



Kapitel 5

Graphentheorie

Zusammenfassung In diesem Kapitel werden fundamentale Konzepte und Verfahren der Graphentheorie vorgestellt. Dabei legen wir den Fokus auf populäre Probleme in diesem Bereich, die zum einen unmittelbare Anwendungen haben und zum anderen auch häufig als Teil- oder Subprobleme in übergeordneten Lösungsverfahren zu finden sind.

Lernziele Kapitel 5

Sie kennen nach Lesen und Verinnerlichen dieses Kapitels

- Verfahren zur Lösung spezieller Probleme im Bereich der Graphentheorie.

Sie können nach Lesen und Verinnerlichen dieses Kapitels

- diese Verfahren anwenden und
- die entsprechenden Probleme lösen.

In diesem Kapitel wollen wir uns wiederum mit der Lösung von bestimmten Problemen beschäftigen. Anders als in den Kapiteln 3 und 4 wird es dabei um einzelne Probleme und nicht um ganze Klassen von Problemen gehen. Jedes der behandelten Probleme lässt sich in einer der Klassen, die in den Kapiteln 3 und 4 diskutiert wurden, verorten und somit auch mit den entsprechenden Methoden lösen. Wir werden uns daher in den Abschnitten 5.1 bis 5.4 auf Verfahren konzentrieren, die speziell auf die jeweiligen Probleme abgestimmt und somit in der Regel deutlich effizienter oder intuitiver sind. Dabei geht es nicht unbedingt darum, das nach Stand der Dinge effizienteste Verfahren zu diskutieren. Diese sind zum Teil über Jahrzehnte weiterentwickelt worden und häufig hochkomplex. Vielmehr wollen wir uns auf die grundlegenden (zum Teil intuitiv motivierbaren) Ideen solcher Verfahren konzentrieren. Die Auswahl der Probleme, die wir dabei betrachten, deckt zentrale, in der Graphentheorie behandelte Probleme ab. Diese Probleme sind nicht nur für

sich genommen motivierbar, sondern treten auch oft als Teil- oder Subproblem in übergeordneten Problemen auf.

Für einen weiteren Überblick über Probleme, Verfahren und strukturelle Erkenntnisse im Bereich der Graphentheorie seien dem Leser Diestel (2017), Jungnickel (2013) und Wallis (2007) empfohlen.

5.1 Kürzeste Wege (und Pfade)

In diesem Abschnitt geht es um die Bestimmung von Wegen und Pfaden in einem Graph. Was Wege und Pfade sind, haben wir bereits in den Definitionen 2.5 und 2.6 spezifiziert. Auch Längen von Wegen haben wir schon in Definition 4.7 definiert. Wir wollen uns diese Definitionen hier aber noch einmal vor Augen führen.

Definition 2.5 (Weg). Ein Weg von einem Knoten $u \in V$ zu einem Knoten $v \in V$ in einem gerichteten Graphen $G = (V, E)$ ist eine Sequenz von n , $n \geq 1$, Kanten $((u_1, v_1), \dots, (u_n, v_n))$, sodass

- $u_1 = u$ und $v_n = v$, d. h., die erste Kante beginnt bei Knoten u und die letzte endet bei Knoten v ,
- $(u_k, v_k) \in E$ für jedes $k = 1, \dots, n$, d. h., die Kanten im Weg sind im Graphen vorhanden, und
- $u_k = v_{k-1}$ für jedes $k = 2, \dots, n$, d. h., jede Kante beginnt bei dem Knoten, bei dem die vorherige endet.

Definition 2.6 (Pfad). Ein Pfad von einem Knoten $u \in V$ zu einem Knoten $v \in V$ in einem gerichteten Graphen $G = (V, E)$ ist ein Weg $((u_1, v_1), \dots, (u_n, v_n))$, sodass

- $u_k \neq u_{k'}$ für jedes $1 \leq k < k' \leq n$,
- $v_k \neq v_{k'}$ für jedes $1 \leq k < k' \leq n$ und
- $u_1 \neq v_n$,

d. h., jeder Knoten in V wird höchstens einmal erreicht.

Definition 4.7 (Länge eines Weges). Die Länge eines Weges ist die Summe der Gewichte (Längen) der im Weg enthaltenen Kanten.

Abb. 5.1 stellt einen Graph dar, in dem es fünf verschiedene Wege von Knoten 1 zu Knoten 8 gibt, die auch Pfade sind. Der Pfad $(1, 2, 5, 4, 7, 8)$ hat eine Länge von $5 + 9 + 3 + 4 + 7 = 28$. Die weiteren Pfade $(1, 2, 3, 6, 9, 8)$, $(1, 2, 5, 8)$, $(1, 2, 5, 6, 9, 8)$ und $(1, 4, 7, 8)$ haben die Längen 11, 22, 18 und 13. An diesem Beispiel erkennen wir schon, dass ein Weg bzw. Pfad kürzer sein kann, obwohl er mehr Kanten enthält. Da der Graph auch einen Kreis, nämlich $(6, 9, 10)$, enthält, gibt es auch Wege, die keine Pfade sind. Weg $(1, 2, 3, 6, 9, 10, 6, 9, 8)$ von Knoten 1 zu Knoten 8 ist kein Pfad, da Knoten 6 und 9 zweifach besucht werden, und hat Länge 18.

Wir hatten uns schon in Abschnitt 4.4 mit der Bestimmung kürzester Wege beschäftigt, uns dort aber auf den recht speziellen Fall eines gerichteten, kreisfreien Graphen beschränkt. Da es in kreisfreien Graphen keine Kreise gibt, mussten

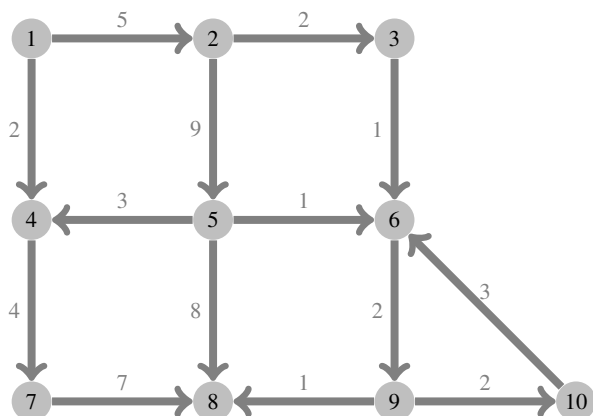


Abb. 5.1 Beispiel zu Wegen, Pfaden und deren Längen

wir nicht zwischen Wegen und Pfaden unterscheiden, da in diesem Fall jeder Weg ein Pfad ist. Diesen Fall werden wir hier ausklammern und daher auch das betrachtete Problem allgemeiner fassen, indem wir uns nicht auf kreisfreie Graphen beschränken.

Problem 5.1 (Kürzester Weg in einem gerichteten Graphen).

Eingabe: Gerichteter, gewichteter Graph $G = (V, E, w)$ und zwei ausgewiesene Knoten i und j

LR: Wege von i nach j

ZB: –

ZS: Länge des Weges ist zu minimieren.

Frage: Was ist eine optimale Lösung?

Wir werden uns in den Abschnitten 5.1.1 und 5.1.2 mit zwei verschiedenen Verfahren beschäftigen. Das Verfahren in Abschnitt 5.1.1 führt nur in dem speziellen Fall, in dem es keine negativen Kantengewichte gibt, zu optimalen Lösungen. Dies ist in vielen naheliegenden Anwendungen, wie z. B. bei der Abbildung eines Streckennetzes, bei der die Kantengewichte die Distanzen darstellen, der Fall und daher hat dieses Verfahren eine hohe Bedeutung. Wenn wir nicht ausschließen können, dass es negative Kantengewichte gibt, dann benötigen wir ein aufwendigeres Verfahren, das wir in Abschnitt 5.1.2 besprechen.

5.1.1 Nicht-negative Kantenlängen

Das Verfahren, das hier vorgestellt werden soll, geht auf Dijkstra (1959) zurück. Dieses Verfahren löst Problem 5.1 (Kürzester Weg in einem gerichteten Graphen), indem der kürzeste Weg von Knoten i zu jedem anderen Knoten ermittelt wird.

Wir wollen davon ausgehen, dass der Ausgangsknoten i Knoten 1 ist. Da Kreise existieren können, kann es auch Wege geben, die keine Pfade sind. Trotzdem werden wir uns um Wege, die keine Pfade sind, nicht explizit kümmern müssen, da kein Weg von 1 zu einem anderen Knoten kürzer sein kann, als der kürzeste Pfad. Dies liegt daran, dass ein Weg, der kein Pfad ist, einen Kreis enthält und dieser eine nicht-negative Länge hat, da kein Kantengewicht negativ ist. Also können wir alle Kreise aus einem Weg entfernen und erhalten einen Pfad, der nicht länger als der Weg ist.

Zusammenfassung: Kürzeste Wege und kürzeste Pfade

Da Kreise nicht-negative Längen haben, wenn Kantengewichte nicht-negativ sind, gibt es keinen Weg der kürzer als der kürzeste Pfad ist.

Um zunächst eine Intuition für die Vorgehensweise zu schaffen, wollen wir noch einmal rekapitulieren, wie wir in Abschnitt 4.4 mittels eines Ansatzes der DP Problem 4.2 (Kürzester Pfad in einem gerichteten kreisfreien Graphen), also den Spezialfall von Problem 5.1 (Kürzester Pfad in einem gerichteten Graphen), in dem der vorliegende Graph keine Kreise enthält, gelöst haben. Auch dort wurden die kürzesten Wege von Knoten 1 zu jedem anderen Knoten ermittelt. Wir waren dort, aufgrund der Kreisfreiheit, in der Lage, die Knoten vorab so zu sortieren, dass wir die kürzesten Wege zu den einzelnen Knoten in dieser Reihenfolge bestimmen konnten. Wir wussten also beispielsweise, dass es von Knoten 3 keinen Weg zu Knoten 2 gibt und wir daher den kürzesten Weg von 1 nach 2 bestimmen können, bevor wir den kürzesten Weg von 1 nach 3 bestimmen. Diese Möglichkeit haben wir bei Problem 5.1 (Kürzester Pfad in einem gerichteten Graphen) nicht, denn potentiell gibt es von jedem Knoten zu jedem anderen Knoten einen Weg.

Überblick über das Verfahren

Wir werden uns daher in jedem Schritt überlegen, zu welchen Knoten wir die kürzesten Wege inzwischen kennen, und dann Wege zu anderen Knoten unter Verwendung dieser Kenntnis entwickeln. Sollte zu einem Knoten der kürzeste Weg bereits bekannt sein, werden weitere Wege zu ihm ignoriert und dadurch Wege, die keine Pfade sind, bei der Konstruktion implizit ignoriert. Wir werden im Rahmen des Verfahrens drei Informationen zu jedem Knoten vorhalten und bei Bedarf aktualisieren.

- Wir merken uns für jeden Knoten k die Länge L_k des kürzesten bisher gefundenen Weges von 1 zu Knoten k . Immer wenn wir einen kürzeren Weg zu k finden, aktualisieren wir diese Information. Für Knoten 1 initialisieren wir diesen Wert mit 0, da wir einen Weg von Knoten 1 zu Knoten 1 der Länge null kennen und es bei nicht-negativen Kantengewichten keinen kürzeren Weg geben kann. Für jeden anderen Knoten k , zu dem wir vorab keinen Weg kennen, initialisieren wir diesen Wert mit ∞ , was dazu führt, dass der erste gefundene Weg zu k auf jeden Fall „kürzer“ ist.

- Wir merken uns für jeden Knoten k den vorletzten Knoten auf dem kürzesten bisher gefundenen Weg von 1 zu k . Wir sprechen von diesem Knoten auch als Vorgänger V_k von k . Immer wenn wir einen kürzeren Weg zu k finden, aktualisieren wir den Vorgänger von k . Wir initialisieren diese Information für alle Knoten mit „–“, denn für Knoten 1 kennen wir schon einen kürzesten Weg, aber Knoten 1 hat auf diesem Weg keinen Vorgänger, und zu jedem anderen Knoten kennen wir noch keinen Weg.
- Wir merken uns für jeden Knoten k , ob wir ihn im weiteren Verlauf des Verfahrens noch berücksichtigen müssen. Dafür halten wir die Menge V' von Knoten vor, die k enthält, falls k noch berücksichtigt werden muss, weil der kürzeste Weg zu k noch nicht sicher bekannt ist. Initial enthält V' alle Knoten.

Nach der Initialisierung bearbeitet das Verfahren in jedem Schritt einen Knoten abschließend. Hierfür wird jeweils der Knoten k gewählt, zu dem es unter allen Knoten, die noch berücksichtigt werden müssen, den kürzesten bekannten Weg von Knoten 1 gibt. Wir prüfen dann für jede von k ausgehende Kante (k, k') zu einem noch nicht abschließend bearbeiteten Knoten k' , ob der kürzeste bekannte Weg von 1 zu k ergänzt um die Kante (k, k') einen neuen kürzesten bekannten Weg von 1 zu k' darstellt. Sollte dies der Fall sein, ergibt sich ein neuer Wert für $L_{k'}$, indem wir L_k um die Länge $w_{k,k'}$ der Kante (k, k') vergrößern, und k als neuer Vorgänger $V_{k'}$ von k' . Nachdem diese Operation für alle von k ausgehenden Kanten ausgeführt wurde, wird k als abschließend bearbeitet markiert und der aktuelle Schritt endet. Nach $|V|$ Schritten endet das Verfahren.

Nach Abschluss müssen wir die für jeden Knoten erhaltene Information noch interpretieren. Wir sehen unmittelbar die Länge L_k des kürzesten gefundenen Weges von 1 nach k und den Vorgänger V_k . Wir können dann den Weg von 1 nach k rückwärts rekonstruieren, indem wir von k ausgehend den Vorgänger V_k einfügen und dann den kürzesten gefundenen Weg von 1 nach V_k ermitteln. So werden wir als nächstes V_{V_k} einfügen und wiederholen diese Vorgehensweise, bis wir Knoten 1 einfügen. Sollte für einen Knoten k nach Abschluss des Verfahrens noch $V_k = -$ gelten, dann gibt es keinen Weg von Knoten 1 nach Knoten k .

Salopp formuliert: Verfahren von Dijkstra

Wir prüfen in jedem Schritt, ob wir von einem Knoten k aus durch Ergänzen einer Kante neue kürzeste Wege zu anderen Knoten finden können. Knoten k wird so gewählt, dass wir den kürzesten Weg zu ihm bereits kennen.

Das gesamte Verfahren lässt sich als Algorithmus 5.1 konkretisieren.

Algorithmus 5.1 (Dijkstra).

Eingabe: Gerichteter Graph $G = (V, E, w)$ mit nicht-negativen Kantengewichten
Ausgabe: Kürzester Weg von Knoten 1 zu jedem Knoten j oder Information, dass keiner existiert

1. Setze $L_1 \leftarrow 0$ und $L_k \leftarrow \infty$ für $k \neq 1$, $V_k \leftarrow -$ für jeden Knoten k , $V' \leftarrow V$
2. Wähle Knoten $k = \arg \min\{L_{k'} \mid k' \in V'\}$
3. Für jede Kante $(k, k') \in E$ mit $k' \in V'$ und $L_{k'} > L_k + w_{k,k'}$
 - $L_{k'} \leftarrow L_k + w_{k,k'}$
 - $V_{k'} \leftarrow k$
4. $V' \leftarrow V' \setminus \{k\}$
5. Wenn $V' \neq \emptyset$, gehe zu 2.

Nach der Initialisierung werden die Schritte 2. bis 5. wiederholt, bis alle Knoten abschließend bearbeitet sind, was wir an $V' = \emptyset$ erkennen. In jedem Durchlauf wird Knoten k bestimmt und jede Kante (k, k') daraufhin untersucht, ob sie zu einem neuen kürzesten bisher gefundenen Weg zu k' führt. Wenn dies der Fall ist, dann werden die entsprechenden Informationen für Knoten k' aktualisiert.

Beispiel

Bevor wir das Verfahren analysieren, wollen wir es an einem Beispiel veranschaulichen. Wir wenden Algorithmus 5.1 auf die Problem Instanz in Abb. 5.2 an.

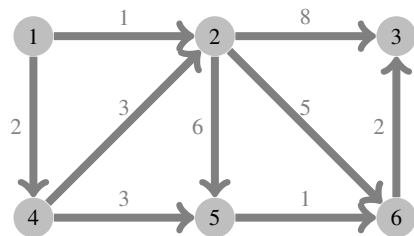


Abb. 5.2 Instanz von Problem 5.1 (Kürzester Pfad in einem gerichteten Graphen) mit nicht-negativen Kantengewichten

Tabelle 5.1 stellt den Ablauf des Verfahrens dar. Die jeweiligen kürzesten bisher gefundenen Wege nach den Iterationen 1 bis 6 sind in Abb. 5.3 bis Abb. 5.8 illustriert.

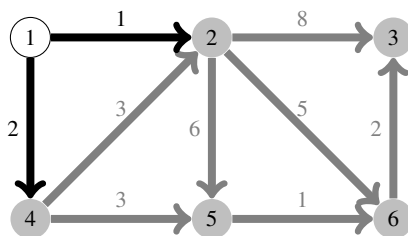
In Tabelle 5.1 korrespondiert jede der nummerierten Zeilen zur Betrachtung eines Knotens k als Ausgangspunkt für zu ergänzende Kanten. Es finden sich jeweils die Werte L_1 bis L_6 , V_1 bis V_6 und die Menge V' vor Wahl des Knoten k , der gewählte Knoten k und ein Verweis auf eine Abbildung mit den resultierenden kürzesten bisher gefundenen Wegen. Die resultierenden Werte für L_1 bis L_6 und V_1 bis V_6 finden sich dann in der folgenden Zeile.

Wir wollen nun nachvollziehen, was in den einzelnen Iterationen geschieht. Wir initialisieren zunächst L_1 bis L_6 , V_1 bis V_6 und V' wie in Zeile 1 von Tabelle 5.1

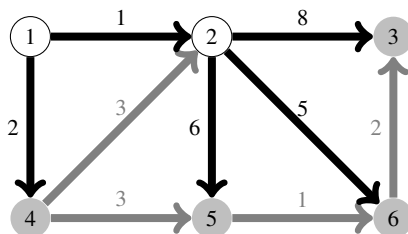
Tabelle 5.1 Ablauf von Algorithmus 5.1 für die Instanz in Abb. 5.2

	L_1	L_2	L_3	L_4	L_5	L_6	V_1	V_2	V_3	V_4	V_5	V_6	V'	k	Abb.
1	0	∞	∞	∞	∞	∞	—	—	—	—	—	—	$\{1, \dots, 6\}$	1	5.3
2	0	1	∞	2	∞	∞	—	1	—	1	—	—	$\{2, \dots, 6\}$	2	5.4
3	0	1	9	2	7	6	—	1	2	1	2	2	$\{3, \dots, 6\}$	4	5.5
4	0	1	9	2	5	6	—	1	2	1	4	2	$\{3, 5, 6\}$	5	5.6
5	0	1	9	2	5	6	—	1	2	1	4	2	$\{3, 6\}$	6	5.7
6	0	1	8	2	5	6	—	1	6	1	4	2	$\{3\}$	3	5.8
	0	1	8	2	5	6	—	1	6	1	4	2	\emptyset		

notiert. Wir wählen dann $k = 1$ und finden bessere Wege für $k' = 2$ und $k' = 4$, da $L_2 = \infty > L_1 + w_{1,2} = 0 + 1 = 1$, d. h. der kürzeste Weg von 1 nach 1 ergänzt um die Kante von 1 nach 2 ergibt einen Weg von 1 nach 2, der kürzer als der kürzeste bisher gefundene ist, und $L_4 = \infty > L_1 + w_{1,4} = 0 + 2 = 2$. Diese Wege sind in Abb. 5.3 durch schwarze Kanten hervorgehoben. Knoten werden weiß dargestellt, wenn sie nicht mehr berücksichtigt werden müssen.

