispiel 5.7

Durch Ergänzung des Dummy-Knoten  $p_5$  haben wir folgende Gewichtsfunktionen.

	$q_1(R)$	$q_2(R)$	$q_3$	$q_4$	$q_5$
$p_1(R)$	11	8	6	7	8
$p_2(R)$	9	7	7	4	9
$p_3$	10	6	8	5	6
$p_4$	4	6	6	1	3
$p_5$	0	0	0	0	0

Tabelle 5.3: Gewichtsfunktion  $a$  mit Dummy-Knoten für Beispiel 5.7

	$q_1(R)$	$q_2(R)$	$q_3$	$q_4$	$q_5$
$p_1(R)$	7	5	5	6	8
$p_2(R)$	9	5	7	6	7
$p_3$	7	5	5	4	9
$p_4$	4	6	4	5	6
$p_5$	0	0	0	0	0

Tabelle 5.4: Gewichtsfunktion  $b$  mit Dummy-Knoten für Beispiel 5.7

Nun berechnen wir Schritt für Schritt ein stabiles Outcome.

- *Erster Schritt:*

$$\begin{array}{ll}
 \mu = \emptyset & D_{p_1} = \{\underline{q_1}\} \\
 u = (0, 0, 0, 0, 0) & D_{p_2} = \{\underline{q_1}, \underline{q_5}\} \\
 v = (0, 0, 0, 0, 0) & D_{p_3} = \{\underline{q_5}\} \\
 \bar{u} = (11, 9, 15, 10, 0) & D_{p_4} = \{\underline{q_3}\} \\
 & D_{p_5} = \{q_3, q_4, q_5\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_1, q_1)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_1 = u_1 = a_{11} = 11$  und  $v_1 = b_{11} = 7$ .

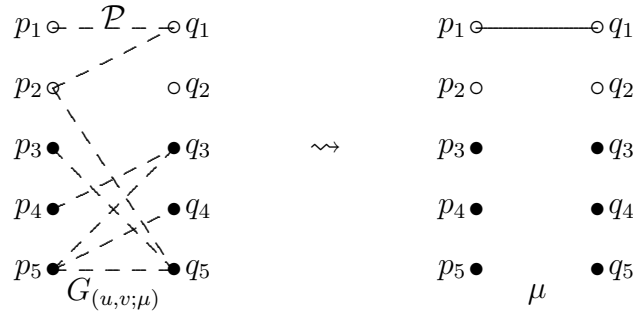


Abbildung 5.1: Erster Schritt

- *Zweiter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 1)\} & D_{p_1} = \emptyset \\
 u = (11, 0, 0, 0, 0) & D_{p_2} = \{\underline{q_1}, \underline{q_5}\} \\
 v = (7, 0, 0, 0, 0) & D_{p_3} = \{\underline{q_5}\} \\
 \bar{u} = (11, 9, 15, 10, 0) & D_{p_4} = \{\underline{q_3}\} \\
 & D_{p_5} = \{q_3, q_4, q_5\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_2, q_1)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_2 = u_2 = a_{21} = 9$ ,  $v_1 = b_{21} = 9$  und  $u_1 = 0$ .

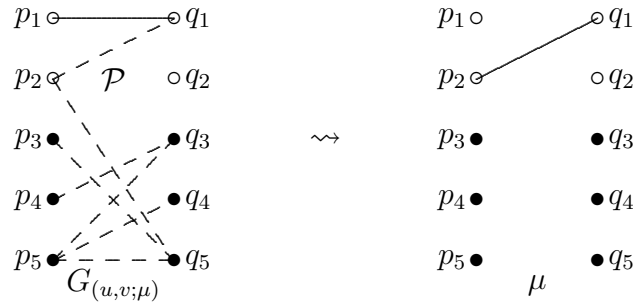


Abbildung 5.2: Zweiter Schritt



- *Dritter Schritt:*

$$\begin{array}{ll}
 \mu = \{(2, 1)\} & D_{p_1} = \{\underline{q_2}, \underline{q_5}\} \\
 u = (0, 9, 0, 0, 0) & D_{p_2} = \{\underline{q_5}\} \\
 v = (9, 0, 0, 0, 0) & D_{p_3} = \{\underline{q_5}\} \\
 \bar{u} = (11, 9, 15, 10, 0) & D_{p_4} = \{\underline{q_3}\} \\
 & D_{p_5} = \{q_3, q_4, q_5\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_1, q_2)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_1 = u_1 = a_{12} = 8$  und  $v_2 = b_{12} = 5$ .

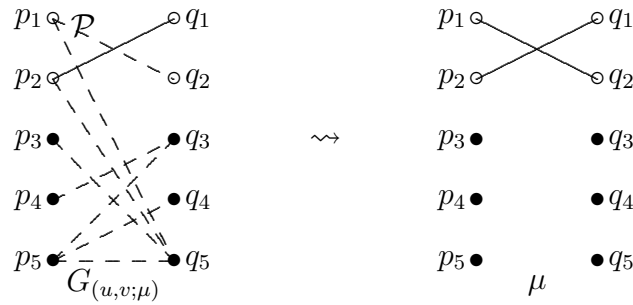


Abbildung 5.3: Dritter Schritt

- *Vierter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 2), (2, 1)\} & D_{p_1} = \{q_5\} \\
 u = (8, 9, 0, 0, 0) & D_{p_2} = \{q_5\} \\
 v = (9, 5, 0, 0, 0) & D_{p_3} = \{\underline{q_5}\} \\
 \bar{u} = (8, 9, 15, 10, 0) & D_{p_4} = \{\underline{q_3}\} \\
 & D_{p_5} = \{q_3, q_4, q_5\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_3, q_5)$ . Fall 1 tritt ein. Setze  $\bar{u}_3 = u_3 = a_{35} + b_{35} - v_5 = 15$ .

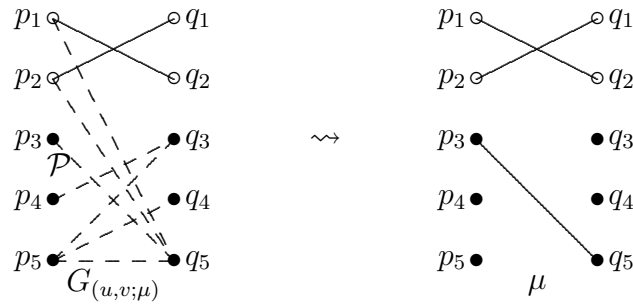


Abbildung 5.4: Vierter Schritt

- *Fünfter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 2), (2, 1), (3, 5)\} & D_{p_1} = \{q_5\} \\
 u = (8, 9, 15, 0, 0) & D_{p_2} = \{q_5\} \\
 v = (9, 5, 0, 0, 0) & D_{p_3} = \emptyset \\
 \bar{u} = (8, 9, 15, 10, 0) & D_{p_4} = \{\underline{q_3}\} \\
 & D_{p_5} = \{q_3, q_4, q_5\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_4, q_3)$ . Fall 1 tritt ein. Setze  $\bar{u}_4 = u_4 = a_{43} + b_{43} - v_3 = 10$ .

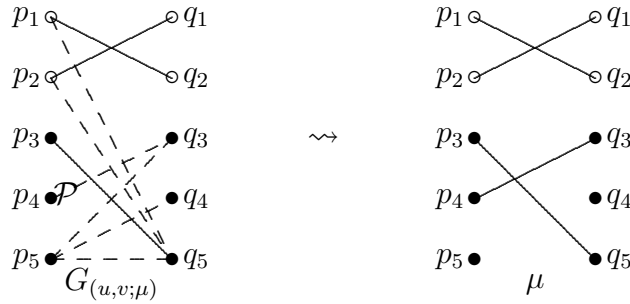


Abbildung 5.5: Fünfter Schritt

- *Output:* Wir erhalten ein stabiles Outcome  $(u, v; \mu)$  mit

$$\begin{array}{l}
 \mu = \{(1, 2), (2, 1), (3, 5), (4, 3)\} \\
 u = (8, 9, 15, 10, 0) \\
 v = (9, 5, 0, 0, 0) \\
 \bar{u} = (8, 9, 15, 10, 0)
 \end{array}$$

### Beispiel 5.8.

$$\begin{array}{l}
 P = \{p_1, p_2, p_3, p_4, p_5\} \\
 Q = \{q_1, q_2, q_3, q_4, q_5\} \\
 R = \{p_1, p_2, q_1, q_2, q_3\} \\
 F = \{p_3, p_4, p_5, q_4, q_5\}
 \end{array}$$

Die Gewichtsfunktionen  $a$  und  $b$  werden gegeben durch

	$q_1(R)$	$q_2(R)$	$q_3(R)$	$q_4$	$q_5$
$p_1(R)$	11	8	6	7	8
$p_2(R)$	9	7	7	4	9
$p_3$	10	6	8	5	6
$p_4$	4	6	6	1	3
$p_5$	6	5	7	2	4

Tabelle 5.5: Gewichtsfunktion  $a$  für Beispiel 5.8

	$q_1(R)$	$q_2(R)$	$q_3(R)$	$q_4$	$q_5$
$p_1(R)$	7	5	5	6	8
$p_2(R)$	9	5	7	6	7
$p_3$	7	5	5	4	9
$p_4$	4	6	4	5	6
$p_5$	7	4	6	6	3

Tabelle 5.6: Gewichtsfunktion  $b$  für Beispiel 5.8

- *Erster Schritt:*

$$\begin{array}{ll}
\mu = \emptyset & D_{p_1} = \{\underline{q_1}\} \\
u = (0, 0, 0, 0, 0) & D_{p_2} = \{\underline{q_1}, \underline{q_5}\} \\
v = (0, 0, 0, 0, 0) & D_{p_3} = \{\underline{q_5}\} \\
\bar{u} = (11, 9, 15, 9, 8) & D_{p_4} = \{\underline{q_5}\} \\
& D_{p_5} = \{\underline{q_4}\}
\end{array}$$

Wähle den Weg  $\mathcal{P} = (p_1, q_1)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_1 = u_1 = a_{11} = 11$  und  $v_1 = b_{11} = 7$ .

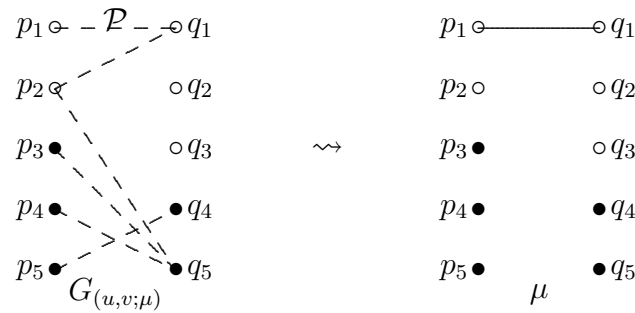


Abbildung 5.6: Erster Schritt

- *Zweiter Schritt:*

$$\begin{array}{ll}
\mu = \{(1, 1)\} & D_{p_1} = \emptyset \\
u = (11, 0, 0, 0, 0) & D_{p_2} = \{\underline{q_1}, \underline{q_5}\} \\
v = (7, 0, 0, 0, 0) & D_{p_3} = \{\underline{q_5}\} \\
\bar{u} = (11, 9, 15, 9, 8) & D_{p_4} = \{\underline{q_5}\} \\
& D_{p_5} = \{\underline{q_4}\}
\end{array}$$

Wähle den Weg  $\mathcal{P} = (p_2, q_5)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_2 = u_2 = a_{25} = 9$  und  $v_5 = b_{25} = 7$ .