**Abb. 5.3** Kürzeste bisher gefundene Wege nach Iteration 1

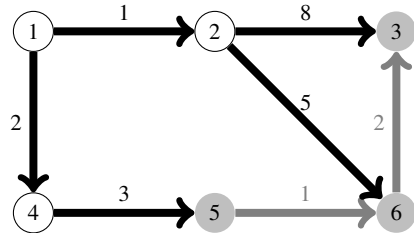
In Iteration 2 wählen wir $k = 2$, da es von Knoten 1 nach Knoten 2 unter allen Knoten in V' den kürzesten gefundenen Weg gibt, und finden bessere Wege für $k' = 3$, $k' = 5$ und $k' = 6$, da $L_3 = \infty > L_2 + w_{2,3} = 1 + 8 = 9$, $L_5 = \infty > L_2 + w_{2,5} = 1 + 6 = 7$ und $L_6 = \infty > L_2 + w_{2,6} = 1 + 5 = 6$, siehe Abb. 5.4.

**Abb. 5.4** Kürzeste bisher gefundene Wege nach Iteration 2

In Iteration 3 wählen wir $k = 4$ und finden einen kürzeren Weg für $k' = 5$, da $L_5 = 7 > L_4 + w_{4,5} = 2 + 3 = 5$. Kante $(4, 2)$ wird nicht berücksichtigt, da $2 \notin V'$. Kante $(4, 2)$ wird somit in keinem kürzesten Weg zu finden sein und wir bilden sie

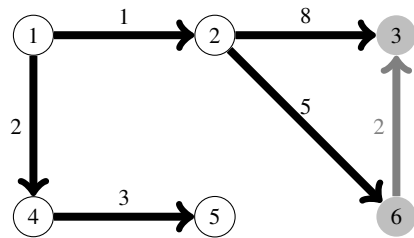
in Abb. 5.5 bis 5.8 dementsprechend auch nicht mehr ab. Außerdem wissen wir nun, dass die Kante $(2,5)$ nicht benötigt wird, da wir Knoten 5 über die Kante $(4,5)$ auf einem kürzeren Weg erreichen.

Abb. 5.5 Kürzeste bisher gefundene Wege nach Iteration 3



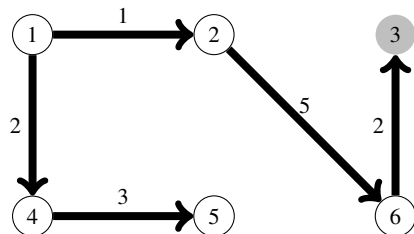
In Iteration 4 wählen wir $k = 5$, finden allerdings keine besseren Wege, da $L_6 = 6 = L_5 + w_{5,6} = 5 + 1 = 6$. Kante $(5,6)$ wird somit in keinem (ausgewiesenen) kürzesten Weg zu finden sein, siehe Abb. 5.6.

Abb. 5.6 Kürzeste bisher gefundene Wege nach Iteration 4



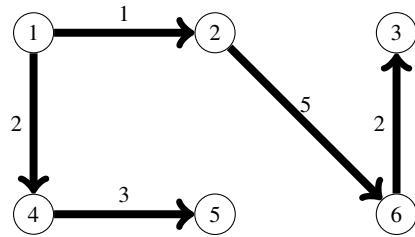
In Iteration 5 wählen wir $k = 6$ und finden einen kürzeren Weg für $k' = 3$, da $L_3 = 9 > L_6 + w_{6,3} = 6 + 2 = 8$. Kante $(2,3)$ wird somit in keinem kürzesten Weg zu finden sein, siehe Abb. 5.7.

Abb. 5.7 Kürzeste bisher gefundene Wege nach Iteration 5



In Iteration 6 wählen wir $k = 3$ und finden keinen kürzeren Weg mehr, da Knoten $k = 3$ keine ausgehenden Kanten hat. Alle kürzesten Wege finden sich dann abschließend in Abb. 5.8.

Abb. 5.8 Kürzeste bisher gefundene Wege nach Iteration 6



Analyse

Nachdem wir gesehen haben, dass Algorithmus 5.1 die Probleminstanz in Abb. 5.2 löst, besagt Theorem 5.1, dass er dies tatsächlich für jede Instanz von Problem 5.1 (Kürzester Pfad in einem gerichteten Graphen) mit nicht-negativen Kantengewichten leistet.

Theorem 5.1. *Algorithmus 5.1 findet einen kürzesten Weg von Knoten 1 zu jedem anderen Knoten j , zu dem es einen Weg von Knoten 1 gibt, wenn kein Kantengewicht negativ ist.*

Einen formalen Beweis führen wir in Anhang C.1. Hier wollen wir allerdings schon intuitiv Analogien zu dem Ansatz der DP in Abschnitt 4.4 aufzeigen. Das DP-Verfahren bestimmt schon vorab eine Reihenfolge der Knoten, sodass keine Kante im Graph von einem späteren Knoten in der Reihenfolge zu einem früheren verläuft. Dies ist hier nicht möglich, da es Kreise geben kann. Algorithmus 5.1 bestimmt daher dynamisch durch die Wahl von Knoten k in jeder Iteration eine Reihenfolge der Knoten, sodass keine „relevante“ Kante im Graph von einem späteren Knoten in der Reihenfolge zu einem früheren verläuft. Dabei sortiert er die Knoten nach den Längen der kürzesten Wege zu ihnen nicht-absteigend. Eine Kante von einem späteren Knoten k zu einem früheren Knoten k' ist dann „irrelevant“, weil sie den ohnehin längeren Weg zu k nicht verbessern kann, da ihr Gewicht nicht-negativ ist.

5.1.2 Allgemeine Graphen

In Abschnitt 5.1.1 haben wir uns auf Graphen beschränkt, in denen es keine negativen Kantengewichte gibt. Dies ermöglicht uns, dynamisch die Reihenfolge zu bestimmen, in der wir für Knoten den bis dato kürzesten gefundenen Weg zu dem tatsächlich kürzesten Weg erklären. Wenn Kantengewichte auch negativ sein können, funktioniert dieser Ansatz nicht.

Das Verfahren, das in diesem Abschnitt vorgestellt wird, geht auf Floyd (1962) zurück, wurde aber in ähnlicher Form auch von Warshall (1962) vorgeschlagen. Daher wird es in der einschlägigen Literatur meistens als der Algorithmus von Floyd und Warshall oder Floyd-Warshall-Algorithmus bezeichnet. Der Floyd-Warshall-Algorithmus zielt darauf ab, für jedes Paar i und j von Knoten den kürzesten Weg

von i nach j zu ermitteln. Dabei wird dann Problem 5.1 (Kürzester Weg in einem gerichteten Graphen) implizit gelöst. Wir müssen allerdings eine Besonderheit beachten, die bisher nicht relevant war. Wenn Kantengewichte negativ sein können und Kreise nicht ausgeschlossen sind, dann kann es Kreise negativer Länge geben wie in dem Beispiel in Abb. 5.9. Der Weg $(1, 4, 5, 2, 1)$ ist ein Kreis negativer Länge.

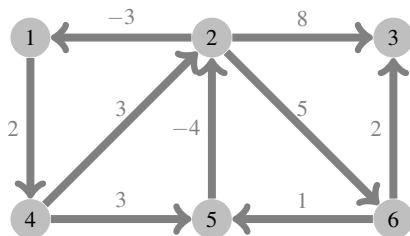


Abb. 5.9 Instanz von Problem 5.1 (Kürzester Pfad in einem gerichteten Graphen)

Wenn es Kreise negativer Länge gibt, müssen wir zwischen Wegen und Pfaden unterscheiden, denn es gibt Wege, die kürzer sind, als der kürzeste Pfad. Es gibt einen Weg von i nach j , der kürzer als der kürzeste Pfad von i nach j ist, wenn es einen Pfad von i nach j gibt, der über einen Knoten k läuft, der in einem Kreis negativer Länge liegt. Wir können dann einen beliebig kurzen Weg konstruieren, indem wir von i nach k gehen, dann von Knoten k aus beliebig oft den Kreis ablaufen, bevor wir von k nach j gehen.

In dem Beispiel in Abb. 5.9 gibt es zwei Pfade von Knoten 6 nach Knoten 3. Der kürzere der beiden führt lediglich über die Kante $(6, 3)$ und hat eine Länge von zwei. Der längere Pfad $(6, 5, 2, 3)$ hat eine Länge von fünf, hat aber Knoten mit dem Kreis negativer Länge gemein. Die Wege $(6, 5, 2, 1, 4, 5, 2, 3)$ und $(6, 5, 2, 1, 4, 5, 2, 1, 4, 5, 2, 3)$ haben Längen von drei und eins. Somit ist der Weg $(6, 5, 2, 1, 4, 5, 2, 1, 4, 5, 2, 3)$ kürzer als der kürzeste Pfad. Durch weitere Durchläufe des Kreises negativer Länge können wir beliebig kurze Wege finden.

Zusammenfassung: Kürzeste Wege und kürzeste Pfade

Wenn es in Graph G einen Kreis negativer Länge gibt, dann gibt es mindestens ein Paar von Knoten i und j , sodass es keinen kürzesten Weg von i nach j gibt. Wenn es einen Weg von i nach j gibt, der kürzer als der kürzeste Pfad von i nach j ist, dann gibt es keinen kürzesten Weg von i nach j .

Das bedeutet, dass der kürzeste Weg von jedem Knoten i zu jedem anderen Knoten j nur dann bestimmt werden kann, wenn es keine Kreise negativer Länge gibt. In diesem Fall ist der kürzeste Pfad von i nach j auch ein kürzester Weg von i nach j . Dabei sollten wir uns vor Augen halten, dass Kreise negativer Länge eher selten vorkommen werden, wenn wir in dem Graph direkt einen Sachverhalt aus der Anwendung abbilden.

An dieser Stelle soll noch kurz erwähnt werden, dass man in einem leicht modifizierten Problem nach dem kürzesten Pfad von i nach j fragen könnte. Ein sol-

cher Pfad existiert auch in einem Graph mit Kreisen negativer Länge, sofern es überhaupt einen Weg von i nach j gibt. Trotz seiner großen Ähnlichkeit zu Problem 5.1 (Kürzester Weg in einem gerichteten Graphen) ist das modifizierte Problem bedeutend schwerer zu lösen und wird im Weiteren nicht behandelt.

Überblick über das Verfahren

Wir nehmen an, dass die Knoten durch die Zahlen $1, \dots, |V|$ benannt sind, und bestimmen in $|V|$ Iterationen Wege von jedem Knoten i zu jedem Knoten j unter bestimmten Einschränkungen. Diese Einschränkungen werden dabei von Iteration zu Iteration weniger streng, sodass in der letzten Iteration ein Weg von jedem Knoten i zu jedem Knoten j ohne weitere Einschränkungen bestimmt wird. Die Einschränkung besteht in der Menge von Knoten, die die jeweiligen Wege (zusätzlich zu i und j) durchlaufen dürfen. In Iteration k ist diese Menge auf die Knoten $1, \dots, k$ beschränkt.

Die Länge des kürzesten Weges von i zu j , den wir in Iteration k bestimmen, bezeichnen wir mit $L_{i,j,k}$. Mit

$$L_{i,j,0} = \begin{cases} 0 & \text{wenn } i = j \\ w_{i,j} & \text{wenn } (i,j) \in E \\ \infty & \text{sonst} \end{cases}$$

initialisieren wir Wert $L_{i,j,0}$ für jedes Paar von Knoten i und j mit der Länge eines kürzesten Weges von i zu j , der nicht über weitere Knoten läuft. Einen solchen Weg gibt es nur, wenn es eine Kante $(i,j) \in E$ gibt, dann hat er Länge $w_{i,j}$, oder wenn $i = j$, dann hat er eine Länge von null. Diese Werte stellen den Kern eines rekursiven Zusammenhangs dar, der dann in den Iterationen k , $k = 1, \dots, |V|$, ausgewertet wird.

In Iteration k betrachten wir die Länge eines Weges von i nach j , der außer über i und j nur über Knoten in $\{1, \dots, k\}$ führt. Dabei müssen nicht zwangsläufig alle Knoten in $\{1, \dots, k\} \setminus \{i, j\}$ besucht werden. Solch ein Weg führt entweder über k oder nicht. Die erste Alternative bewerten wir mit $L_{i,k,k-1} + L_{k,j,k-1}$, da wir den kürzesten bekannten Weg von i nach k um den kürzesten bekannten Weg von k nach j ergänzen können, um einen Weg von i nach j über k zu konstruieren. Der konstruierte Weg führt nur über Knoten in $\{i, j, 1, \dots, k\}$, denn die jeweiligen Teilwege führen außer über i , j und k nur über Knoten in $\{1, \dots, k-1\}$. Die zweite Alternative bewerten wir mit $L_{i,j,k-1}$, d. h., wir übernehmen den bekannten Weg, der nicht über k läuft. Unter diesen beiden Alternativen wählen wir die kürzere, d. h., wir erhalten

$$L_{i,j,k} = \min \{L_{i,k,k-1} + L_{k,j,k-1}, L_{i,j,k-1}\}$$

für alle Paare von Knoten i und j und alle $k = 1, \dots, |V|$. Wenn es keinen Weg von i nach j nur über Knoten in $\{1, \dots, k-1\}$ gibt, dann gilt $L_{i,j,k-1} = \infty$. Wenn es keinen Weg von i nach k nur über Knoten in $\{1, \dots, k-1\}$ oder keinen Weg von k nach j

nur über Knoten in $\{1, \dots, k-1\}$ gibt, dann gilt $L_{i,k,k-1} + L_{k,j,k-1} = \infty$. Wir erhalten also $L_{i,j,k} = \infty$, wenn es keinen Weg von i nach j nur über Knoten in $\{1, \dots, k\}$ gibt.

Diese Abwägung wird für jedes Paar von Knoten i und j vorgenommen. Die Länge des kürzesten bekannten Weges sinkt tendenziell im Verlauf der Iterationen, d. h., wir haben $L_{i,j,k-1} \geq L_{i,j,k}$, da die Einschränkungen für die ermittelten Wege abnehmen. Wir merken uns zusätzlich noch laufend den größten Knoten $k(i, j)$, den wir in den kürzesten bekannten Weg von i nach j aufgenommen haben. Dieser Wert wurde ursprünglich mit $k(i, j) = 0$ für alle Paare von Knoten i und j initialisiert. Nach $|V|$ Iterationen endet das Verfahren mit den Werten $L_{i,j,|V|}$ und $k(i, j)$ für jedes Paar von Knoten i und j .

Nach Abschluss müssen wir die für jedes Paar von Knoten erhaltene Information noch interpretieren. Wir sehen unmittelbar die Länge $L_{i,j,|V|}$ des kürzesten gefundenen Weges von i nach j und den größten Knoten $k(i, j)$ auf diesem Weg (oder $k(i, j) = 0$). Wenn $L_{i,j,|V|} < \infty$, d. h. wenn wir überhaupt einen Weg von i nach j gefunden haben, wissen wir, dass der kürzeste gefundene Weg von i über potentiell mehrere Knoten zu $k(i, j)$ und dann weiter über potentiell mehrere Knoten zu j verläuft (oder dass er direkt über die Kante (i, j) führt, wenn $k(i, j) = 0$). Die beiden Abschnitte von i nach k und von k nach j werden von den jeweils kürzesten gefundenen Wegen von i nach k und von k nach j gebildet und analog interpretiert.

Salopp formuliert: Verfahren von Floyd und Warshall

Wir bestimmen in Iteration k einen kürzesten Weg von i nach j , der nur über Knoten in $\{1, \dots, k\}$ läuft. Indem wir k erhöhen und somit den Freiheitsgrad für die Konstruktion von Wegen erhöhen, verkürzen wir schrittweise die bekannten Wege. Nach Iteration $|V|$ erhalten wir dann die kürzesten gefundenen Wege (ohne Einschränkung auf Teilmengen von Knoten).

Das Verfahren lässt sich als Algorithmus 5.2 konkretisieren.

Algorithmus 5.2 (Floyd-Warshall).

Eingabe: Gerichteter Graph $G = (V, E, w)$

Ausgabe: Kürzester Weg von jedem Knoten i zu jedem Knoten j oder Information, dass keiner existiert

1. Setze $L_{i,j,0} \leftarrow w_{i,j}$ für alle $(i, j) \in E$, $L_{i,i,0} \leftarrow 0$ für alle $i \in V$, $L_{i,j,0} \leftarrow \infty$ für alle $(i, j) \notin E$ mit $i \neq j$, $k(i, j) \leftarrow 0$ für alle $i, j \in V$ und $k \leftarrow 1$
2. Setze $L_{i,j,k} = \min \{L_{i,k,k-1} + L_{k,j,k-1}, L_{i,j,k-1}\}$ für jedes Paar von Knoten i und j und aktualisiere ggf. $k(i, j)$.
3. Wenn $k < |V|$, setze $k \leftarrow k + 1$ und gehe zu 2.
4. Interpretiere $L_{i,j,|V|}$ und $k(i, j)$ für alle Paare von Knoten i und j .

Beispiel

Bevor wir das Verfahren analysieren, wollen wir es an einem Beispiel veranschaulichen. Wir wenden Algorithmus 5.2 auf die Problem Instanz in Abb. 5.10 an.

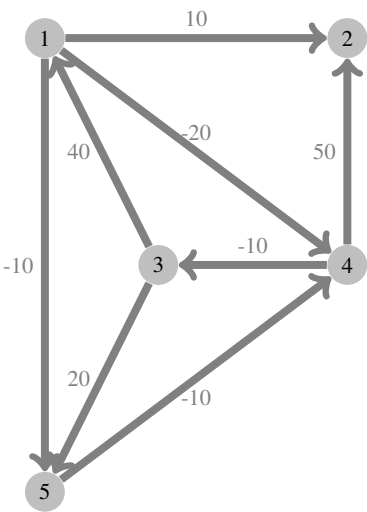


Abb. 5.10 Instanz von Problem 5.1 (Kürzester Pfad in einem gerichteten Graphen)

Durch „scharfes Hinsehen“ können wir erkennen, dass es keine Kreise negativer Länge gibt. Tabellen 5.2 bis 5.6 stellen den Ablauf des Verfahrens dar, indem die Werte von $L_{i,j,0}$ bis $L_{i,j,5}$ auf der linken Seite und die Werte von $k_{i,j}$ auf der rechten Seite abgebildet werden.

Tabelle 5.2 Initialisierung durch Algorithmus 5.2 für die Instanz in Abb. 5.10

$L_{i,j,0}$	1	2	3	4	5	$k(i,j)$	1	2	3	4	5
1	0	10	∞	-20	-10	1	0	0	0	0	0
2	∞	0	∞	∞	∞	2	0	0	0	0	0
3	40	∞	0	∞	20	3	0	0	0	0	0
4	∞	50	-10	0	∞	4	0	0	0	0	0
5	∞	∞	∞	-10	0	5	0	0	0	0	0

Wir initialisieren die Tabellen (die dargestellten Werte) wie oben besprochen und erhalten Tabelle 5.2. Für alle Kanten wird deren Länge in die entsprechende Zelle auf der linken Seite eingetragen. Die Werte in den übrigen Zellen auf der linken Seite werden mit ∞ und alle Werte auf der rechten Seite mit null initialisiert.

Nach der ersten Iteration von Algorithmus 5.2 wurden einzelne kürzere Wege gefunden, nämlich von 3 nach 2 und von 3 nach 4. Dass nur Wege, die von Knoten 3 ausgehen, verbessert werden, liegt daran, dass nur von Knoten 3 eine Kante zu Knoten 1 führt, wir in Iteration 1 nur Wege über Knoten 1 suchen und noch keine Wege, die aus mehr als einer Kante bestehen, bekannt sind. In Iteration 2 werden keine weiteren Verbesserungen erzielt. Dies liegt daran, dass von Knoten 2 keine Kanten ausgehen und wir daher keine Wege, die über Knoten 2 führen, konstruieren können.

Tabelle 5.3 Werte nach erster und zweiter Iteration von Algorithmus 5.2 für die Instanz in Abb. 5.10

$L_{i,j,1}/L_{i,j,2}$	1	2	3	4	5	$k(i,j)$	1	2	3	4	5
1	0	10	∞	-20	-10	1	0	0	0	0	0
2	∞	0	∞	∞	∞	2	0	0	0	0	0
3	40	50	0	20	20	3	0	1	0	1	0
4	∞	50	-10	0	∞	4	0	0	0	0	0
5	∞	∞	∞	-10	0	5	0	0	0	0	0

Tabelle 5.4 Werte nach dritter Iteration von Algorithmus 5.2 für die Instanz in Abb. 5.10

$L_{i,j,3}$	1	2	3	4	5	$k(i,j)$	1	2	3	4	5
1	0	10	∞	-20	-10	1	0	0	0	0	0
2	∞	0	∞	∞	∞	2	0	0	0	0	0
3	40	50	0	20	20	3	0	1	0	1	0
4	30	40	-10	0	10	4	3	3	0	0	3
5	∞	∞	∞	-10	0	5	0	0	0	0	0

Nach der dritten Iteration von Algorithmus 5.2 wurden wiederum einzelne kürzere Wege gefunden, nämlich von 4 nach 1, von 4 nach 2 und von 4 nach 5.

Tabelle 5.5 Werte nach vierter Iteration von Algorithmus 5.2 für die Instanz in Abb. 5.10

$L_{i,j,4}$	1	2	3	4	5	$k(i,j)$	1	2	3	4	5
1	0	10	-30	-20	-10	1	0	0	4	0	0
2	∞	0	∞	∞	∞	2	0	0	0	0	0
3	40	50	0	20	20	3	0	1	0	1	0
4	30	40	-10	0	10	4	3	3	0	0	3
5	20	30	-20	-10	0	5	4	4	4	0	0

In der vierten Iteration von Algorithmus 5.2 werden wiederum einzelne kürzere Wege gefunden, nämlich von 1 nach 3, von 5 nach 1, von 5 nach 2 und von 5 nach 3. Zudem wird in dieser Iteration durch $i = 5$, $j = 5$ und $k = 4$ ein Kreis der Länge null identifiziert, denn wir sehen, dass wir von $i = 5$ zu $k = 4$ und weiter (zurück) zu $j = 5$ gehen können und dabei eine Strecke von null zurücklegen. In Iteration 5 wird ein einzelner kürzerer Weg gefunden, nämlich der von $i = 3$ nach $j = 4$. Weiterhin werden einige neue Wege gefunden, die so lang wie die bisher kürzesten gefundenen sind, z. B. durch $i = 1$, $j = 4$ und $k = 5$. In diesen Fällen ist aber keine Aktualisierung nötig.

Es bleibt, die Ergebnisse zu interpretieren. In Tabelle 5.7 werden die gefundenen Wege von jedem Knoten i zu jedem Knoten j dargestellt.

Wir wollen am Beispiel des Weges von 5 nach 2 erläutern, wie wir die Ergebnisse in Tabelle 5.6 interpretieren. Wir erkennen an der rechten Seite von Tabelle 5.6 mit Eintrag $k(5,2) = 4$, dass der kürzeste gefundene Weg von 5 nach 2 über Knoten 4 läuft und somit aus zwei Teilwegen von 5 nach 4 und von 4 nach 2 besteht. Diese

Tabelle 5.6 Werte nach fünfter Iteration von Algorithmus 5.2 für die Instanz in Abb. 5.10

$L_{i,j,5}$	1	2	3	4	5	$k(i,j)$	1	2	3	4	5
1	0	10	-30	-20	-10	1	0	0	4	0	0
2	∞	0	∞	∞	∞	2	0	0	0	0	0
3	40	50	0	10	20	3	0	1	0	5	0
4	30	40	-10	0	10	4	3	3	0	0	3
5	20	30	-20	-10	0	5	4	4	4	0	0

Tabelle 5.7 Kürzeste durch Algorithmus 5.2 gefundene Wege für die Instanz in Abb. 5.10

Wege	1	2	3	4	5
1	(1)	(1,2)	(1,4,3)	(1,4)	(1,5)
2	–	(2)	–	–	–
3	(3,1)	(3,1,2)	(3)	(3,5,4)	(3,5)
4	(4,3,1)	(4,3,1,2)	(4,3)	(4)	(4,3,5)
5	(5,4,3,1)	(5,4,3,1,2)	(5,4,3)	(5,4)	(5)

Teilwege müssen wir jeweils auch ableiten. Der Teilweg von 5 nach 4 besteht nur aus der Kante (5,4), denn in diesen Weg wurden keine weiteren Knoten eingefügt ($k(5,4) = 0$). Der Teilweg von 4 nach 2 verläuft über Knoten 3, denn $k(4,2) = 3$, und besteht somit wiederum aus zwei Teilwegen von 4 nach 3 und von 3 nach 2. Der Teilweg von 4 nach 3 besteht nur aus der Kante (4,3), denn in diesen Weg wurden keine weiteren Knoten eingefügt ($k(4,3) = 0$). Der Teilweg von 3 nach 2 verläuft über Knoten 1, denn $k(3,2) = 1$, und besteht somit aus zwei Teilwegen von 3 nach 1 und von 1 nach 2. Diese Teilwege bestehen aus den einzelnen Kanten (3,1) und (1,2). Somit ergibt sich der gefundene Weg von 5 nach 2 als (5,4,3,1,2).

Analyse

Wir können durch „scharfes Hinsehen“ erkennen, dass Algorithmus 5.2 für jedes Paar von Knoten in der Problem Instanz in Abb. 5.10 den kürzesten Weg ermittelt. Theorem 5.2 besagt, dass er dies tatsächlich für jede Instanz von Problem 5.1 (Kürzester Pfad in einem gerichteten Graphen) leistet, sofern es möglich ist.

Theorem 5.2. *Algorithmus 5.2 findet einen kürzesten Weg von Knoten i zu Knoten j , falls es einen kürzesten Weg von Knoten i zu Knoten j gibt, und gibt $L_{i,j,|V|} = \infty$ genau dann aus, wenn es keinen Weg von i zu j gibt.*

Den Beweis führen wir in Anhang C.2. Theorem 5.2 gibt uns unmittelbar Gewissheit über die Aussagekraft der Ergebnisse von Algorithmus 5.2 für Graphen, die keine Kreise negativer Länge enthalten. Denn in solchen Graphen gibt es von jedem Knoten i zu jedem Knoten j entweder keinen Weg oder einen kürzesten Weg. Wir wissen in diesem Fall also, dass $L_{i,j,|V|} < \infty$ die Länge eines kürzesten Weges von i nach j angibt und $L_{i,j,|V|} = \infty$ impliziert, dass es keinen Weg von i nach j gibt.

Wenn wir aber gar nicht wissen, ob der vorliegende Graph Kreise negativer Länge enthält, dann wäre es schön, wenn wir einen Indikator dafür hätten, ob es welche gibt. Wenn dies der Fall ist, müssen wir das Ergebnis von Algorithmus 5.2 nämlich vorsichtig interpretieren, denn ein ermittelter Weg von i nach j muss nicht zwangsläufig ein kürzester Weg sein. Tatsächlich können wir uns auch für die Identifikation von Kreisen negativer Länge an dem Ergebnis von Algorithmus 5.2 orientieren, wie das folgende Theorem besagt.

Theorem 5.3. *In jedem Kreis negativer Länge gibt es mindestens einen Knoten i mit $L_{i,i,|V|} < 0$.*

Den Beweis führen wir in Anhang C.3. Theorem 5.3 können wir nutzen, um zu ermitteln, ob wir $L_{i,j,|V|}$ als kürzesten Weg von i und j interpretieren können oder nicht. Wir wollen unser bisheriges Wissen kurz zusammenfassen.

- Wenn es keinen Weg von i zu j gibt, dann gibt Algorithmus 5.2 als Ergebnis $L_{i,j,|V|} = \infty$ aus.
- Es gibt einen kürzesten Weg von i nach j genau dann, wenn es einen Weg von i nach j gibt, es aber keinen Kreis negativer Länge gibt, der von i aus erreicht werden kann und von dem aus man j erreichen kann.
- In jedem Kreis negativer Länge gibt es mindestens einen Knoten k mit $L_{k,k,|V|} < 0$. Den Wert $L_{k,k,|V|} < 0$ können wir als Beleg für die Existenz des Kreises negativer Länge sehen.
- Wir können einen Kreis C negativer Länge von i aus genau dann erreichen, wenn wir jeden Knoten in C von i aus erreichen können, denn wenn wir einen Knoten des Kreises erreichen, können wir alle erreichen, indem wir den Kreis ablaufen.
- Analog können wir j von einem Kreis C negativer Länge aus genau dann erreichen, wenn wir j von jedem Knoten in C aus erreichen können.

Dieses Wissen können wir wie folgt zusammenfassen.

Theorem 5.4. *Die Werte $L_{i,j,|V|}$ und $k(i, j)$ repräsentieren einen kürzesten Weg von i nach j genau dann, wenn $L_{i,j,|V|} \neq \infty$ gilt und es keinen Knoten k gibt mit $L_{k,k,|V|} < 0$, $L_{i,k,|V|} \neq \infty$ und $L_{k,j,|V|} \neq \infty$.*

Wir verzichten auf einen formalen Beweis und erläutern das Theorem nur kurz. Gemäß Theorem 5.2 belegt $L_{i,j,|V|} \neq \infty$, dass es einen Weg von i nach j gibt. Wenn es einen Kreis negativer Länge gibt, dann muss es gemäß Theorem 5.3 einen Knoten k mit $L_{k,k,|V|} < 0$ in ihm geben. Der ganze Kreis ist genau dann von i aus erreichbar, wenn k von i aus erreichbar ist. Knoten k ist gemäß Theorem 5.2 von i aus genau dann erreichbar, wenn $L_{i,k,|V|} \neq \infty$. Analog gilt, dass j von dem Kreis aus genau dann erreichbar ist, wenn $L_{k,j,|V|} \neq \infty$.

Wir wollen die Bedeutung von Theorem 5.4 noch an einem Beispiel verdeutlichen. Wir betrachten dafür die Instanz in Abb. 5.11.

Durch „scharfes Hinsehen“ können wir schon einen Kreis negativer Länge erkennen. Tabelle 5.8 stellt das Ergebnis von Algorithmus 5.2 dar.

Das Ergebnis in Tabelle 5.8 wollen wir uns etwas genauer anschauen. Die Knoten 1 und 2 sind beide als Knoten in einem Kreis negativer Länge erkennbar, da $L_{1,1,5} <$

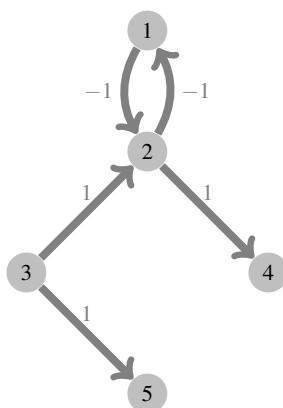


Abb. 5.11 Instanz von Problem 5.1 (Kürzester Pfad in einem gerichteten Graphen)

Tabelle 5.8 Ergebnis von Algorithmus 5.2 für die Instanz in Abb. 5.11

$L_{i,j,5}$	1	2	3	4	5
1	-2	-3	∞	0	∞
2	-3	-4	∞	-1	∞
3	0	-1	0	2	1
4	∞	∞	∞	0	∞
5	∞	∞	∞	∞	0

$k(i,j)$	1	2	3	4	5
1	2	2	0	2	0
2	2	2	0	2	0
3	2	2	0	2	0
4	0	0	0	0	0
5	0	0	0	0	0

0 und $L_{2,2,5} < 0$. Der Kreis, der Knoten 1 und 2 enthält, ist der einzige Kreis und damit auch der einzige Kreis negativer Länge. Knoten 3, 4 und 5 sind in keinem Kreis negativer Länge enthalten. Dennoch gibt es zwischen diesen Knoten nur einen kürzesten Weg. Von 4 nach 3, von 4 nach 5, von 5 nach 3 und von 5 nach 4 gibt es überhaupt keinen Weg. Von 3 nach 4 gibt es einen Weg. Der Kreis (2, 1, 2) mit negativer Länge kann von 3 aus erreicht werden und von dem Kreis aus kann 4 erreicht werden. Daher gibt es beliebig kurze Wege von 3 nach 4. Von dem Kreis aus kann man allerdings nicht Knoten 5 erreichen. Da es keinen anderen Kreis negativer Länge gibt und es einen Weg von 3 nach 5 gibt, gibt es auch einen kürzesten Weg von 3 nach 5. Wir können dies auch unter Verwendung von Theorem 5.4 erkennen. Da $L_{1,5,5} = L_{2,5,5} = \infty$ sowie $L_{3,3,5} \geq 0$, $L_{4,4,5} \geq 0$ und $L_{5,5,5} \geq 0$, gibt es keinen Knoten k mit $L_{k,k,5} < 0$, $L_{3,k,5} \neq \infty$ und $L_{k,5,5} \neq \infty$.

5.2 Minimale Spannbäume

In diesem Abschnitt wollen wir uns mit der Bestimmung von Spannbäumen beschäftigen. Was Bäume sind, haben wir bereits in Definition 2.10 auf S. 25 spezifiziert.

Definition 5.1 (Spannbaum). Ein Spannbaum eines Graphen $G = (V, E)$ ist eine Teilmenge $E' \subseteq E$ von Kanten, sodass (V, E') ein Baum ist.

Definition 5.2 (Gewicht eines Spannbaums). Das Gewicht eines Spannbaums E' eines gewichteten Graphen $G = (V, E, w)$ ist die Summe der Gewichte der Kanten in E' .

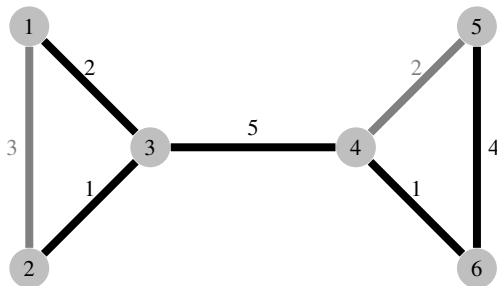


Abb. 5.12 Graph und Spannbaum

Abb. 5.12 stellt einen Graphen G dar, in dem es neun verschiedene Spannäume gibt. Graph G enthält zwei Kreise mit jeweils drei Knoten. Wir wissen, dass ein Baum eine Kante weniger als Knoten hat und somit ein Spannbaum des Graphen in Abb. 5.12 fünf Kanten hat. Es gibt also nur zwei Kanten in $E \setminus E'$ und jeweils eine muss in jedem Kreis liegen, denn sonst ist (V, E') nicht kreisfrei. Wir haben neun Möglichkeiten, diese beiden Kanten zu wählen. Der schwarz in Abb. 5.12 hervorgehobene Spannbaum hat ein Gewicht von $2 + 1 + 5 + 1 + 4 = 13$.

Spannbäume haben z. B. Anwendungen bei der Installation von Energie- oder Kommunikationsnetzwerken, bei denen gewährleistet sein soll oder muss, dass alle Knoten untereinander durch das Netzwerk direkt oder indirekt miteinander verbunden sind. Wenn die Einrichtung verschiedener Verbindungen unterschiedliche Kosten verursacht, ist es von Interesse, das Netzwerk zu wählen, das die geringsten Gesamtkosten verursacht. Genau diese Aufgabe findet sich in der folgenden Problemstellung.

Problem 5.2 (Minimaler Spannbaum).

Eingabe: (Ungerichteter) Gewichteter Graph G

LR: Spannbäume von G

ZB: –

ZS: Gewicht des Spannbaums ist zu minimieren.

Frage: Was ist eine optimale Lösung?

Wir werden uns im Folgenden mit einem recht einfachen Verfahren befassen, das die optimale Lösung zu dem Problem liefert.

Überblick über das Verfahren

Das Verfahren, das in diesem Abschnitt vorgestellt wird, geht auf Kruskal (1956) zurück. Es betrachtet in $|E|$ Schritten jeweils eine Kante und wählt diese für den

Spannbaum aus oder verwirft sie. Die Kanten werden in einer Reihenfolge nicht-absteigender Kantengewichte betrachtet. Voraussetzung für die Aufnahme der jeweils betrachteten Kante $\{i, j\}$ ist, dass es zwischen den Knoten i und j bisher noch keinen Weg aus gewählten Kanten gibt.

Salopp formuliert: Verfahren von Kruskal

Wir konstruieren einen Spannbaum minimalen Gewichts, indem wir Kanten auswählen, sofern sie keinen Kreis schließen. Dabei betrachten wir gemäß Zielfunktion attraktive Kanten zuerst.