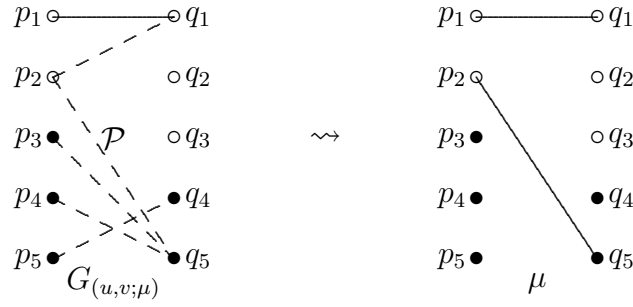


Abbildung 5.7: Zweiter Schritt

- *Dritter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 1), (2, 5)\} & D_{p_1} = \emptyset \\
 u = (11, 9, 0, 0, 0) & D_{p_2} = \{q_1\} \\
 v = (7, 0, 0, 0, 7) & D_{p_3} = \{q_4\} \\
 \bar{u} = (11, 9, 15, 9, 8) & D_{p_4} = \{q_2, q_3, q_4\} \\
 & D_{p_5} = \{q_4, q_5\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_3, q_4)$ . Fall 1 tritt ein. Setze  $\bar{u}_3 = u_3 = a_{34} + b_{34} - v_4 = 9$ .

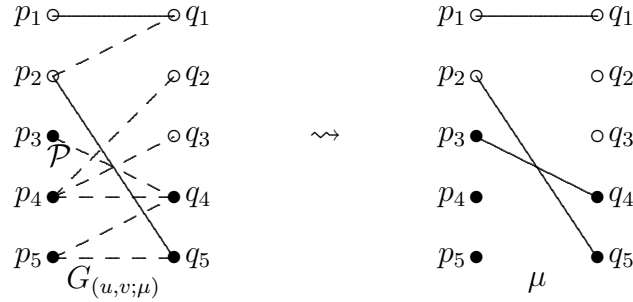


Abbildung 5.8: Dritter Schritt

- *Vierter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 1), (2, 5), (3, 4)\} & D_{p_1} = \emptyset \\
 u = (11, 9, 9, 0, 0) & D_{p_2} = \{q_1\} \\
 v = (7, 0, 0, 0, 7) & D_{p_3} = \emptyset \\
 \bar{u} = (11, 9, 9, 9, 8) & D_{p_4} = \{q_2, q_3, q_4\} \\
 & D_{p_5} = \{q_4, q_5\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_4, q_2)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_4 = u_4 = a_{42} = 6$  und  $v_2 = b_{42} = 6$ .

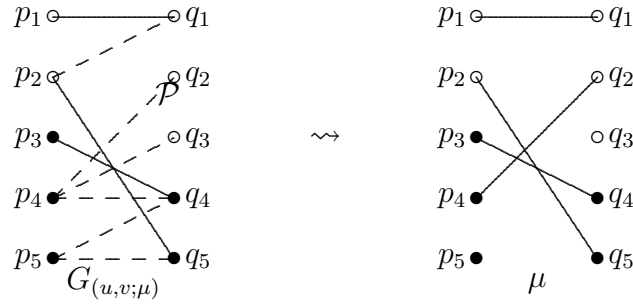


Abbildung 5.9: Vierter Schritt

- *Fünfter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 1), (2, 5), (3, 4), (4, 2)\} & D_{p_1} = \emptyset \\
 u = (11, 9, 9, 6, 0) & D_{p_2} = \{q_1\} \\
 v = (7, 6, 0, 0, 7) & D_{p_3} = \emptyset \\
 \bar{u} = (11, 9, 9, 6, 8) & D_{p_4} = \{q_3, q_4\} \\
 & D_{p_5} = \{\underline{q_4}, q_5\}
 \end{array}$$

Es gibt nur eine ungematchte Firma  $p_5$ , die blockierende Paare erzeugt.  $q_4$  ist der einzige blockierende Traumpartner von  $p_5$ . So müssen  $i_1 = p_5$  und  $j_1 = q_4$  sein. Kein gewünschter Weg  $\mathcal{P}$  existiert. Deswegen wird das Verfahren  $\text{HUNGARIANUPDATE}(q_4)$  aufgerufen.  $\bar{P} = \{p_3\}$ ,  $\bar{Q} = \{q_4\}$  und  $\delta = 1$ . Ein neuer blockierender Traumpartner  $q_3$  von  $p_5$  entsteht.

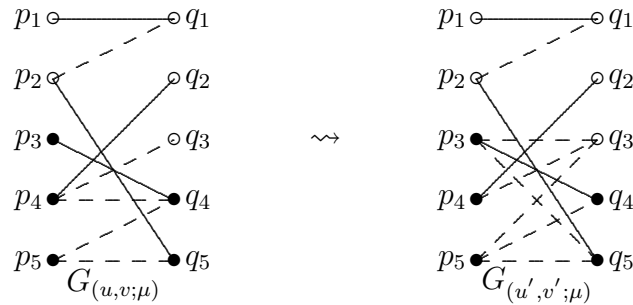


Abbildung 5.10: Fünfter Schritt

- *Sechster Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 1), (2, 5), (3, 4), (4, 2)\} & D_{p_1} = \emptyset \\
 u = (11, 9, 8, 6, 0) & D_{p_2} = \{q_1\} \\
 v = (7, 6, 0, 1, 7) & D_{p_3} = \{q_3, q_5\} \\
 \bar{u} = (11, 9, 8, 6, 8) & D_{p_4} = \{q_3\} \\
 & D_{p_5} = \{\underline{q_3}, \underline{q_4}, q_5\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_5, q_3)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_5 = u_5 = a_{53} = 7$  und  $v_3 = b_{53} = 6$ .