Das gesamte Verfahren lässt sich als Algorithmus 5.3 konkretisieren.

Algorithmus 5.3 (Kruskal).

Eingabe: Graph $G = (V, E, w)$

Ausgabe: Spannbaum E' minimalen Gewichts oder Information, dass es keinen Spannbaum von G gibt

1. Wenn G nicht zusammenhängend ist, hat G keinen Spannbaum. STOP.
2. Sortiere Kanten nach Kantengewichten nicht-absteigend (und bezeichne sie gemäß dieser Sortierung mit $(e_1, \dots, e_{|E|})$).
3. $k \leftarrow 1$, $E' \leftarrow \emptyset$
4. Wenn $|E'| = |V| - 1$, dann ist E' ein Spannbaum minimalen Gewichts. STOP.
5. Wenn es keinen Weg zwischen den Knoten, die durch e_k verbunden werden, in (V, E') gibt, dann setze $E' \leftarrow E' \cup \{e_k\}$
6. $k \leftarrow k + 1$
7. Gehe zu 4.

Zunächst wird geprüft, ob der Graph zusammenhängend ist, denn ansonsten gibt es keinen Spannbaum. Danach werden die Kanten nach ihren Gewichten nicht-absteigend sortiert. Wir wählen nun $|V| - 1$ Kanten für den Spannbaum, indem wir die Kanten in dieser Reihenfolge betrachten und die aktuelle Kante aufnehmen, wenn sie keinen Kreis in (V, E') schließt. Da ein Graph ein Baum ist, wenn er kreisfrei ist und genau eine Kante weniger als Knoten hat, liegt bei Abbruch des Verfahrens in Schritt 4. ein Spannbaum vor.

Beispiel

Bevor wir das Verfahren analysieren, wollen wir es an einem Beispiel veranschaulichen. Wir wenden Algorithmus 5.3 auf die Problem Instanz in Abb. 5.13 an.

Tabelle 5.9 stellt den Ablauf des Verfahrens dar. Die jeweils ausgewählten und noch verfügbaren Kanten werden in Abb. 5.14 bis Abb. 5.16 illustriert.

In Tabelle 5.9 korrespondiert jede der nummerierten Zeilen zur Betrachtung einer Kante e_k für eine mögliche Aufnahme in E' . Es finden sich jeweils die Kante e_k , der aktuelle Stand der Menge E' , ein Verweis auf eine Abbildung der sich aus dem

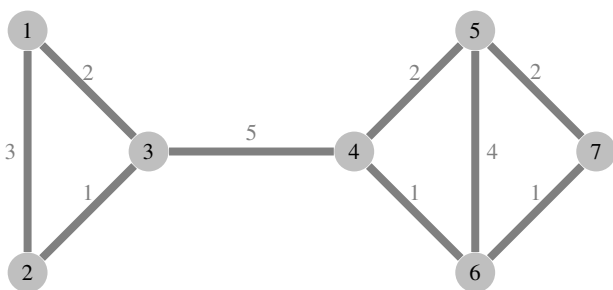


Abb. 5.13 Instanz von Problem 5.2 (Minimaler Spannbaum)

Tabelle 5.9 Ablauf von Algorithmus 5.3 für die Instanz in Abb. 5.13

	e_k	E'	Abb.	Entscheidung
1	$\{2,3\}$	\emptyset	5.13	$\{2,3\}$ aufnehmen
2	$\{4,6\}$	$\{\{2,3\}\}$	5.14	$\{4,6\}$ aufnehmen
3	$\{6,7\}$	$\{\{2,3\}, \{4,6\}\}$		$\{6,7\}$ aufnehmen
4	$\{1,3\}$	$\{\{2,3\}, \{4,6\}, \{6,7\}\}$		$\{1,3\}$ aufnehmen
5	$\{4,5\}$	$\{\{1,3\}, \{2,3\}, \{4,6\}, \{6,7\}\}$		$\{4,5\}$ aufnehmen
6	$\{5,7\}$	$\{\{1,3\}, \{2,3\}, \{4,5\}, \{4,6\}, \{6,7\}\}$	5.15	$\{5,7\}$ nicht aufnehmen
7	$\{1,2\}$	$\{\{1,3\}, \{2,3\}, \{4,5\}, \{4,6\}, \{6,7\}\}$		$\{1,2\}$ nicht aufnehmen
8	$\{5,6\}$	$\{\{1,3\}, \{2,3\}, \{4,5\}, \{4,6\}, \{6,7\}\}$		$\{5,6\}$ nicht aufnehmen
9	$\{3,4\}$	$\{\{1,3\}, \{2,3\}, \{4,5\}, \{4,6\}, \{6,7\}\}$		$\{3,4\}$ aufnehmen
		$\{\{1,3\}, \{2,3\}, \{3,4\}, \{4,5\}, \{4,6\}, \{6,7\}\}$	5.16	

Schritt ergebenden Situation. Die resultierende Menge E' findet sich dann in der jeweils folgenden Zeile.

Wir können der zweiten Spalte von Tabelle 5.9 die Sortierung der Kanten entnehmen. Dabei stellen wir fest, dass diese Sortierung nicht eindeutig ist, da es verschiedene Kanten mit identischen Gewichten gibt. Wir legen uns hier nicht fest, wie wir Kanten mit identischen Gewichten sortieren. Wir werden zwar sehen, dass wir abhängig von der Sortierung dieser Kanten unterschiedliche Spannäume erhalten können, werden aber in der Analyse erschließen, dass diese unterschiedlichen Spannäume gleiche (und zwar minimale) Gewichte haben.

Wir wollen nun nachvollziehen, was in den einzelnen Iterationen geschieht. Nachdem wir geprüft haben, dass der Graph zusammenhängend ist, sortieren wir zunächst die Kanten und initialisieren E' . In Iteration 1 betrachten wir dann $e_1 = \{2,3\}$, stellen fest, dass es keinen Weg zwischen den Knoten 2 und 3 in (V, E') gibt, und nehmen daher $\{2,3\}$ in E' auf. Die Kanten in E' sind in Abb. 5.14 (und den folgenden Abbildungen) schwarz hervorgehoben. Zusätzlich sind die Kanten $e_{k+1}, \dots, e_{|E|}$, also die, die noch nicht betrachtet wurden und über deren Aufnahme in E' daher noch nicht entschieden wurde, grau dargestellt.

In Iteration 2 betrachten wir dann $e_2 = \{4,6\}$, stellen fest, dass es keinen Weg zwischen den Knoten 4 und 6 in (V, E') gibt, und nehmen daher $\{4,6\}$ in E' auf.

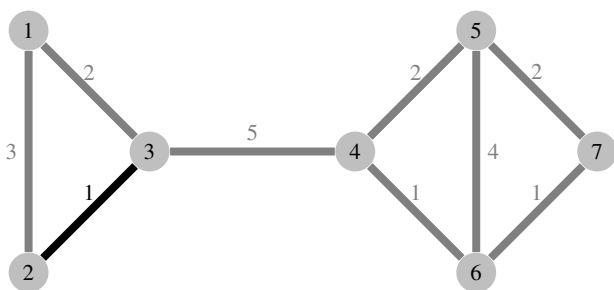


Abb. 5.14 Kanten in E' und noch verfügbare Kanten nach Iteration 1

Dasselbe passiert in den Iterationen 3, 4 und 5 mit den Kanten $\{6,7\}$, $\{1,3\}$ und $\{4,5\}$, sodass sich die Situation in Abb. 5.15 ergibt.

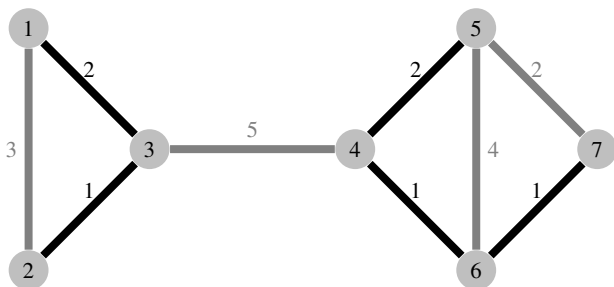


Abb. 5.15 Kanten in E' und noch verfügbare Kanten nach Iteration 5

In Iteration 6 betrachten wir dann $e_6 = \{5,7\}$, stellen fest, dass es den Weg $(5,4,6,7)$ von Knoten 5 nach Knoten 7 in (V, E') gibt, und nehmen daher $\{5,7\}$ nicht in E' auf. Wir können an dieser Stelle leicht erkennen, dass wir einen anderen Spannbaum erhalten würden, wenn die Kante $\{5,7\}$ vor Kante $\{4,5\}$ betrachtet würde.

In den Iterationen 7 und 8 betrachten wir dann $e_7 = \{1,2\}$ und $e_8 = \{5,6\}$, stellen fest, dass es die Wege $(1,3,2)$ und $(5,4,6)$ in (V, E') gibt, und nehmen daher weder $\{1,2\}$ noch $\{5,6\}$ in E' auf. Abschließend wird in Iteration 9 Kante $e_9 = \{3,4\}$ in E' aufgenommen, siehe Abb. 5.16.

Da wir nun $|E'| = |V| - 1 = 6$ Kanten in E' aufgenommen und dabei keinen Kreis geschlossen haben, ist E' ein Spannbaum. Dass E' unter allen Spannbäumen minimales Gewicht hat, können wir am Beispiel erkennen. Im Folgenden wollen wir das Verfahren allgemein analysieren.

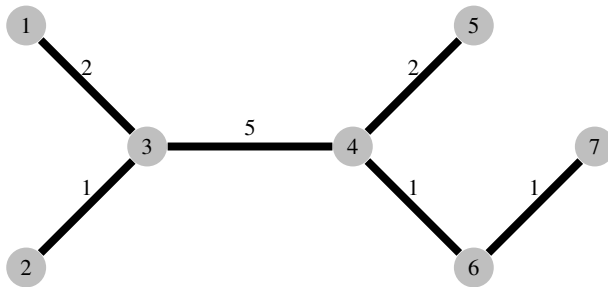


Abb. 5.16 Kanten in E' und noch verfügbare Kanten nach Iteration 9

Analyse

Nachdem wir gesehen haben, dass Algorithmus 5.3 die Probleminstanz in Abb. 5.13 löst, besagt Theorem 5.5, dass er dies tatsächlich für jede Instanz von Problem 5.2 (Minimaler Spannbaum) leistet. Den Beweis führen wir in Anhang C.4.

Theorem 5.5. *Algorithmus 5.3 findet einen Spannbaum minimalen Gewichts, wenn der Graph zusammenhängend ist.*

5.3 Matchings

In diesem Abschnitt soll es um die Bestimmung von Matchings gehen. Zunächst wollen wir spezifizieren, was Matchings überhaupt sind und welches Problem wir lösen wollen.

Definition 5.3 (Matching). Ein Matching in einem Graphen $G = (V, E)$ ist eine Teilmenge $M \subseteq E$ von Kanten, sodass in $G' = (V, M)$ jeder Knoten in V adjazent zu höchstens einem anderen Knoten ist.

Anders formuliert ist ein Matching eine Menge von zweielementigen Teilmengen von V , wobei kein Element von V in mehr als einer Teilmenge vorkommt. Wir sagen, dass Knoten $v \in V$ in Bezug auf ein Matching M in Graph $G = (V, E)$ *einsam* (*gepaart*) ist, wenn es keinen (einen) Knoten $v' \in V$ gibt, der zu v in $G' = (V, M)$ adjazent ist. Wir nennen außerdem v' den *Partner* von v , wenn $\{v, v'\} \in M$.

Zusammenfassung: Einsame und gepaarte Knoten in Matchings

Ein Matching ist eine Menge von Kanten, sodass kein Knoten durch diese Kanten mit mehr als einem anderen Knoten verbunden wird. Der Knoten, mit dem ein anderer Knoten verbunden wird, ist sein Partner. Ein Knoten mit Partner ist gepaart, ein Knoten ohne Partner ist einsam.

Abb. 5.17 stellt ein Matching dar. Die schwarzen Kanten sind im Matching enthalten, die grauen Kanten sind im Graphen vorhanden, aber nicht im Matching. Knoten 1, 2, 5 und 6 sind gepaart und Knoten 3 und 4 sind einsam.

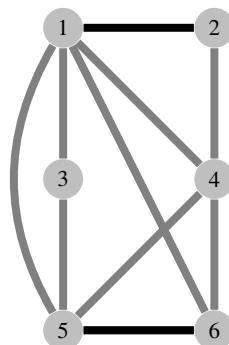


Abb. 5.17 Matching

Definition 5.4 (Perfektes Matching). Ein Matching M in einem Graphen $G = (V, E)$ ist perfekt, wenn $|M| = |V|/2$.

In einem perfekten Matching gibt es also für jeden Knoten in V genau einen Partner. Abb. 5.18 stellt ein perfektes Matching in demselben Graphen dar.

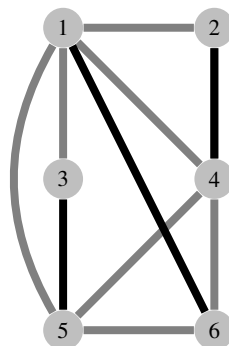


Abb. 5.18 Perfektes Matching

Definition 5.5 (Gewicht eines Matchings). Das Gewicht eines Matchings M in einem gewichteten Graphen $G = (V, E, w)$ ist die Summe der Gewichte der Kanten in M .

Das Gewicht des perfekten Matchings in Abb. 5.19 ist $5 + 6 + 7 = 18$. Wir werden uns in den Abschnitten 5.3.1 und 5.3.2 mit der Lösung der folgenden beiden Probleme 5.3 und 5.4 befassen.

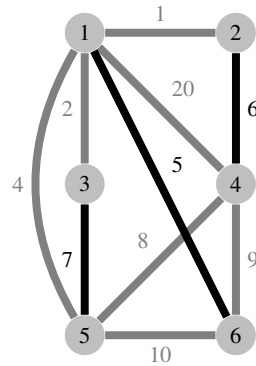


Abb. 5.19 Perfektes Matching mit Gewichten

Problem 5.3 (Matching maximaler Kardinalität).

Eingabe: (Ungerichteter) Graph $G = (V, E)$

LR: Teilmengen von E

ZB: Die gewählte Teilmenge von Kanten muss ein Matching sein.

ZS: Anzahl der Kanten im Matching ist zu maximieren.

Frage: Was ist eine optimale Lösung?

Wenn es für Graph G ein perfektes Matching gibt, dann ist ein solches Matching auch ein Matching maximaler Kardinalität, denn ein Matching kann nicht mehr als $|V|/2$ Kanten enthalten. Selbst wenn wir wissen, dass es ein perfektes Matching gibt, ist nicht offensichtlich, wie man eins findet. Außerdem hat nicht jeder Graph ein perfektes Matching und dann ist weder offensichtlich, wie viele Kanten ein Matching maximaler Kardinalität hat, noch, wie man ein solches Matching findet.

Problem 5.4 (Matching maximalen Gewichts).

Eingabe: (Ungerichteter) Gewichteter Graph $G = (V, E, w)$

LR: Teilmengen von E

ZB: Die gewählte Teilmenge von Kanten muss ein Matching sein.

ZS: Das Gewicht des Matchings ist zu maximieren.

Frage: Was ist eine optimale Lösung?

Problem 5.4 (Matching maximalen Gewichts) ist eine Verallgemeinerung von Problem 5.3 (Matching maximaler Kardinalität), denn wenn alle Kanten das gleiche Gewicht haben, dann entspricht ein Matching maximalen Gewichts einem Matching maximaler Kardinalität. Wenn die Kanten sich in ihren Gewichten unterscheiden, muss ein Matching maximaler Kardinalität aber nicht zwangsläufig ein Matching maximalen Gewichts sein. Das Gewicht des perfekten Matchings in Abb. 5.19 ist $5 + 6 + 7 = 18$, allerdings hat das Matching $\{\{1, 4\}, \{5, 6\}\}$ mit maximalem Gewicht $20 + 10 = 30$ lediglich zwei Kanten.

Es gibt noch weitere häufig betrachtete Problemvarianten im Zusammenhang mit Matchings. Beispielsweise wird häufig nach einem perfekten Matching minimalen Gewichts gefragt. Viele der Varianten lassen sich aber mit Techniken lösen, die denen, die wir im Folgenden vorstellen, ähnlich sind.

Zentral für die Verfahren in den Abschnitten 5.3.1 und 5.3.2 ist das Konzept eines vergrößernden Pfades (im Englischen „Augmenting Path“). Ein vergrößernder Pfad hilft uns, ein bestehendes Matching M in einem Graphen $G = (V, E)$ zu vergrößern, d. h., ein anderes Matching mit größerer Kardinalität zu finden.

Definition 5.6 (Vergrößernder Pfad). Ein vergrößernder Pfad für Graph $G = (V, E)$ und Matching M in G ist ein Pfad in $G = (V, E)$, der an einem einsamen Knoten startet, der an einem einsamen Knoten endet und dessen Kanten abwechselnd in M und in $E \setminus M$ sind.

Es sollte betont werden, dass eine einzelne Kante in $E \setminus M$, die zwei einsame Knoten verbindet, auch schon einen vergrößernden Pfad für G und M darstellt. Aus Definition 5.6 folgt unmittelbar, dass ein vergrößernder Pfad eine ungerade Anzahl an Kanten hat. Das folgende Ergebnis von Berge (1957) wird sich im Weiteren als sehr nützlich erweisen.

Theorem 5.6. *Ein Matching M in einem Graphen $G = (V, E)$ hat genau dann maximale Kardinalität, wenn es keinen vergrößernden Pfad für G und M gibt.*

Der formale Beweis findet sich in Anhang C.5. Hier wollen wir den Fokus darauf legen, wie wir dieses Ergebnis nutzen können. Wir können erkennen, dass wir ein Matching M' mit Kardinalität $|M'| = |M| + 1$ erzeugen können, indem wir alle Kanten im Pfad, die bisher nicht in M sind, in das Matching aufnehmen und alle Kanten im Pfad, die in M sind, aus dem Matching entfernen. Alle Knoten in V bis auf Start- und Endknoten des vergrößernden Pfades haben dieselbe Anzahl inzidenter Kanten in M' wie in M . Start- und Endknoten des vergrößernden Pfades waren einsam in M und sind gepaart in M' . Wir werden im Folgenden sagen, dass wir M modifizieren, indem wir den Pfad auf M anwenden.

Zumindest Problem 5.3 (Matching maximaler Kardinalität) können wir nun in einem iterativen Schema lösen.

Algorithmus 5.4 (Matching maximaler Kardinalität).

Eingabe: Graph $G = (V, E)$

Ausgabe: Matching M in G mit maximaler Kardinalität

1. Setze $M \leftarrow \emptyset$ (oder verwende beliebiges bekanntes Matching M).
2. Prüfe, ob es einen vergrößernden Pfad P für G und M gibt. Wenn es keinen gibt, hat M maximale Kardinalität. STOP.
3. Wende P auf M an.
4. Gehe zu 2.

Wir starten mit einem beliebigen Matching für Graph G , z. B. mit $M = \emptyset$. Wir prüfen, ob es einen vergrößernden Pfad für G und M gibt, und ermitteln einen solchen, falls es einen gibt. Nun modifizieren wir M und vergrößern dadurch seine Kardinalität um eins. Dieser Schritt wird wiederholt, bis es keinen vergrößernden Pfad

für G und M gibt. Das dann vorliegende Matching hat maximale Kardinalität. Die Korrektheit dieses Verfahrens folgt direkt aus Theorem 5.6, denn Algorithmus 5.4 endet erst, wenn es keinen vergrößernden Pfad für G und das vorliegende Matching gibt, und dann ist das vorliegende Matching gemäß Theorem 5.6 maximal.

Salopp formuliert: Matching maximaler Kardinalität durch Pfade

Ausgehend von einem beliebigen Matching erhöhen wir die Kardinalität des vorliegenden Matchings in jedem Schritt um eins, indem wir einen vergrößernden Pfad ermitteln und die Kanten, die sowohl im Pfad als auch im Matching sind, durch die Kanten, die im Pfad, aber nicht im Matching sind, ersetzen. Wenn es keinen vergrößernden Pfad gibt, hat das vorliegende Matching maximale Kardinalität.

In Abb. 5.20 bis 5.23 wird ein Beispiel dargestellt. Abb. 5.20 bildet den zugrunde liegenden Graphen $G = (V, E)$ und ein erstes Matching M in G ab. Wie schon zuvor sind dabei Kanten in $E \setminus M$ grau und Kanten in M schwarz dargestellt.

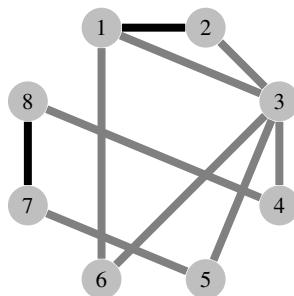


Abb. 5.20 Instanz von Problem 5.3 (Matching maximaler Kardinalität) und Matching M

Es gibt mehrere vergrößernde Pfade für G und M . Wir betrachten einen sehr einfachen, der nur aus der Kante $\{3, 5\}$ besteht. Das resultierende Matching M' mit drei Kanten sehen wir in Abb. 5.21.

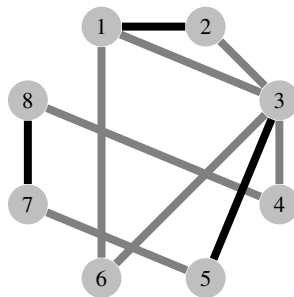


Abb. 5.21 Zweites Matching M' zur Instanz in Abb. 5.20

Wir können keinen vergrößernden Pfad für G und M' , der nur aus einer Kante besteht, finden. Ein längerer vergrößernder Pfad existiert aber und wird in Abb. 5.22 dargestellt. Durch Verwendung dieses Pfades erhalten wir Matching M'' mit vier Kanten, das in Abb. 5.23 abgebildet ist. Matching M'' ist ein perfektes Matching und somit ein Matching maximaler Kardinalität. Es existiert kein vergrößernder Pfad für G und M'' .

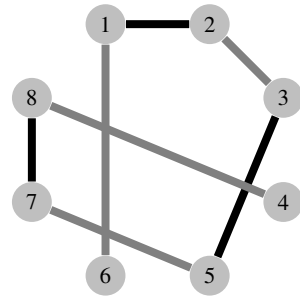


Abb. 5.22 Vergrößernder Pfad für G und M'

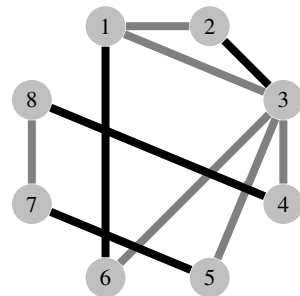


Abb. 5.23 Drittes Matching M'' zur Instanz in Abb. 5.20

Offen ist an dieser Stelle, wie wir prüfen, ob es einen vergrößernden Pfad für G und M gibt, und diesen ggf. finden. In dem obigen Beispiel ist dies aufgrund der Größe der Instanz „mit genauem Hinschauen“ machbar, für größere Instanzen brauchen wir aber eine systematische Vorgehensweise. Offen ist darüber hinaus, ob und wie wir dieses Verfahren adaptieren können, um Problem 5.4 (Matching maximalen Gewichts) zu lösen. Darum soll es im Folgenden gehen.

Wir werden uns in Abschnitt 5.3.1 zunächst mit dem Spezialfall eines bipartiten Graphen beschäftigen und diese Ansätze dann in Abschnitt 5.3.2 für beliebige Graphen verallgemeinern. Die Argumentationen orientieren sich dabei zu weiten Teilen an denen in Galil (1986). Die Verfahren gehen auf Edmonds (1965a), Edmonds (1965b) und Ford Jr. und Fulkerson (1956) zurück.

5.3.1 Bipartite Graphen

Matchings in bipartiten Graphen (siehe Definition 2.11 auf S. 26) spielen häufig eine Rolle, wenn es darum geht, Paare aus Elementen zweier Mengen zu bilden. Ein klassisches Anwendungsbeispiel ist die Zuordnung von Personal zu Aufgaben. Hierbei gibt es für jede Person einen Knoten in der ersten Menge V von Knoten und für jede Aufgabe einen Knoten in der zweiten Menge U von Knoten. Es gibt eine Kante zwischen Knoten $v \in V$ und $u \in U$, wenn Person v Aufgabe u übernehmen kann. Ein Matching maximaler Kardinalität würde dann eine maximale Anzahl übernommener Aufgaben darstellen, wobei jede Person eine Aufgabe übernimmt, die sie auch beherrscht. Obwohl wir ungerichtete Graphen behandeln, werden wir in diesem Abschnitt eine Kante nicht als zweielementige Menge, sondern als Paar (v, u) mit $v \in V$ und $u \in U$ darstellen. Dies ist dadurch gerechtfertigt, dass die beiden adjazenten Knoten bereits durch die Zugehörigkeit zu den verschiedenen Mengen verschiedene Rollen haben.

In einem gewichteten Graph könnte das Gewicht der Kante (v, u) den Profit darstellen, der mit der Erledigung von Aufgabe u durch Person v einhergeht. Dieser Profit mag von Person und Aufgabe abhängen, da unterschiedliche Personen dieselbe Aufgabe womöglich unterschiedlich gut oder unterschiedlich effizient erledigen und unterschiedliche Aufgaben unterschiedlichen Aufwand bedeuten. Ein Matching mit maximalem Gewicht stellt dann eine Auswahl von Aufgaben und deren Zuordnung zu Personen dar, sodass der erzielte Profit maximal ist.

Abgesehen davon, dass der Spezialfall eines bipartiten Graphen an sich wichtige Anwendungsbereiche hat, wird uns die Analyse der Verfahren in den Abschnitten 5.3.1.1 und 5.3.1.2 helfen, die Verfahren in Abschnitt 5.3.2 zu verstehen. Wir werden im Folgenden zunächst Problem 5.3 (Matching maximaler Kardinalität) betrachten und uns im Anschluss Problem 5.4 (Matching maximalen Gewichts) zuwenden.

5.3.1.1 Matchings maximaler Kardinalität

Überblick über das Verfahren

Algorithmus 5.4 gibt ein Gerüst vor, dessen wir uns bedienen wollen. Allerdings müssen wir noch eine systematische Vorgehensweise zur Ermittlung von vergrößerten Pfaden ergänzen. In bipartiten Graphen liegen die beiden einsamen Start- und Endknoten eines vergrößerten Pfades nicht in derselben Teilmenge von Knoten. Wir werden uns im Folgenden vorstellen, dass wir vergrößernde Pfade suchen, indem wir von einsamen Knoten in V ausgehen und versuchen, durch schrittweises Ergänzen weiterer Kanten einen Pfad zu einem einsamen Knoten in U zu finden. Wir markieren dabei Knoten, die wir schon erreicht haben und die dementsprechend als Ausgangspunkt für weitere anzufügende Kanten dienen können. Initial werden die einsamen Knoten in V markiert, da diese schon zu Beginn des Verfahrens als Ausgangspunkt für anzufügende Kanten fungieren können. Erreichte Knoten in V

bzw. U bezeichnen wir als *ungerade* bzw. *gerade* Knoten entsprechend einer Nummerierung entlang der gefundenen Wege ausgehend von einem einsamen Knoten in V .

Zusammenfassung: Ungerade und gerade Knoten in (vergrößernden) Pfaden

Für ein vorliegendes Matching M konstruieren wir Pfade, die an einem einsamen Knoten in V starten und abwechselnd Kanten in $E \setminus M$ und in M enthalten. Dafür beginnen wir bei einsamen Knoten in V und verlängern die Pfade schrittweise durch geeignete Kanten und markieren dabei Knoten auf Pfaden in V bzw. U als ungerade bzw. gerade.

Algorithmus 5.5 (Vergrößernde Pfade in bipartiten Graphen).

Eingabe: Graph $G = (V, U, E)$, Matching M

Ausgabe: Vergrößernder Pfad für G und M oder Information, dass keiner existiert

1. Markiere alle einsamen Knoten in V als ungerade.
2. Wenn es einen ungeraden Knoten $v \in V$ und einen nicht-markierten Knoten $u \in U$ mit $(v, u) \in E \setminus M$ gibt, dann
 - a. markiere u und merke dir v als Vorgänger von u und
 - b. wenn u einsam ist, gehe zu Schritt 5., sonst wiederhole Schritt 2.
3. Wenn es einen geraden Knoten $u \in U$ und einen nicht-markierten Knoten $v \in V$ mit $(v, u) \in M$ gibt, dann markiere v . Gehe zu Schritt 2.
4. Es gibt keinen vergrößernden Pfad. STOP.
5. Gehe von dem zuletzt erreichten geraden Knoten aus und rekonstruiere den vergrößernden Pfad. STOP.

Das Vorgehen in Algorithmus 5.5 ist recht einfach. Wir prüfen wiederholt, ob wir in Schritt 2. oder Schritt 3. einen weiteren Knoten markieren können. Dies geschieht in Schritt 2., wenn wir von einem ungeraden Knoten v aus eine Kante, die nicht im Matching, aber in E ist, zu einem unmarkierten Knoten $u \in U$ an einen bestehenden Pfad anhängen können. Sollte der Knoten u gepaart sein, dann dient u in weiteren Iterationen als potentieller Ausgangspunkt für Verlängerungen des Pfades und wir merken uns den Knoten $v \in V$, von dem aus wir u erreicht haben, um später den Pfad rekonstruieren zu können. Sollte u einsam sein, dann haben wir einen vergrößernden Pfad gefunden, denn alle konstruierten Pfade starten an einem einsamen Knoten in V und bestehen alternierend aus Kanten in $E \setminus M$ und M . Wir rekonstruieren diesen Pfad in Schritt 5. und das Verfahren endet. In Schritt 3. markieren wir einen weiteren Knoten, wenn wir von einem geraden Knoten u aus eine Kante, die im Matching ist, zu einem unmarkierten Knoten $v \in V$ an einen bestehenden Pfad anhängen können. In diesem Fall ist klar, über welchen Knoten v erreicht wurde, denn v hat in M nur einen Partner. Nur wenn beide Schritte zu keiner neuen Markierung führen, wird Schritt 4. erreicht. In diesem Fall gibt es keinen vergrößernden Pfad. Dies ist nicht offensichtlich und wir sollten dafür im Folgenden noch argumentieren.

Beispiel

Zunächst wollen wir uns allerdings ein Beispiel anschauen. In den Abb. 5.24 bis 5.31 wird dargestellt, wie wir in zwei Schritten jeweils Algorithmus 5.5 anwenden, um ein gegebenes Matching um eine Kante zu vergrößern. Abb. 5.24 stellt den vollständigen Graphen $G = (V, U, E)$ mit den schwarz hervorgehobenen Kanten des gegebenen Matchings M dar.

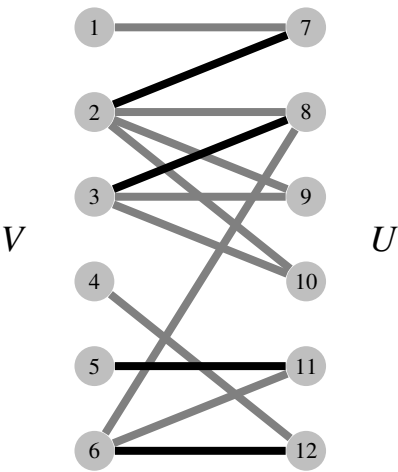


Abb. 5.24 Graph G und Matching M

Wir wenden nun Algorithmus 5.5 an und markieren zunächst die einsamen Knoten 1 und 4 in Schritt 1. Wir erhalten die Markierungen, wie in Abb. 5.25 dargestellt.

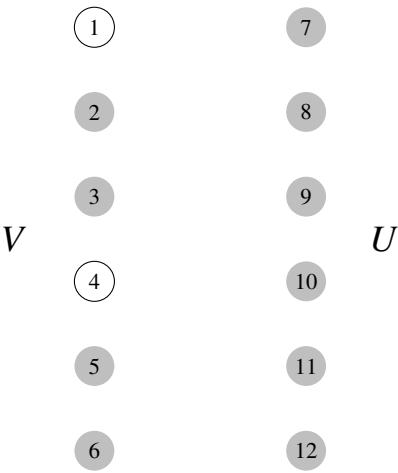


Abb. 5.25 Markierungen der Knoten nach Schritt 1.

In Abb. 5.25 bzw. in allen folgenden Abbildungen mit Pfaden sind markierte Knoten mit einem schwarzen Rand dargestellt. Dabei sind ungerade Knoten weiß und gerade Knoten grau. Alle übrigen, also die unmarkierten, Knoten sind ohne Rand und in grau dargestellt.

Wir gehen jetzt iterativ vor und wenden die Schritte 2. und 3. in jeder Iteration für alle Kanten an, für die dies zu Beginn der Iteration möglich ist (eine Alternative wäre, die Schritte immer nur für eine Kante anzuwenden und danach die nächste Iteration zu beginnen). Wir wenden dann zunächst zweimal Schritt 2. an und markieren die gepaarten Knoten 7 und 12. In dieser Iteration können wir Schritt 3. nicht anwenden und erhalten daher Pfade und Markierungen wie in Abb. 5.26 dargestellt.

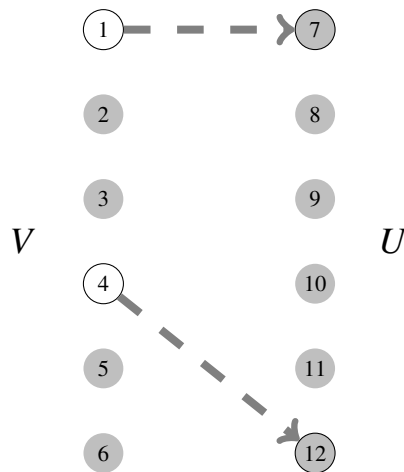


Abb. 5.26 Pfade und Markierungen zu Matching M in Abb. 5.24 nach der ersten Iteration

In Abb. 5.26 und in allen folgenden Abbildungen mit Pfaden sind Kanten, die in einem gefundenen Pfad, aber nicht im vorliegenden Matching sind, gestrichelt dargestellt. Die angedeutete Orientierung der Kanten findet sich nicht in dem Graph, sondern soll die Funktion im Pfad veranschaulichen. Die Kante ist von dem zuerst markierten Knoten zu dem von ihm aus erreichten Knoten orientiert.

In der nächsten Iteration können wir Schritt 2. nicht anwenden, da alle markierten Knoten in V schon betrachtet wurden und die entsprechenden Kanten an die Pfade angehängen wurden. Durch Anwendung von Schritt 3. können wir dann aber die Knoten 2 und 6 markieren. Wir erhalten Pfade und Markierungen wie in Abb. 5.27 dargestellt.

In der dritten Iteration markieren wir durch Schritt 2. die Knoten 8, 9, 10 und 11. Knoten 8 wird hierbei zweimal von einem Pfad erreicht und zwar von den Knoten 2 und 6 aus. Für das Finden eines vergrößerten Pfades (sofern einer existiert) ist hierbei unerheblich, welchen Vorgänger wir uns merken. Wenn es einen vergrößerten Pfad gibt, der über Knoten 8 läuft, dann gibt es mindestens zwei Pfade (der Teil des Pfades, der bei Knoten 8 beginnt, kann um zwei Anfänge, nämlich die beiden Teilpfade zu Knoten 8, ergänzt werden). Da aber gleichzeitig der gerade Knoten 9

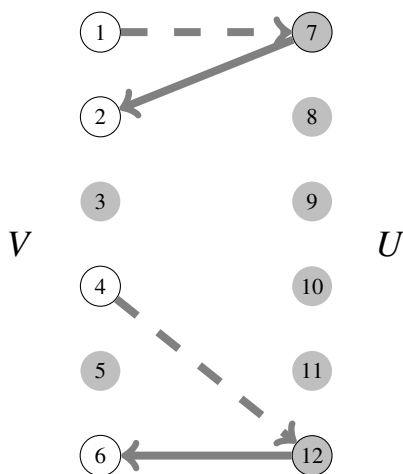


Abb. 5.27 Pfade und Markierungen zu Matching M in Abb. 5.24 nach der zweiten Iteration

erreicht wird, der einsam ist, haben wir schon einen vergrößernden Pfad von Knoten 1, über die Knoten 7 und 2 zu Knoten 9 gefunden (und gehen durch Schritt 2b zu Schritt 5). Simultan wird der einsame, gerade Knoten 10 erreicht, sodass ein zweiter vergrößernder Pfad ermittelt wird. Wir können einen dieser vergrößernden Pfade auswählen und entscheiden uns willkürlich dafür, den ersten anzuwenden. Schritt 3 entfällt somit. Abb. 5.28 stellt die gefundenen Pfade und Abb. 5.29 das vergrößerte Matching M' dar.