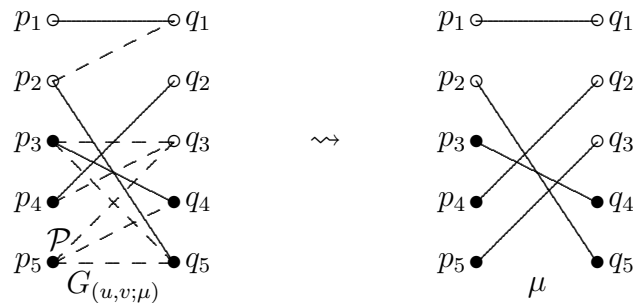


Abbildung 5.11: Sechster Schritt

- *Output:* Wir erhalten ein stabiles Outcome  $(u, v; \mu)$  mit

$$\begin{aligned} \mu &= \{(1, 1), (2, 5), (3, 4), (4, 2), (5, 3)\} \\ u &= (11, 9, 8, 6, 7) \\ v &= (7, 6, 6, 1, 7) \\ \bar{u} &= (11, 9, 8, 6, 7) \end{aligned}$$

Wir betrachten Beispiel 3.5. Es kann z.B. wie folgt formuliert werden.

**Beispiel 5.9.**

$$\begin{aligned} P &= \{p_1, p_2, p_3, p_4, p_5\} \\ Q &= \{q_1, q_2, q_3, q_4\} \\ R &= P \cup Q \end{aligned}$$

Die Gewichtsfunktionen  $a$  und  $b$  werden gegeben durch

	$q_1(R)$	$q_2(R)$	$q_3(R)$	$q_4(R)$
$p_1(R)$	4	5	2	3
$p_2(R)$	4	2	5	3
$p_3(R)$	2	3	5	4
$p_4(R)$	2	5	4	3
$p_5(R)$	4	5	3	0

Tabelle 5.7: Gewichtsfunktion  $a$  für Beispiel 5.9

	$q_1(R)$	$q_2(R)$	$q_3(R)$	$q_4$
$p_1(R)$	4	3	5	3
$p_2(R)$	3	5	2	2
$p_3$	5	4	1	1
$p_4$	2	2	4	4
$p_5$	1	1	3	5

Tabelle 5.8: Gewichtsfunktion  $b$  für Beispiel 5.9

Fügen wir Dummy-Knoten  $q_5$  hinzu, dann haben wir folgendes:

$$\begin{aligned}
P &= \{p_1, p_2, p_3, p_4, p_5\} \\
Q &= \{q_1, q_2, q_3, q_4, q_5\} \\
R &= \{p_1, p_2, p_3, p_4, p_5, q_1, q_2, q_3, q_4\} \\
F &= \{q_5\}
\end{aligned}$$

	$q_1(R)$	$q_2(R)$	$q_3(R)$	$q_4(R)$	$q_5$
$p_1(R)$	4	5	2	3	0
$p_2(R)$	4	2	5	3	0
$p_3(R)$	2	3	5	4	0
$p_4(R)$	2	5	4	3	0
$p_5(R)$	4	5	3	0	0

Tabelle 5.9: Gewichtsfunktion  $a$  mit Dummy-Knoten für Beispiel 5.9

	$q_1(R)$	$q_2(R)$	$q_3(R)$	$q_4(R)$	$q_5$
$p_1(R)$	4	3	5	3	0
$p_2(R)$	3	5	2	2	0
$p_3(R)$	5	4	1	1	0
$p_4(R)$	2	2	4	4	0
$p_5(R)$	1	1	3	5	0

Tabelle 5.10: Gewichtsfunktion  $b$  mit Dummy-Knoten für Beispiel 5.9

- *Erster Schritt:*

$$\begin{aligned}
\mu &= \emptyset & D_{p_1} &= \{\underline{q_2}\} \\
u &= (0, 0, 0, 0, 0) & D_{p_2} &= \{\underline{q_3}\} \\
v &= (0, 0, 0, 0, 0) & D_{p_3} &= \{\underline{q_3}\} \\
\bar{u} &= (5, 5, 5, 5, 5) & D_{p_4} &= \{\underline{q_2}\} \\
& & D_{p_5} &= \{\underline{q_2}\}
\end{aligned}$$

Wähle den Weg  $\mathcal{P} = (p_1, q_2)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_1 = u_1 = a_{12} = 5$  und  $v_1 = b_{12} = 3$ .

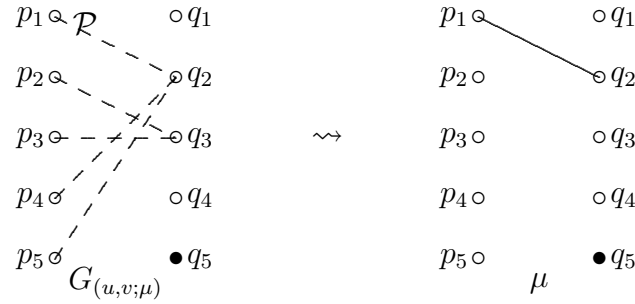


Abbildung 5.12: Erster Schritt

- *Zweiter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 2)\} & D_{p_1} = \emptyset \\
 u = (5, 0, 0, 0, 0) & D_{p_2} = \{\underline{q_3}\} \\
 v = (0, 3, 0, 0, 0) & D_{p_3} = \{\underline{q_3}\} \\
 \bar{u} = (5, 5, 5, 5, 5) & D_{p_4} = \{\underline{q_3}\} \\
 & D_{p_5} = \{\underline{q_1}\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_2, q_3)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_2 = u_2 = a_{23} = 5$  und  $v_3 = b_{23} = 2$ .