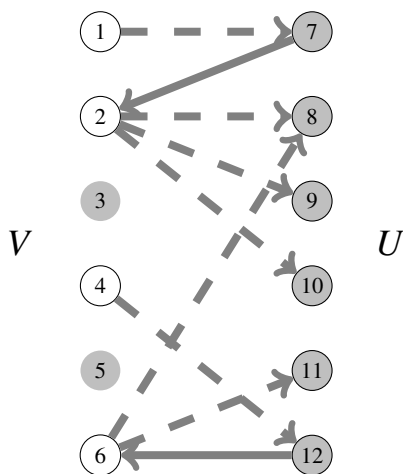


Abb. 5.28 Pfade zu Matching M in Abb. 5.24

Wir wenden erneut Algorithmus 5.5 an und markieren den einsamen Knoten 4 in Schritt 1. In der ersten folgenden Iteration führen wir Schritt 2. aus und markieren Knoten 12. In der zweiten, dritten und vierten Iteration markieren wir zunächst in Schritt 3. Knoten 6, dann in Schritt 2. die Knoten 8 und 11 und schließlich in Schritt

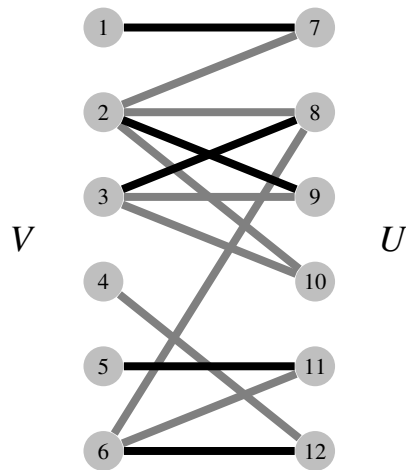


Abb. 5.29 Zweites Matching M' zur Instanz in Abb. 5.20

3. die Knoten 3 und 5. Wir sehen nun, dass wir von Knoten 5 aus den Pfad nicht verlängern können, denn Knoten 5 liegt an keiner Kante, die nicht in Matching M' ist. Allerdings können wir von Knoten 3 ausgehend in Schritt 2. die Knoten 9 und 10 markieren. Da Knoten 10 einsam ist, haben wir einen vergrößerten Pfad von Knoten 4 über die Knoten 12, 6, 8 und 3 zu Knoten 10 gefunden. Abb. 5.30 stellt die gefundenen Pfade und Abb. 5.31 das vergrößerte Matching M'' dar.

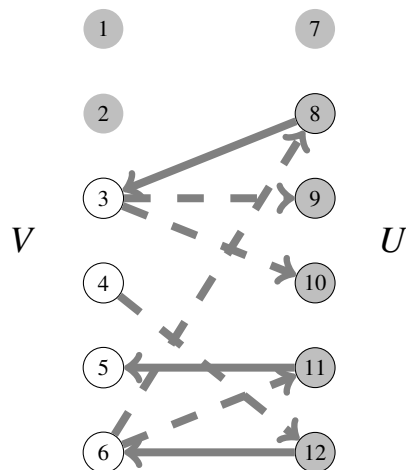


Abb. 5.30 Pfade zu Matching M' in Abb. 5.29

Wenn wir nun Algorithmus 5.5 wieder anwenden, um zu prüfen, ob es einen weiteren vergrößerten Pfad gibt, dann werden wir feststellen, dass dies nicht der Fall ist. Wir können in Schritt 1. keinen Knoten markieren und somit keine Pfade verlängern. Es ist grundsätzlich auch möglich, dass in Schritt 1. Knoten markiert werden, aber alle konstruierten Pfade nicht bei einem einsamen Knoten in U enden.

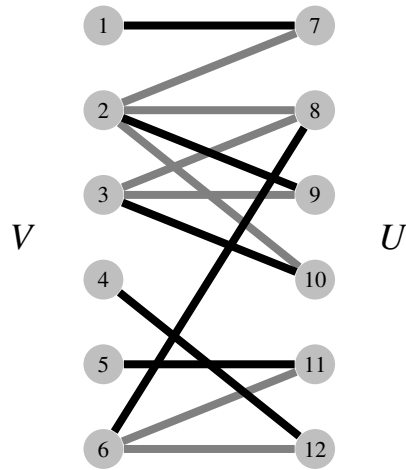


Abb. 5.31 Drittes Matching M'' zur Instanz in Abb. 5.20

Analyse

Nachdem wir gesehen haben, dass wir die Problem Instanz in Abb. 5.24 optimal lösen können, wollen wir uns nun überlegen, dass dies kein Zufall war.

Theorem 5.7. *Wenn Algorithmus 5.5 keinen vergrößernden Pfad für Graph G und Matching M findet, dann gibt es keinen.*

Den Beweis führen wir in Anhang C.6. Theorem 5.7 bringt uns Gewissheit, dass Algorithmus 5.4 ein Matching maximaler Kardinalität liefert, wenn wir Algorithmus 5.5 in Schritt 2. verwenden, um zu prüfen, ob es einen vergrößernden Pfad für den vorliegenden Graph und das aktuelle Matching gibt, und ggf. einen zu finden.

Zusammenfassung: Matchings maximaler Kardinalität in bip. Graphen

Wir können Matchings maximaler Kardinalität in einem bipartiten Graphen G bestimmen, indem wir von einem beliebigen Matching ausgehend in jedem Schritt mittels Algorithmus 5.5 prüfen, ob es einen vergrößernden Pfad für G und das vorliegende Matching M gibt. Falls Algorithmus 5.5 einen vergrößernden Pfad findet, wenden wir ihn auf M an und wiederholen den Schritt. Wenn Algorithmus 5.5 keinen vergrößernden Pfad für G und M findet, dann hat M maximale Kardinalität.

5.3.1.2 Matchings maximalen Gewichts

Nun wollen wir uns Problem 5.4 (Matching maximalen Gewichts) zuwenden und repräsentieren dieses Problem zunächst für bipartite Graphen in dem folgenden MO.

Modell 5.1 (Matching maximalen Gewichts in bipartiten Graphen (Problem 5.4 für bipartite Graphen)).

$$\max \sum_{v \in V} \sum_{u \in U, (v,u) \in E} w_{v,u} \cdot x_{v,u} \quad (5.1)$$

$$u.B.d.N. \quad \sum_{u \in U, (v,u) \in E} x_{v,u} \leq 1 \quad \forall v \in V \quad (5.2)$$

$$\sum_{v \in V, (v,u) \in E} x_{v,u} \leq 1 \quad \forall u \in U \quad (5.3)$$

$$x_{v,u} \geq 0 \quad \forall v \in V; u \in U, (v,u) \in E \quad (5.4)$$

Die Variable $x_{v,u}$ signalisiert, ob die Kante (v,u) im Matching ist ($x_{v,u} = 1$) oder nicht ($x_{v,u} = 0$). Obwohl sie nicht auf binäre Werte beschränkt ist, gibt es immer optimale Lösungen, in denen diese Variable binäre Werte annimmt. Wir haben diesen Effekt bereits in Abschnitt 2.3.2.4 für das ähnliche Modell 2.17 diskutiert. Die NB (5.2) und (5.3) stellen sicher, dass eine Lösung einem Matching entspricht, während die Zielfunktion (5.1) das Ziel der Gewichtsmaximierung repräsentiert.

Mit den aus Abschnitt 3.4 bekannten Mitteln leiten wir das zu Modell 5.1 duale MO ab.

Modell 5.2 (Duales Modell zu Modell 5.1).

$$\min \sum_{v \in V} \alpha_v + \sum_{u \in U} \beta_u \quad (5.5)$$

$$u.B.d.N. \quad \alpha_v + \beta_u \geq w_{v,u} \quad \forall v \in V; u \in U, (v,u) \in E \quad (5.6)$$

$$\alpha_v \geq 0 \quad \forall v \in V \quad (5.7)$$

$$\beta_u \geq 0 \quad \forall u \in U \quad (5.8)$$

Die Variablen α_v , $v \in V$, und β_u , $u \in U$, sind die Dualvariablen, die zu den NB (5.2) und (5.3) von Modell 5.1 korrespondieren. Die Struktur von Modell 5.2 ergibt sich dann, wie in Abschnitt 3.4 diskutiert.

Die Idee des folgenden Verfahrens ist, eine primale zulässige Lösung (zu Modell 5.1) und eine duale zulässige Lösung (zu Modell 5.2) schrittweise so zu modifizieren, dass sie abschließend beide (für das jeweilige Modell) optimal sind. Wir wollen uns noch einmal Theorem 3.10 vor Augen führen.

Theorem 3.10 (Komplementärer Schlupf). Eine zulässige Lösung \mathbf{x} des primalen LP 3.8 und eine zulässige Lösung ω des dualen LP 3.19 sind genau dann optimal, wenn

$$x_j \cdot \left(\sum_{i=1}^m a_{i,j} \cdot \omega_i - c_j \right) = 0 \quad \text{für jedes } j = 1, \dots, n \quad \text{und}$$

$$\omega_i \cdot \left(b_i - \sum_{j=1}^n a_{i,j} \cdot x_j \right) = 0 \quad \text{für jedes } i = 1, \dots, m.$$

Wenn wir dies auf die Modelle 5.1 und 5.2 übertragen, ergibt sich, dass zwei jeweils zulässige Lösungen \mathbf{x} und (α, β) genau dann optimal sind, wenn

$$x_{v,u} \cdot (\alpha_v + \beta_u - w_{v,u}) = 0 \quad \text{für jedes } (v, u) \in E,$$

$$\alpha_v \cdot \left(1 - \sum_{u \in U, (v,u) \in E} x_{v,u} \right) = 0 \quad \text{für jedes } v \in V \quad \text{und}$$

$$\beta_u \cdot \left(1 - \sum_{v \in V, (v,u) \in E} x_{v,u} \right) = 0 \quad \text{für jedes } u \in U.$$

Wir können dies verbal(er) fassen. Ein Matching (die primale Lösung) und (α, β) (die duale Lösung) sind genau dann optimal, wenn

1. für jede Kante (v, u) im Matching $\alpha_v + \beta_u = w_{v,u}$ gilt,
2. für jeden einsamen Knoten $v \in V$ $\alpha_v = 0$ gilt und
3. für jeden einsamen Knoten $u \in U$ $\beta_u = 0$ gilt.

Wir bezeichnen im Folgenden eine Kante (v, u) als *verfügbar* (für Matchings), wenn $\alpha_v + \beta_u = w_{v,u}$. Im Laufe des Verfahrens werden wir verschiedene duale Lösungen betrachten, die unterschiedliche Mengen von verfügbaren Kanten implizieren. Während des Verfahrens werden immer nur Matchings aus aktuell verfügbaren Kanten konstruiert, wodurch Bedingung 1. eingehalten wird.

Zusammenfassung: Matchings maximalen Gewichts in bipartiten Graphen

Wir ermitteln ein Matching maximalen Gewichts in einem bipartiten Graphen, indem wir optimale Lösungen für die zueinander dualen Modelle 5.1 und 5.2 ermitteln. Optimalität der vorliegenden Lösungen prüfen wir mittels der Bedingungen 1. bis 3. Die erste dieser Bedingungen werden wir laufend einhalten, indem wir eine Kante (v, u) nur dann in Matchings verwenden, wenn sie verfügbar ist, d. h. wenn $\alpha_v + \beta_u = w_{v,u}$ gilt.

Überblick über das Verfahren

Das Verfahren startet mit einer primalen zulässigen Lösung und einer dualen zulässigen Lösung, die Bedingungen 1. und 3. erfüllen. Wir werden dann die Lösungen so verändern, dass nach jedem Schritt die Bedingungen 1. und 3. nach wie vor erfüllt sind und früher oder später auch Bedingung 2. erfüllt wird. Wir starten dabei mit einem leeren Matching $M = \emptyset$ sowie $\alpha_v = \max\{w_{v',u'} \mid (v',u') \in E\}$ für jedes $v \in V$ und $\beta_u = 0$ für jedes $u \in U$. Wir können recht leicht erkennen, dass beide Lösungen zulässig sind, denn jedes Matching ist primal zulässig und wir haben duale Zulässigkeit wegen $\alpha_v + \beta_u = \max\{w_{v',u'} \mid (v',u') \in E\} \geq w_{v,u}$ für jede Kante (v,u) . Zudem sind offensichtlich 1. und 3. erfüllt.

Wir wenden Algorithmus 5.5 an, beschränken uns dabei aber auf verfügbare Kanten. Wir prüfen also, ob es einen vergrößernden Pfad für $G' = (V, E')$ und M gibt, wobei E' die Menge der verfügbaren Kanten ist. Da wir Matchings immer nur aus verfügbaren Kanten konstruieren, gilt $M \subseteq E'$.

- Wenn wir einen vergrößernden Pfad finden, setzen wir ihn um und erhalten Matching M' , d. h., wir modifizieren die primale Lösung. Dabei bleibt zum einen Bedingung 1. erfüllt, denn $M' \subseteq E'$, d. h. das Matching besteht nur aus verfügbaren Kanten. Zum anderen bleibt jeder Knoten, der in M gepaart war auch in M' weiterhin gepaart. Somit kann auch Bedingung 3. nicht verletzt werden.
- Wenn wir keinen vergrößernden Pfad finden, modifizieren wir die duale Lösung wie folgt. Wir bestimmen $\delta = \min\{\delta^V, \delta^E\}$ mit

$$\delta^V = \min\{\alpha_v \mid v \in V, v \text{ ist ungerade}\} \text{ und}$$

$$\delta^E = \min\{\alpha_v + \beta_u - w_{v,u} \mid (v,u) \in E \setminus E', v \text{ ist ungerade}, u \text{ ist nicht markiert}\}.$$

Der Wert von δ^E stellt hier auf solche Kanten ab, die nicht in E' sind, aber zur Verlängerung von Pfaden genutzt werden könnten, wenn sie in E' wären. Nun verringern wir α_v für jeden ungeraden Knoten $v \in V$ um δ und erhöhen β_u für jeden geraden Knoten $u \in U$ um δ . Alle übrigen Dualvariablen behalten ihren bisherigen Wert.

Sollte nach Durchführung dieses Schrittes auch Bedingung 2. erfüllt sein, endet das Verfahren mit der optimalen Lösung. Sollte dies nicht der Fall sein, dann führt die Modifikation der dualen Lösung zu neuen verfügbaren Kanten und der Schritt wird wiederholt. Das Verfahren lässt sich als Algorithmus 5.6 konkretisieren.

Algorithmus 5.6 (Matching maximalen Gewichts in bipartiten Graphen).*Eingabe: Bipartiter Graph $G = (V, U, E, w)$* *Ausgabe: Matching maximalen Gewichts*

1. Setze $M \leftarrow \emptyset$, $\alpha_v \leftarrow \max\{w_{v',u'} \mid (v',u') \in E\}$ für jedes $v \in V$, $\beta_u = 0$ für jedes $u \in U$ und Menge $E' \leftarrow \{(v,u) \mid (v,u) \in E, \alpha_v + \beta_u = w_{v,u}\}$ der verfügbaren Kanten
2. Prüfe mit Algorithmus 5.5, ob ein vergrößernder Pfad für $G' = (V, E')$ und M existiert.
 - Falls einer existiert, finde einen und wende ihn auf M an.
 - Falls keiner existiert,
 - $\delta^V \leftarrow \min\{\alpha_v \mid v \in V, v \text{ ist ungerade}\}$ ($\delta^V \leftarrow \infty$, wenn es keinen ungeraden Knoten v gibt)
 - $\delta^E = \min\{\alpha_v + \beta_u - w_{v,u} \mid (v,u) \in E \setminus E', v \text{ ist ungerade}, u \text{ ist nicht markiert}\}$ ($\delta^E \leftarrow \infty$, wenn es keine solche Kante gibt)
 - $\delta \leftarrow \min\{\delta^V, \delta^E\}$
 - $\alpha_v \leftarrow \alpha_v - \delta$ für jeden ungeraden Knoten $v \in V$
 - $\beta_u \leftarrow \beta_u + \delta$ für jeden geraden Knoten $u \in U$
 - aktualisiere E'
3. Falls $\alpha_v = 0$ für jeden einsamen Knoten $v \in V$, STOP.
4. Gehe zu 2.

Zusammenfassung: Matchings maximalen Gewichts in bipartiten Graphen

Wir gehen von einem leeren Matching aus und versuchen einen vergrößernden Pfad aus verfügbaren Kanten zu ermitteln. Wenn wir einen finden, wenden wir ihn an und verkleinern so die Menge der einsamen Knoten. Wenn wir keinen finden, modifizieren wir die duale Lösung so, dass wir Optimalität beider Lösungen erkennen können oder weitere Kanten verfügbar werden und unsere Suche nach einem vergrößernden Pfad fortgesetzt werden kann.

Beispiel

Bevor wir das Verfahren analysieren, wollen wir es an einem Beispiel veranschaulichen. Wir wenden Algorithmus 5.6 auf die Problem Instanz in Abb. 5.32 an.

Tabelle 5.10 stellt den Ablauf des Verfahrens dar. Die jeweiligen Teilgraphen, die gefundene Matchings und Pfade bilden, sind in Abb. 5.33 bis Abb. 5.42 illustriert.

In Tabelle 5.10 korrespondiert jede der nummerierten Zeilen zu einer Iteration von Algorithmus 5.6. In der letzten Zeile sind die finale primale und die finale duale Lösung abgetragen, die nach der siebten Iteration erreicht werden. Wir wollen nun nachvollziehen, was in den einzelnen Iterationen geschieht. Wir initialisieren zunächst die primale und duale Lösung und erhalten ein leeres Matching und Werte

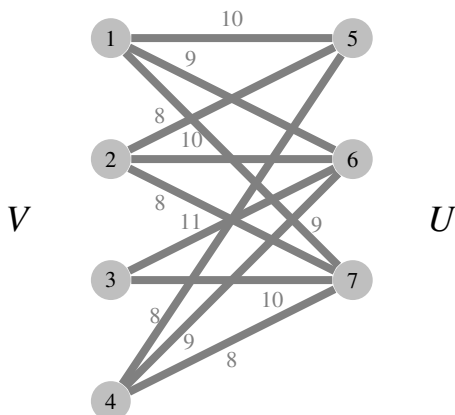


Abb. 5.32 Instanz von Problem 5.4 (Matching maximalen Gewichts) in bipartiten Graphen

Tabelle 5.10 Ablauf von Algorithmus 5.6 für die Instanz in Abb. 5.32

	M	α				β			E'	Abb.	Resultat
		1	2	3	4	5	6	7			
1	\emptyset	11	11	11	11	0	0	0	$\{(3, 6)\}$	5.33,5.34	Pfad (3, 6)
2	$\{(3, 6)\}$	11	11	11	11	0	0	0	$\{(3, 6)\}$	5.35	$\delta = \min\{11, 1\}$
3	$\{(3, 6)\}$	10	10	11	10	0	0	0	$\{(1, 5), (2, 6), (3, 6)\}$	5.36,5.37	Pfad (1, 5)
4	$\{(1, 5), (3, 6)\}$	10	10	11	10	0	0	0	$\{(1, 5), (2, 6), (3, 6)\}$	5.38	$\delta = \min\{10, 1\}$
5	$\{(1, 5), (3, 6)\}$	10	9	10	9	0	1	0	$\{(1, 5), (2, 6), (3, 6), (3, 7)\}$	5.39,5.40	Pfad (2, 6, 3, 7)
6	$\{(1, 5), (2, 6), (3, 7)\}$	10	9	10	9	0	1	0	$\{(1, 5), (2, 6), (3, 6), (3, 7)\}$	5.41	$\delta = \min\{9, 1\}$
7	$\{(1, 5), (2, 6), (3, 7)\}$	10	9	10	8	0	1	0	$\{(1, 5), (2, 6), (3, 6), (3, 7), (4, 5), (4, 6), (4, 7)\}$	5.42	$\delta = \min\{8, \infty\}$
	$\{(1, 5), (2, 6), (3, 7)\}$	2	1	2	0	8	9	8			

für die Dualvariablen, wie in Zeile 1 eingetragen. Die Werte der primalen und der dualen Lösung sind 0 und 44.

In Abb. 5.33 und in allen folgenden Abbildungen mit Pfaden sind alle verfügbaren Kanten dargestellt. Verfügbare Kanten, die in gefundenen Pfaden sind, sind dunkler abgebildet und mit einer Orientierung, wie aus Abschnitt 5.3.1.1 bekannt, versehen. Gestrichelte Kanten in Pfaden sind nicht im vorliegenden Matching und durchgezogene Kanten in Pfaden sind im Matching. In der ersten Iteration ist nur Kante (3, 6) verfügbar. Sie ist verfügbar, da $\alpha_3 + \beta_6 = 11 = w_{3,6}$ gilt. Sie bildet auch einen vergrößernden Pfad, den wir mittels Algorithmus 5.5 ermitteln, siehe Abb. 5.33.

Die Kante (3, 6) wird daher in das Matching aufgenommen und wir erhalten das Matching in Abb. 5.34.

In Iteration 2 gelingt es uns nicht, einen vergrößernden Pfad zu finden. Abb. 5.35 stellt alle mittels Algorithmus 5.5 gefundenen Pfade dar. Da die einzige Kante in E' schon im Matching ist, können wir gar keine Pfade positiver Länge finden und lediglich die einsamen Knoten in V werden markiert.

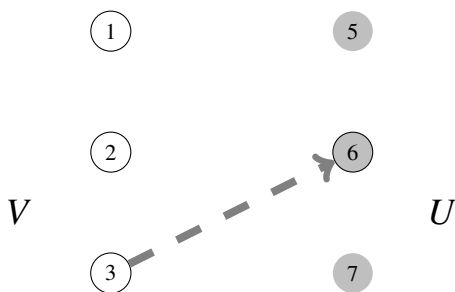


Abb. 5.33 Pfade in Iteration 1

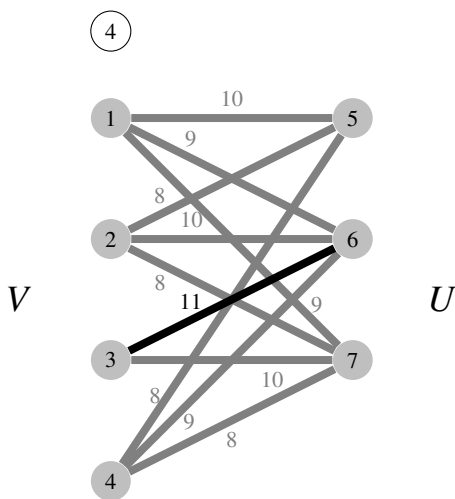


Abb. 5.34 Matching nach Iteration 1

Nun ermitteln wir δ^V , δ^E und abschließend δ . Wir haben $\alpha_v = 11$ für jeden Knoten $v \in V$. Dies gilt insbesondere für die ungeraden. Es gibt in E neun Kanten von einem ungeraden Knoten $v \in V$ zu einem unmarkierten Knoten $u \in U$. Davon minimieren die Kanten $(1, 5)$ und $(2, 6)$ den Term $\alpha_v + \beta_u - w_{v,u}$ und wir erhalten $\delta^E = 1$ und daher $\delta = 1$. Lediglich die einsamen Knoten 1, 2 und 4 wurden durch Algorithmus 5.5 markiert, siehe Abb. 5.35. Wir nehmen daher eine Anpassung der dualen Lösung vor, indem wir die dualen Variablen α_1 , α_2 und α_4 um $\delta = 1$ reduzieren, und verwenden diese in Iteration 3. Hier finden wir die zwei Pfade in Abb. 5.36.

Die Kante $(1, 5)$ bildet einen vergrößerten Pfad, wird daher in das Matching aufgenommen und wir erhalten das Matching in Abb. 5.37.

In Iteration 4 gelingt es uns nicht, einen vergrößerten Pfad zu finden. Abb. 5.38 stellt alle gefundenen Pfade dar. Wir erinnern uns, dass Kanten in E' , die nicht in Pfaden eingebunden sind, hellgrau dargestellt sind.

Wir ermitteln wiederum δ^V , δ^E und abschließend δ . Wir haben $\alpha_2 = \alpha_4 = 10$ und $\alpha_3 = 11$ für die ungeraden Knoten und daher $\delta^V = 10$. Es gibt in E fünf Kanten von einem ungeraden Knoten $v \in V$ zu einem unmarkierten Knoten $u \in U$. Un-

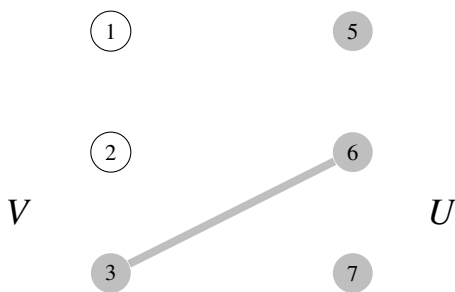


Abb. 5.35 Pfade in Iteration
2

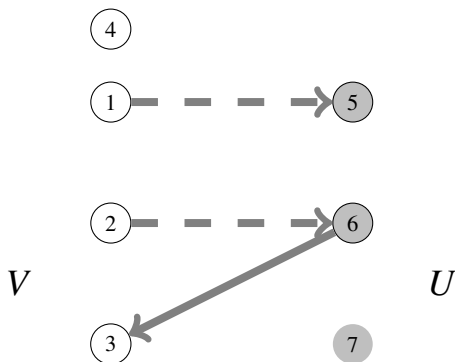


Abb. 5.36 Pfade in Iteration
3

ter diesen minimiert nur Kante $(3, 7)$ den Term $\alpha_v + \beta_u - w_{v,u}$ und wir erhalten $\delta^E = 1$ und daher $\delta = 1$. Lediglich die Knoten 2, 3, 4 und 6 wurden markiert, siehe Abb. 5.38. Wir nehmen daher eine Anpassung der dualen Lösung vor, indem wir die dualen Variablen α_2 , α_3 und α_4 um $\delta = 1$ reduzieren und β_6 um $\delta = 1$ vergrößern. Wir schauen uns hier etwas genauer an, was dies für die Dualvariablen β_5 bis β_7 und die Kante $(3, 6)$ im Matching bedeutet. Der Wert der Dualvariablen β_u wird nur vergrößert, wenn u durch eine Kante im Matching mit Knoten $v \in V$ verbunden ist, denn u muss gerade, aber nicht einsam sein (wäre u einsam, hätten wir einen vergrößerten Pfad gefunden). Daher bleibt Bedingungen 3. erfüllt. Außerdem verändert sich die Summe der beiden in Kante $(3, 6)$ involvierten Dualvariablen α_3 und β_6 nicht und daher bleibt $(3, 6)$ verfügbar und Bedingung 1. erfüllt. Die neue duale Lösung verwenden wir dann in Iteration 5. Hier finden wir die zwei Pfade in Abb. 5.39.

Wir haben den vergrößerten Pfad $(2, 6, 3, 7)$ ermittelt, wenden diesen auf das aktuelle Matching (siehe Abb. 5.37) an und erhalten das Matching in Abb. 5.40.

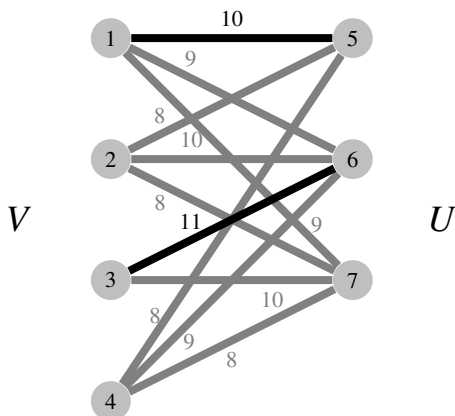


Abb. 5.37 Matching nach Iteration 3

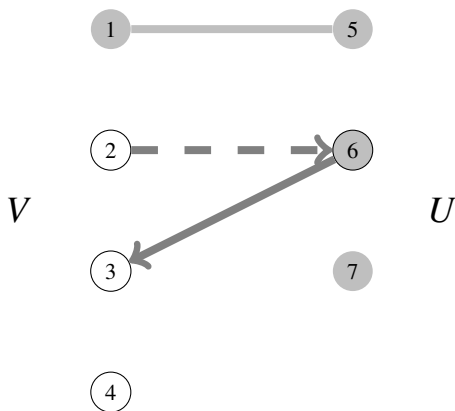


Abb. 5.38 Pfade in Iteration 4

In Iteration 6 gelingt es uns nicht, einen vergrößernden Pfad zu finden. Abb. 5.41 stellt alle gefundenen Pfade dar. Wir können lediglich den einsamen Knoten 4 markieren.

Nun ermitteln wir δ^V , δ^E und abschließend δ . Wir haben $\delta^V = \alpha_4 = 9$. Alle drei zu Knoten 4 inzidenten Kanten verbinden einen ungeraden Knoten mit einem unmarkierten Knoten $u \in U$. Diese drei Kanten minimieren auch jeweils den Term $\alpha_v + \beta_u - w_{v,u}$ und wir erhalten $\delta^E = 1$ und daher $\delta = 1$. Lediglich Knoten 4 wurde markiert, siehe Abb. 5.41. Wir nehmen daher eine Anpassung der dualen Lösung vor, indem wir die duale Variable α_4 um $\delta = 1$ reduzieren, und verwenden diese in Iteration 7. Allerdings finden wir auch in Iteration 7 keine vergrößernden Pfade, siehe Abb. 5.42.

Nun ermitteln wir δ^V , δ^E und abschließend δ . Es ergibt sich $\delta^V = \alpha_4 = 8$. Es gibt keine Kanten, die einen ungeraden Knoten mit einem unmarkierten Knoten $u \in U$ verbinden. Wir erhalten daher $\delta^E = \infty$ und somit $\delta = 8$. Alle Knoten wurden markiert, siehe Abb. 5.42. Wir nehmen daher eine Anpassung der dualen Lösung vor, indem wir die dualen Variablen α_1 , α_2 , α_3 und α_4 um $\delta = 8$ reduzieren und β_5 ,

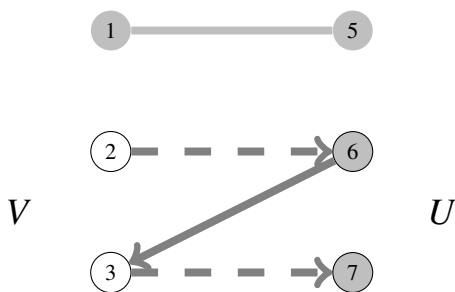


Abb. 5.39 Pfade in Iteration 5

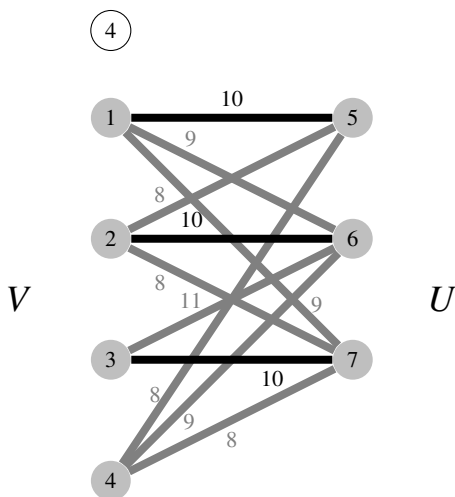


Abb. 5.40 Matching nach Iteration 5

β_6 und β_7 um $\delta = 8$ vergrößern. Nun ist auch Bedingung 2. erfüllt und das Verfahren endet. Wir sehen zudem, dass wir eine zulässige primale Lösung und eine zulässige duale Lösung mit gleichen Zielfunktionswerten von 30 gefunden haben.

Analyse

Nachdem wir gesehen haben, dass Algorithmus 5.6 die Probleminstanz in Abb. 5.32 löst, wollen wir uns nun überlegen, dass dies kein Zufall war. Wir werden hierbei in zwei Abschnitten vorgehen. Zunächst wollen wir argumentieren, dass die Modifikationen an der primalen Lösung und der dualen Lösung stets so vorgenommen werden, dass die Bedingungen 1. und 3. erfüllt sind. Danach werden wir herleiten, dass früher oder später auch Bedingung 2. erfüllt ist. Aus den obigen Überlegungen, die sich auf Theorem 3.10 stützen, folgt dann, dass Algorithmus 5.6 optimale Lösungen zu Problem 5.4 (Matching maximalen Gewichts) in bipartiten Graphen garantiert.

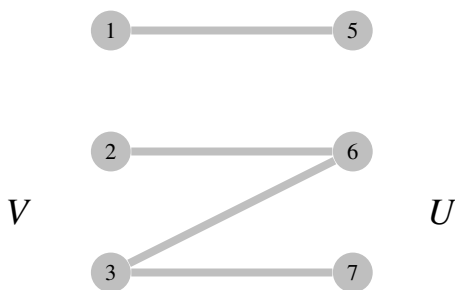


Abb. 5.41 Pfade in Iteration 6

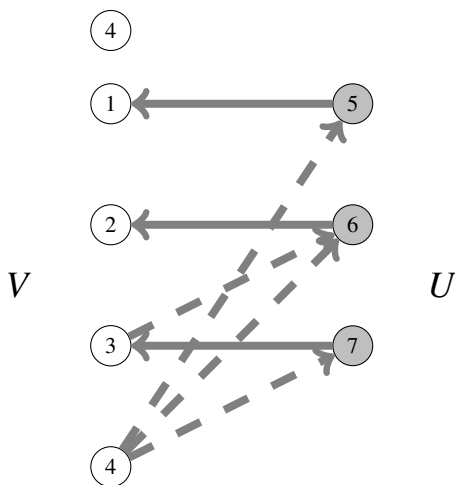


Abb. 5.42 Pfade in Iteration 7

Es ist relativ leicht ersichtlich, dass die initialen Lösungen zulässig sind und darüber hinaus Bedingungen 1. und 3. erfüllen. Bedingung 1. ist erfüllt, da $M = \emptyset$ und daher $\alpha_v + \beta_u = w_{v,u}$ für keine Kante (v, u) gelten muss. Bedingung 3. ist erfüllt, da $\beta_u = 0$ für jeden Knoten $u \in U$. Nun wollen wir uns die Veränderungen anschauen, die wir an den Lösungen vornehmen, und uns überlegen, dass wir nach jeder Modifikation zulässige primale und duale Lösungen erhalten und die Bedingungen 1. und 3. nach der jeweiligen Veränderung erfüllt sind, wenn sie vor der Veränderung erfüllt waren.

- Die primale Lösung verändern wir durch die Anwendung vergrößernder Pfade. Die primale Lösung bleibt in jedem Fall zulässig, denn wir erhalten ein neues Matching. Da in einem solchen Pfad nur verfügbare Kanten enthalten sind, wird durch die Veränderung des Matchings Bedingung 1. nicht verletzt, wenn das vorherige Matching und die duale Lösung Bedingung 1. erfüllten. Da zudem kein Knoten einsam wird (der zuvor nicht einsam war), ist auch Bedingung 3. erfüllt, wenn Bedingung 3. vor der Modifikation erfüllt war.

- Die duale Lösung verändern wir durch Verringerung von α_v für jeden ungeraden Knoten v und Vergrößern von β_u für jeden geraden Knoten u . Diese Modifikation führen wir nur dann durch, wenn wir keinen vergrößernden Pfad gefunden haben. In diesem Fall ist jeder gerade Knoten $u \in U$ gepaart, denn sonst würde bei u ein vergrößernder Pfad enden.

Die Modifikation kann nicht zu einer unzulässigen dualen Lösung führen. Da $\delta \leq \delta^V \leq \alpha_v$ für jeden ungeraden Knoten $v \in V$, gilt $\alpha_v - \delta \geq 0$ und alle Variablenwerte bleiben nicht-negativ. Da zudem $\delta \leq \delta^E \leq \alpha_v + \beta_u - w_{v,u}$ für jede Kante zwischen ungeradem Knoten $v \in V$ und unmarkiertem Knoten $u \in U$, gilt $\alpha_v - \delta + \beta_u \geq w_{v,u}$. Für jede andere Kante (u, v) gilt, dass der Wert von $\alpha_v + \beta_u$ nicht geändert wird, denn wenn u gerade ist, ist u 's Partner ungerade, und daher gilt auch nach der Modifikation $\alpha_v + \beta_u \geq w_{v,u}$.

Aus den vorherigen Überlegungen folgt auch, dass Bedingung 1. nach der Modifikation erfüllt ist, denn der Wert von $\alpha_v + \beta_u$ wird für keine Kante (u, v) vergrößert und somit kann keine vormals verfügbare Kante nicht mehr verfügbar sein. Schließlich kann Bedingung 3. durch die Modifikation nicht verletzt werden, da u nicht einsam ist, falls u gerade ist, denn sonst läge ein vergrößernder Pfad vor.

Zusammenfassung: Erfüllung der Bedingungen 1. und 3. durch zulässige Lösungen

Durch Modifikation der primalen bzw. dualen Lösung erhalten wir eine zulässige primale bzw. zulässige duale Lösung. Nach Modifikation einer der beiden Lösungen erfüllen die vorliegende primale Lösung und die vorliegende duale Lösung die Bedingungen 1. und 3.

Nachdem wir nun gesehen haben, dass die Bedingungen 1. und 3. im Ablauf von Algorithmus 5.6 nicht verletzt werden, wollen wir nun dafür argumentieren, dass früher oder später auch Bedingung 2. erfüllt ist. Hierfür legen wir unser Hauptaugenmerk zunächst auf die dualen Lösungen bzw. deren Modifikation. In der initialen dualen Lösung haben alle Dualvariablen zu Knoten in V denselben Wert. Bei jeder Modifikation der dualen Lösung werden die Werte der Dualvariablen zu einsamen Knoten in V reduziert (womöglich gibt es noch weitere ungerade Knoten, aber die einsamen Knoten in V sind auf jeden Fall ungerade). Wie schon oben besprochen, wird ein Knoten im Verlauf des Verfahrens nicht mehr einsam werden, nachdem er einmal gepaart war. Daraus folgt, dass die Dualvariablen zu einsamen Knoten in V gleiche Werte haben und dass die Dualvariable keines anderen Knotens in V einen niedrigeren Wert haben kann.