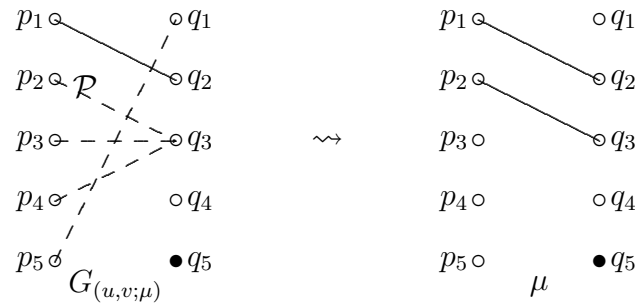


Abbildung 5.13: Zweiter Schritt

- *Dritter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 2), (2, 3)\} & D_{p_1} = \emptyset \\
 u = (5, 5, 0, 0, 0) & D_{p_2} = \emptyset \\
 v = (0, 3, 2, 0, 0) & D_{p_3} = \{\underline{q_4}\} \\
 \bar{u} = (5, 5, 5, 5, 5) & D_{p_4} = \{\underline{q_3}\} \\
 & D_{p_5} = \{\underline{q_1}\}
 \end{array}$$



Wähle den Weg  $\mathcal{P} = (p_3, q_4)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_3 = u_3 = a_{34} = 4$  und  $v_4 = b_{34} = 1$ .

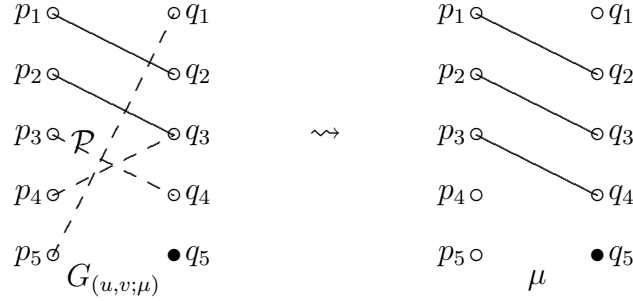


Abbildung 5.14: Dritter Schritt

- *Vierter Schritt:*

$$\begin{aligned}
 \mu &= \{(1, 2), (2, 3), (3, 4)\} & D_{p_1} &= \emptyset \\
 u &= (5, 5, 4, 0, 0) & D_{p_2} &= \emptyset \\
 v &= (0, 3, 2, 1, 0) & D_{p_3} &= \emptyset \\
 \bar{u} &= (5, 5, 4, 5, 5) & D_{p_4} &= \{q_3\} \\
 & & D_{p_5} &= \{q_1\}
 \end{aligned}$$

Wähle den Weg  $\mathcal{P} = (p_4, q_3)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_3 = u_3 = a_{34} = 4$ ,  $v_4 = b_{34} = 1$  und  $u_2 = 0$ .

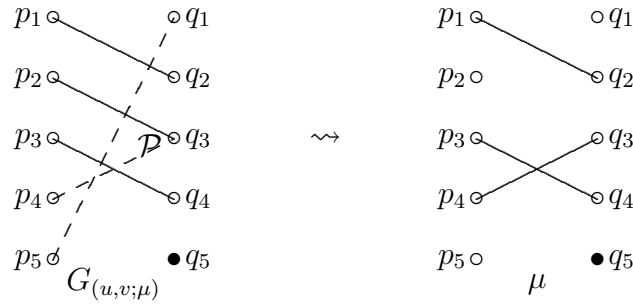


Abbildung 5.15: Vierter Schritt

- *Fünfter Schritt:*

$$\begin{aligned}
 \mu &= \{(1, 2), (3, 4), (4, 3)\} & D_{p_1} &= \emptyset \\
 u &= (5, 0, 4, 4, 0) & D_{p_2} &= \{q_1\} \\
 v &= (0, 3, 4, 1, 0) & D_{p_3} &= \emptyset \\
 \bar{u} &= (5, 5, 4, 4, 5) & D_{p_4} &= \emptyset \\
 & & D_{p_5} &= \{q_1\}
 \end{aligned}$$

Wähle den Weg  $\mathcal{P} = (p_2, q_1)$ . Fall 3.1 tritt ein. Setze  $\bar{u}_2 = u_2 = a_{21} = 4$  und  $v_1 = b_{21} = 3$ .

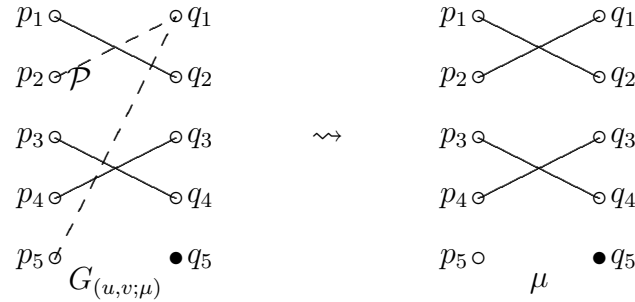


Abbildung 5.16: Fünfter Schritt

- *Output:* Wir erhalten ein stabiles Outcome  $(u, v; \mu)$  mit

$$\begin{aligned} \mu &= \{(1, 2), (2, 1), (3, 4), (4, 3)\} \\ u &= (5, 4, 4, 4, 0) \\ v &= (3, 3, 4, 1, 0) \\ \bar{u} &= (5, 4, 4, 4, 0) \end{aligned}$$

Betrachten wir wieder Beispiel 4.5. Es kann z.B. wie folgt formuliert werden.

**Beispiel 5.10.**

$$\begin{aligned} P &= \{p_1, p_2, p_3, p_4, p_5\} \\ Q &= \{q_1, q_2, q_3, q_4, q_5\} \\ F &= P \cup Q \\ R &= \emptyset \end{aligned}$$

Die Gewichtsfunktionen  $a = b$  werden gegeben durch

	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$
$p_1$	1.5	4	4.5	0.5	3
$p_2$	0.5	2	0.5	2.5	2.5
$p_3$	3.5	1	3.5	4.5	1
$p_4$	1.5	0.5	3	4	4
$p_5$	1	3	1.5	3	1

Tabelle 5.11: Gewichtsfunktion  $a$  und  $b$  für Beispiel 5.10

- *Erster Schritt:*

$$\begin{array}{ll}
 \mu = \emptyset & D_{p_1} = \{\underline{q_3}\} \\
 u = (0, 0, 0, 0, 0) & D_{p_2} = \{\underline{q_4}, \underline{q_5}\} \\
 v = (0, 0, 0, 0, 0) & D_{p_3} = \{\underline{q_4}\} \\
 \bar{u} = (9, 5, 9, 8, 6) & D_{p_4} = \{\underline{q_4}, \underline{q_5}\} \\
 & D_{p_5} = \{\underline{q_2}, \underline{q_4}\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_1, q_3)$ . Fall 2 tritt ein. Setze  $\bar{u}_1 = u_1 = a_{13} + b_{13} - v_3 = 9$ .

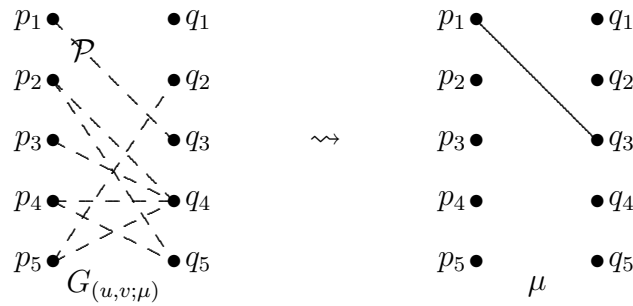


Abbildung 5.17: Erster Schritt

- *Zweiter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 3)\} & D_{p_1} = \emptyset \\
 u = (9, 0, 0, 0, 0) & D_{p_2} = \{\underline{q_4}, \underline{q_5}\} \\
 v = (0, 0, 0, 0, 0) & D_{p_3} = \{\underline{q_4}\} \\
 \bar{u} = (9, 5, 9, 8, 6) & D_{p_4} = \{\underline{q_4}, \underline{q_5}\} \\
 & D_{p_5} = \{\underline{q_2}, \underline{q_4}\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_2, q_4)$ . Fall 2 tritt ein. Setze  $\bar{u}_2 = u_2 = a_{24} + b_{24} - v_4 = 5$ .

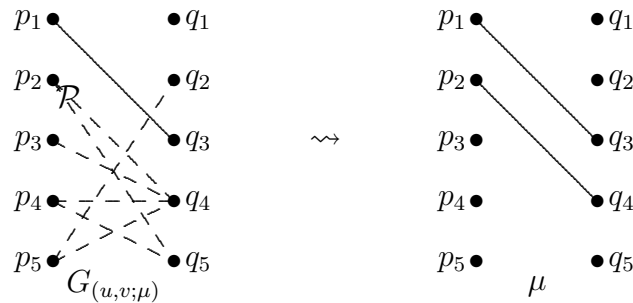


Abbildung 5.18: Zweiter Schritt

- *Dritter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 3), (2, 4)\} & D_{p_1} = \emptyset \\
 u = (9, 5, 0, 0, 0) & D_{p_2} = \{q_5\} \\
 v = (0, 0, 0, 0, 0) & D_{p_3} = \{\underline{q_4}\} \\
 \bar{u} = (9, 5, 9, 8, 6) & D_{p_4} = \{q_4, \underline{q_5}\} \\
 & D_{p_5} = \{\underline{q_2}, \underline{q_4}\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_3, q_4, p_2, q_5)$ . Fall 2 tritt ein. Setze  $\bar{u}_3 = u_3 = a_{34} + b_{34} - v_4 = 9$ .

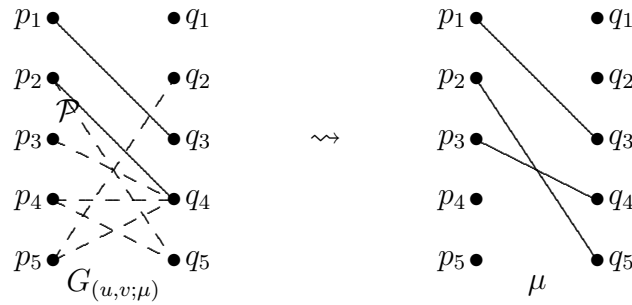


Abbildung 5.19: Dritter Schritt

- *Vierter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 3), (2, 5), (3, 4)\} & D_{p_1} = \emptyset \\
 u = (9, 5, 9, 0, 0) & D_{p_2} = \{q_4\} \\
 v = (0, 0, 0, 0, 0) & D_{p_3} = \emptyset \\
 \bar{u} = (9, 5, 9, 8, 6) & D_{p_4} = \{q_4, \underline{q_5}\} \\
 & D_{p_5} = \{\underline{q_2}, \underline{q_4}\}
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_5, q_2)$ . Fall 2 tritt ein. Setze  $\bar{u}_5 = u_5 = a_{52} + b_{52} - v_2 = 6$ .

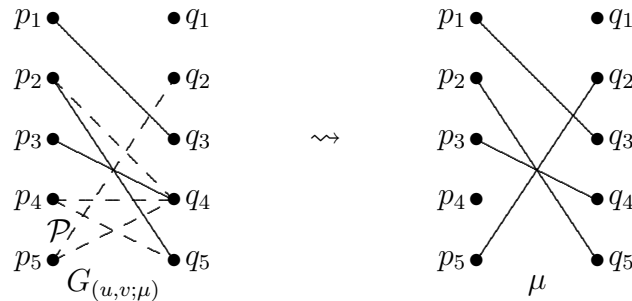


Abbildung 5.20: Vierter Schritt

- *Fünfter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 3), (2, 5), (3, 4), (5, 2)\} & D_{p_1} = \emptyset \\
 u = (9, 5, 9, 0, 6) & D_{p_2} = \{q_4\} \\
 v = (0, 0, 0, 0, 0) & D_{p_3} = \emptyset \\
 \bar{u} = (9, 5, 9, 8, 6) & D_{p_4} = \{\underline{q_4}, \underline{q_5}\} \\
 & D_{p_5} = \{q_4\}
 \end{array}$$

Es gibt nur eine ungematchte Firma  $p_4$ , die blockierende Paare erzeugt. Für weder  $j_1 = q_4$  noch  $j_1 = q_5$  existiert ein gewünschter Weg  $\mathcal{P}$ . Rufe das Verfahren  $\text{HUNGARIANUPDATE}(q_4)$  auf.  $\bar{P} = \{p_3\}$ ,  $\bar{Q} = \{q_4\}$  und  $\delta = 2$ . Dann erhalten wir die neuen Kanten  $(p_3, q_1)$  und  $(p_3, q_3)$ .

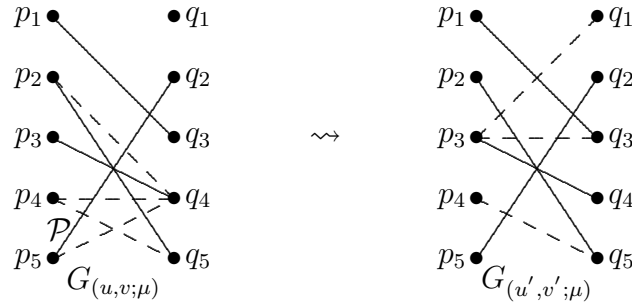


Abbildung 5.21: Fünfter Schritt

- *Sechster Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 3), (2, 5), (3, 4), (5, 2)\} & D_{p_1} = \emptyset \\
 u = (9, 5, 7, 0, 6) & D_{p_2} = \emptyset \\
 v = (0, 0, 0, 2, 0) & D_{p_3} = \{q_1, q_3\} \\
 \bar{u} = (9, 5, 7, 8, 6) & D_{p_4} = \{\underline{q_5}\} \\
 & D_{p_5} = \emptyset
 \end{array}$$

Es gibt nur eine ungematchte Firma  $p_4$ , die blockierende Paare erzeugt.  $j_1 = q_5$  ist der einzige blockierende Traumpartner. Kein gewünschter Weg  $\mathcal{P}$  existiert. Rufe das Verfahren  $\text{HUNGARIANUPDATE}(q_5)$  auf.  $\bar{P} = \{p_2\}$ ,  $\bar{Q} = \{q_5\}$  und  $\delta = 1$ . Somit erhalten wir die neue Kante  $(p_2, q_2)$ .

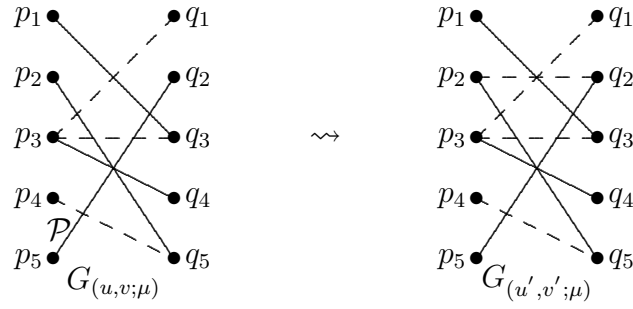


Abbildung 5.22: Sechster Schritt

- *Siebter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 3), (2, 5), (3, 4), (5, 2)\} & D_{p_1} = \emptyset \\
 u = (9, 4, 7, 0, 6) & D_{p_2} = \{q_2\} \\
 v = (0, 0, 0, 2, 1) & D_{p_3} = \{q_1, q_3\} \\
 \bar{u} = (9, 4, 7, 8, 6) & D_{p_4} = \{q_5\} \\
 & D_{p_5} = \emptyset
 \end{array}$$

Es gibt nur eine ungematchte Firma  $p_4$ , die blockierende Paare erzeugt.  $j_1 = q_5$  ist der einzige blockierende Traumpartner. Kein gewünschter Weg  $\mathcal{P}$  existiert. Rufe das Verfahren HUNGARIANUPDATE( $q_5$ ) auf.  $\bar{P} = \{p_2, p_5\}$ ,  $\bar{Q} = \{q_2, q_5\}$  und  $\delta = 1$ . Danach bekommen wir einen neuen blockierenden Traumpartner  $q_4$  von  $p_4$ .

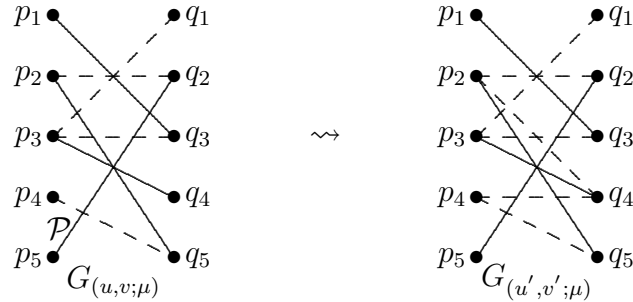


Abbildung 5.23: Siebter Schritt

- *Achter Schritt:*

$$\begin{array}{ll}
 \mu = \{(1, 3), (2, 5), (3, 4), (5, 2)\} & D_{p_1} = \emptyset \\
 u = (9, 3, 7, 0, 5) & D_{p_2} = \{q_2, q_4\} \\
 v = (0, 1, 0, 2, 2) & D_{p_3} = \{q_1, q_3\} \\
 \bar{u} = (9, 3, 7, 6, 5) & D_{p_4} = \{q_4, q_5\} \\
 & D_{p_5} = \emptyset
 \end{array}$$

Wähle den Weg  $\mathcal{P} = (p_4, q_4, p_3, q_1)$ . Fall 2 tritt ein. Setze  $\bar{u}_4 = u_4 = a_{44} + b_{44} - v_4 = 6$ .

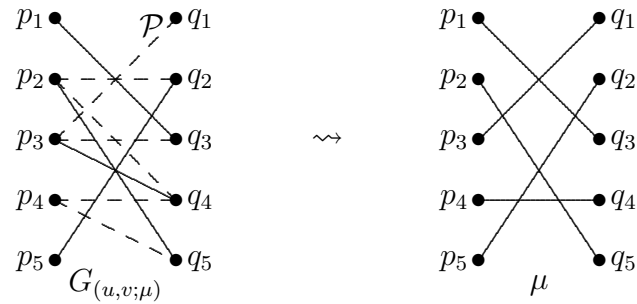


Abbildung 5.24: Achter Schritt

• *Output:*

$$\mu = \{(1, 3), (2, 5)(3, 1), (4, 4), (5, 2)\}$$

$$u = (9, 3, 7, 6, 5)$$

$$v = (0, 1, 0, 2, 2)$$

$$\bar{u} = (9, 3, 7, 6, 5)$$

## .1 Notation

---

$G = (V, E)$	Graph mit Knotenmenge $V$ und Kantenmenge $E$
$G = (P \dot{\cup} Q, E)$	bipartiter Graph mit Knotenpartition $P$ und $Q$ und Kantenmenge $E$
$\mu$	Matching
$K_{n,n}$	$n$ -vollständiger bipartiter Graph
$F$	Menge aller flexiblen Knoten
$R$	Menge aller tariflichen Knoten
$(u, v)$	Payoff
$(u, v; \mu)$	Outcome
$Y_i$	Menge aller blockierenden Partner von $i$
$\nu$	charakteristische Funktion eines kooperativen Spiels

---



# Literaturverzeichnis

- [1] András Frank. On Kuhn's Hungarian Method - A tribute from Hungary. Technical report, Egerváry Research Group on Combinatorial Optimization, October 2004.
- [2] Saturo Fujishige und Akihisa Tamura. A two-sided discrete-concave market with bounded side payments: An approach by discrete convex analysis. RIMS Preprint No. 1470, Kyoto University, August 2004.
- [3] David Gale und Lloyd S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9-15, 1962.
- [4] Zvi Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys*, 18(1):23-38, 1986.
- [5] Kimmo Eriksson und Johan Karlander. Stable Matching in a common generalization of the marriage and assignment models. *Discrete Mathematics*, 217(1-3):135-156, 1997.
- [6] Winfried Hochstättler und Robert Nickel. Note on an auction procedure for matching games in polynomial time. Technical report, FernUniversität in Hagen, Germany, 2005.
- [7] Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83-97, 1955.
- [8] Marida Sotomayor. Existence of stable outcomes and the lattice property for a unified matching market. *Mathematical Social Sciences*, 39:119-132, 2000.
- [10] Alvin E. Roth und Marida Sotomayor. Stable outcomes in discrete and continuous models of two-sided matching: A unified treatment. *Revista de Econometria, The Brazilian Review of Econometrics*, 16(2), November 1996.

- [10] Alvin E. Roth und Marida Sotomayor. *two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge University Press, Cambridge, 1991.
- [11] Lloyd S. Shapley und Martin Shubik. The assignment game I: The core. *International Journal of Game Theory*, 1:111-130, 2000.
- [12] A. Roth, U.G. Rothblum, J.H. Vande Vate. Stable matching, optimal assignments and linear programming. *Mathematics of Operations Research* 18:803-828, 1993
- [13] A.E. Roth. The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory. *J. Political Economy* 92:991-1016, 1984
- [14] A.E. Roth. A Natural Experiment in the Organization of Entry Level Labor Markets: Regional Markets for New Physicians and Surgeons in the U.K. *American Economic Review* 81:415-440, 1991
- [15] Dieter Jungnickel. *Graphs, Networks and Algorithms*. 2005
- [18] Winfried Hochstättler, Hui Jin und Robert Nickel. The Hungarian Method in a Mixed Matching Market. 2005
- [17] Martin Schönhacker. <http://www.ads.tuwien.ac.at/teaching/ss05/ak5/Heiratsproblem.pdf>
- [18] Winfried Hochstättler. Skript zur Vorlesung Linearen Programmierung WS 2002/2003. <http://www.math.tu-cottbus.de/INSTITUT/lsgdi/LP/Materialien/kapitel2.pdf>
- [19] Winfried Hochstättler. Skript zur Vorlesung Graphentheorie SS 2005. <http://www.math.tu-cottbus.de/INSTITUT/lsgdi/Graph2004/kapitel2.pdf>
- [20] <http://members.aol.com/mnsjava/referate/graphen/breitensuche.htm>
- [21] John von Neumann und Oskar Morgenstern. *Theory of games and economic behavior*. 1972