Wenn also $\delta^V \leq \delta^E$ bei einer Modifikation der dualen Lösung gilt, dann haben in der neuen dualen Lösung die Dualvariablen zu allen einsamen Knoten in V einen Wert von null, das Verfahren endet und das vorliegende Matching ist gemäß Theorem 3.10 optimal. Wenn hingegen $\delta^V > \delta^E$, dann wird mindestens eine weitere Kante (nämlich jede Kante in $E \setminus E'$, die einen ungeraden Knoten $v \in V$ mit einem unmarkierten Knoten $u \in U$ verbindet und den Term $\alpha_v + \beta_u - w_{v,u}$ unter

diesen Kanten minimiert) in E' aufgenommen. Letzterer Fall kann aber nur $|E|$ -mal eintreten, da keine Kante (v, u) aus E' entnommen wird (wir haben schon oben argumentiert, dass der Wert von $\alpha_v + \beta_u$ für keine Kante (u, v) vergrößert wird). D. h., spätestens nach $|E|$ Iterationen wird entweder das Verfahren abgebrochen (da die Dualvariablen zu den einsamen Knoten in V einen Wert von null haben) oder die primale Lösung wird modifiziert. Da aber bei Modifikation der primalen Lösung die Kardinalität des vorliegenden Matchings steigt, kann auch dies nur begrenzt häufig passieren.

Daher muss nach einer endlichen Anzahl von Iterationen der Fall eintreten, in dem das Verfahren abbricht, und da es nur mit einer optimalen Lösung abbricht, können wir folgendes Theorem festhalten.

Theorem 5.8. *Algorithmus 5.6 findet in endlicher Zeit eine optimale Lösung zu Problem 5.4 (Matching maximalen Gewichts) in bipartiten Graphen.*

5.3.2 Allgemeine Graphen

Matchings in allgemeinen Graphen spielen häufig eine Rolle, wenn es darum geht, Paare aus Elementen einer einzelnen Menge zu bilden. Ein klassisches Anwendungsbeispiel ist die Bildung von Teams bestehend aus jeweils zwei Mitarbeitern. Hierbei gibt es für jeden Mitarbeiter einen Knoten. Es gibt eine Kante zwischen zwei Knoten, wenn die beiden Mitarbeiter ein Team bilden können. Ein Matching maximaler Kardinalität würde dann eine maximale Anzahl von Teams darstellen. In einem gewichteten Graphen könnte das Gewicht der Kante $\{v, u\}$ die Effizienz des Teams darstellen, die davon abhängt, wie gut sich die jeweiligen Mitarbeiter v und u in ihren Fähigkeiten ergänzen. Ein Matching mit maximalem Gewicht stellt dann eine Menge von Teams dar, sodass die erzielte Effizienz maximal ist.

Wir werden im Folgenden zunächst Problem 5.3 (Matching maximaler Kardinalität) betrachten und uns im Anschluss Problem 5.4 (Matching maximalen Gewichts) zuwenden. Wir werden auch für allgemeine Graphen das Gerüst in Algorithmus 5.4 auf S. 315 nutzen und uns zunächst überlegen, wie wir vergrößernde Pfade finden können, um Matchings maximaler Kardinalität in allgemeinen Graphen zu finden. Im Anschluss werden wir analog zu Abschnitt 5.3.1 ein Verfahren betrachten, das eine primale und eine duale Lösung simultan modifiziert, um Matchings maximalen Gewichts in allgemeinen Graphen zu ermitteln. Bei der Analyse dieser Verfahren wird uns das Verständnis der Verfahren in Abschnitt 5.3.1 helfen.

5.3.2.1 Matchings maximaler Kardinalität

Die Suche nach vergrößernden Pfaden wurde uns in Abschnitt 5.3.1 durch die Beschränkung auf bipartite Graphen recht einfach gemacht. Wir konnten uns vorstellen, Pfade von einsamen Knoten in V aus voranzutreiben und dabei abwechselnd ungerade und gerade Knoten zu besuchen. Wenn sich zwei solche Pfade bei einem

Knoten $u \in U$ trafen, waren sie insofern gleichwertig, als dass wir beide von u aus zu einem vergrößernden Pfad vervollständigen konnten oder dies für beide nicht möglich war. Dies begründet sich durch den bipartiten Graphen, denn beide Pfade haben ungerade Länge (und durch die Konstruktion abwechselnd Kanten in $E \setminus M$ und M). Diese Gewissheit haben wir bei allgemeinen Graphen $G = (V, E)$ nicht und müssen uns daher genau überlegen, wie wir mit der Situation, in der sich zwei Pfade treffen, umgehen.

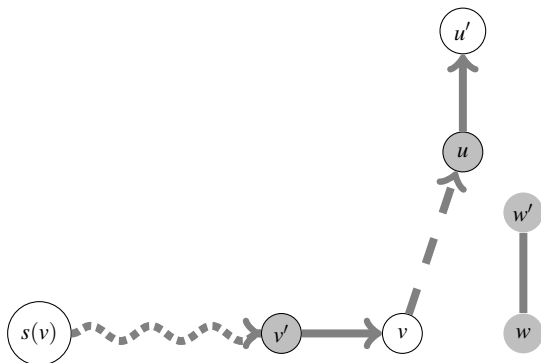
Überblick über das Verfahren

Wir werden uns im Folgenden vorstellen, dass wir vergrößernde Pfade für einen Graph G und ein vorliegendes Matching M suchen, indem wir von einsamen Knoten in V ausgehen und versuchen, durch schrittweises Ergänzen weiterer Kanten einen Pfad zu einem anderen einsamen Knoten in V zu finden. Wir markieren dabei Knoten, die wir schon erreicht haben und die dementsprechend als Ausgangspunkt für weitere anzufügende Kanten dienen können. Initial markieren wir alle einsamen Knoten als ungerade. Weiterhin nummerieren wir markierte Knoten im Verlauf des Verfahrens entlang der gefundenen Pfade und bezeichnen die markierten Knoten dann entsprechend dieser Nummerierung als *ungerade* oder *gerade*. Wir werden uns zudem für jeden erreichten Knoten v merken, bei welchem einsamen Knoten der Pfad, der v erreicht, startet, und diesen als Startpunkt $s(v)$ von v bezeichnen. Diese Vorgehensweise entspricht weitestgehend der in Algorithmus 5.5. In jedem Schritt betrachten wir einen ungeraden Knoten v und eine inzidente Kante $\{v, u\} \in E \setminus M$. Dabei unterscheiden wir drei Fälle.

1. Wenn die Kante von v zu einem unmarkierten Knoten u führt, dann muss dieser Knoten gepaart sein, denn sonst wäre er markiert. Wir markieren u dann als gerade und seinen Partner als ungerade. Auch der Partner kann zuvor nicht markiert gewesen sein, denn sonst wäre u durch den gerade beschriebenen Mechanismus schon markiert gewesen.

Diese Situation ist in Abb. 5.43 dargestellt. Wie schon in Abschnitt 5.3.1 stellen wir Kanten hier mit einer Orientierung entsprechend ihrer Orientierung im Pfad dar. Gestrichelte Kanten sind in $E \setminus M$ und führen von einem ungeraden Knoten zu einem geraden Knoten, während durchgezogene Kanten in M sind und von einem geraden Knoten zu seinem Partner führen, der als ungerade markiert wird. Zudem sind markierte Knoten mit einem schwarzen Rand dargestellt. Dabei sind ungerade Knoten weiß und gerade Knoten grau. Alle übrigen, also die unmarkierten, Knoten sind ohne Rand und in grau dargestellt.

Wir sehen in Abb. 5.43 den ungeraden Knoten v und den Startpunkt $s(v)$ des Pfades, der zu v führt. Der Pfad selber ist nicht näher ausgewiesen. Es kann sein, dass $s(v) = v$, d. h., v ist selbst ein einsamer Knoten. Wenn dies nicht der Fall ist, dann gibt es einen Partner v' von v im vorliegenden Matching, von dem aus v auf dem Pfad erreicht wurde. Knoten w und w' sind zwei unmarkierte Knoten mit $\{w, w'\} \in M$. Vor der Verlängerung des Pfades waren u und u' ebenfalls unmarkiert. Wir haben dann den ungeraden Knoten v und die inzidente Kante

Abb. 5.43 Pfad von $s(v)$ zu u'

$\{v, u\} \in E \setminus M$ betrachtet, u als gerade und dessen Partner u' als ungerade markiert und so den Pfad verlängert.

2. Wenn die Kante zu einem ungeraden Knoten u führt, der nicht denselben Startpunkt wie v hat, d. h., wir haben $s(v) \neq s(u)$, dann ist der Pfad von $s(v)$ zu v , verlängert um die Kante $\{v, u\}$ und den Pfad von u zu $s(u)$, ein vergrößernder Pfad. Diese Situation ist in Abb. 5.44 dargestellt. Die Orientierung der Kanten entspricht der Orientierung, in der sie in Pfade aufgenommen wurden.

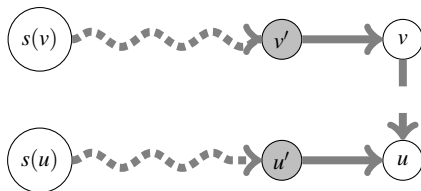


Abb. 5.44 Vergrößernder Pfad

In Abb. 5.44 sehen wir die ungeraden Knoten v und u sowie ihre Startpunkte $s(v)$ und $s(u)$ und die zugehörigen Pfade. Zunächst waren die Pfade nicht verbunden. Dann haben wir den ungeraden Knoten v und die inzidente Kante $\{v, u\} \in E \setminus M$ betrachtet und u als ungerade markiert erkannt. Da $s(v) \neq s(u)$, haben wir einen vergrößernden Pfad von $s(v)$ zu $s(u)$ (oder umgekehrt) gefunden.

3. Wenn die Kante zu einem ungeraden Knoten u mit demselben Startpunkt führt, d. h., wir haben $s(v) = s(u)$, dann bilden die beiden Pfade zusammen eine Struktur, die wir Blume nennen werden. Die Blume ist die Vereinigung der Kantenmengen der Pfade zu v und zu u , ergänzt um die Kante $\{v, u\}$. Eine *Blume* besteht dann aus einem Kreis, der *Blüte* (im Englischen „Blossom“), und einem Pfad, dem *Stiel* (im Englischen „Stem“), der die Blüte mit $s(v) = s(u)$ verbindet. Den Knoten, der sowohl Teil der Blüte als auch Teil des Stiels ist, nennen wir den *Ursprung der Blüte*. Knoten $s(v) = s(u)$ nennen wir die *Wurzel der Blume*. Es ist möglich, dass der Stiel eine Länge von null hat und somit die Wurzel der Blume auch der Ursprung der Blüte ist. In jedem Fall sind sowohl die Wurzel als auch

der Ursprung der Blüte ungerade. Jeder Knoten in der Blume bis auf die Wurzel ist gepaart und sein Partner ist ebenfalls in der Blume.

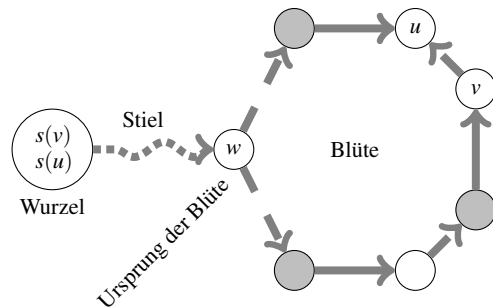


Abb. 5.45 Blume

In Abb. 5.45 sehen wir die ungeraden Knoten v und u sowie ihren gemeinsamen Startpunkt $s(v) = s(u)$ und die jeweiligen Pfade, die zu v und u führen. Die beiden Pfade verlaufen bis Knoten w , dem Ursprung der Blüte, gleich. Wir haben dann den ungeraden Knoten v und die inzidente Kante $\{v, u\} \in E \setminus M$ betrachtet und u als ungerade markiert erkannt. Da $s(v) = s(u)$, haben wir einen Kreis, die Blüte, gefunden. Die Blume besteht neben der Blüte aus dem Stiel, dem Pfad von w zu $s(v) = s(u)$.

Eine Blume erschwert uns das Finden eines vergrößernden Pfades insofern, als dass jeder Knoten in der Blüte außer dem Ursprung sowohl gerade als auch ungerade sein könnte (abhängig davon, in welcher Richtung man vom Stiel aus die Blüte durchläuft). Edmonds (1965b) entwickelte für dieses Problem eine einfache, elegante Lösung. Eine gefundene Blüte wird *geschrumpft* (im Englischen „to shrink“), d. h., wir modifizieren den vorliegenden Graphen, sodass alle Knoten in der Blüte durch einen einzelnen Knoten ersetzt werden. Wenn $B \subset V$ die Knoten in der Blüte sind, erhalten wir also einen neuen Graphen $G' = (V', E')$ mit $V' = (V \setminus B) \cup \{b\}$, wobei b der neue Knoten ist, der die Blüte ersetzt. Die Kantenmenge E' ergibt sich durch Kanten in G zwischen Knoten, die nicht in der Blüte sind, und einer Kante zwischen b und jedem Knoten $v \notin B$, der in G adjazent zu einem Knoten in B ist. Alle Markierungen von Knoten außerhalb der Blüte (also in $V \setminus B$) bleiben bestehen und b selber wird als ungerade markiert (entsprechend des bereits gefundenen Pfades von $s(v)$ zu der Blüte bzw. zu der geschrumpften Blüte b). Den Vorgang des Schrumpfens werden wir im Folgenden noch im Detail betrachten.

Zusammenfassung: Vergrößernde Pfade und Blumen

Für ein vorliegendes Matching M konstruieren wir Pfade, die an einem einsamen Knoten in V starten und abwechselnd Kanten in $E \setminus M$ und in M enthalten. Dabei können wir einen vergrößernden Pfad, den wir auf das vorliegende Matching anwenden, oder eine Blume finden.

Eine Blume besteht aus zwei Pfaden zu ungeraden Knoten mit demselben Startpunkt und einer Kante zwischen diesen beiden ungeraden Knoten. Der Startpunkt ist die Wurzel der Blume, der Teil, in dem die beiden Pfade übereinstimmen, ist der Stiel der Blume und die übrigen Kanten bilden die Blüte. Wenn wir eine Blume finden, schrumpfen wir deren Blüte und verringern so die Größen des betrachteten Graphen und des betrachteten Matchings.

Wir wollen uns nun intensiver mit Blüten und dem Schrumpfen derselben beschäftigen und uns dafür zunächst ein Beispiel anschauen. In Abb. 5.46 sehen wir einen Graphen mit den 18 Knoten 1 bis 16 sowie v und u . Kanten, die in keinem Pfad sind, sind hellgrau dargestellt.

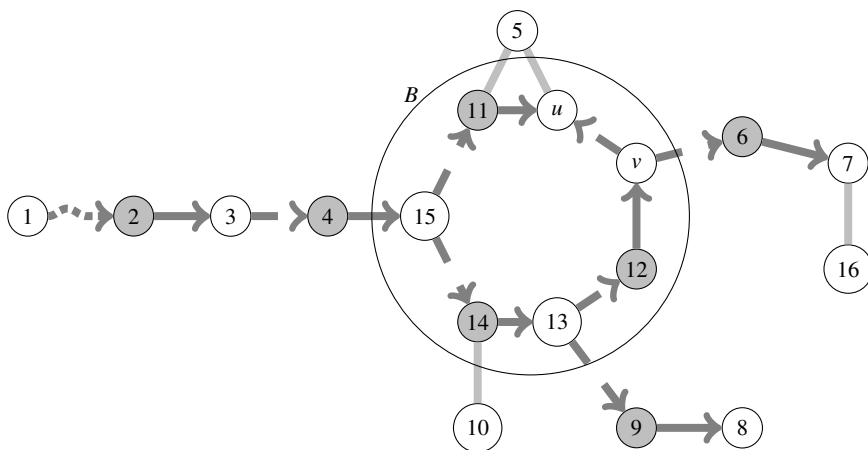


Abb. 5.46 Noch eine Blume

Die Blume wird durch die Betrachtung der Kante $\{v, u\}$ gefunden und besteht aus der Blüte B mit Knoten 11 bis 15 sowie v und u und dem Stiel mit Knoten 1 und 2, allen ggf. zwischen 1 und 2 liegenden Knoten sowie Knoten 3, 4 und 15. Knoten 1 ist die Wurzel der Blume und Knoten 15 der Ursprung der Blüte. Bevor die Blume gefunden wurde, wurden neben $(1, \dots, 2, 3, 4, 15, 11, u)$ weitere Pfade, die von Knoten 1 ausgehen und durch Knoten der Blüte verlaufen, gefunden, nämlich $(1, \dots, 2, 3, 4, 15, 14, 13, 9, 8)$ und $(1, \dots, 2, 3, 4, 15, 14, 13, 12, v, 6, 7)$. Weitere Kanten in $E \setminus M$ sind inzident mit Knoten in der Blüte, nämlich $\{5, 11\}$, $\{5, u\}$ und $\{10, 14\}$. Darüber hinaus gibt es eine weitere Kante $\{7, 16\}$ in $E \setminus M$. Alle Kan-

ten im vorliegenden Matching sind also auch als durchgezogene Kanten in Pfaden eingebunden (daher sind Knoten 5, 10 und 16 einsam und als ungerade markiert).

Für dieses Beispiel können wir erraten, dass es vergrößernde Pfade für G und M gibt. Knoten 16 ist einsam und somit gibt es den vergrößernden Pfad $(1, \dots, 2, 3, 4, 15, 14, 13, 12, v, 6, 7, 16)$. Wir können drei weitere vergrößernde Pfade erkennen, die bei Knoten 1 starten und bei Knoten 5 bzw. 10 enden, nämlich $(1, \dots, 2, 3, 4, 15, 11, u, 5)$, $(1, \dots, 2, 3, 4, 15, 14, 13, 12, v, 11, 5)$ und $(1, \dots, 2, 3, 4, 15, 11, u, v, 12, 13, 14, 10)$. Wir unterstellen allerdings, dass wir keine Kenntnis dieser vergrößernden Pfade haben und daher die Blüte schrumpfen.

Das Schrumpfen der Blüte B erzeugt den Knoten b , der adjazent zu genau den Knoten in $V \setminus B$ ist, die adjazent zu mindestens einem Knoten in B waren. Kanten zwischen Knoten in der Blüte und Knoten außerhalb der Blüte können dabei unterschiedliche Rollen im vorliegenden Matching und gefundenen Pfaden haben.

- Es kann höchstens eine Kante zwischen B und $V \setminus B$ geben, die im vorliegenden Matching ist. Nur der Ursprung der Blüte hat keinen Partner in der Blüte. Wenn er auch die Wurzel ist, hat er gar keinen Partner, ansonsten liegt sein Partner im Stiel und außerhalb der Blüte.
- Es kann beliebig viele Kanten in $E \setminus M$ zwischen B und $V \setminus B$ geben, die in Pfaden vorkommen. Für jede solche Kante $\{w, w'\} \in E \setminus M$ mit $w \in B$ und $w' \notin B$ ist w der Vorgänger von w' auf dem gefundenen Pfad von $s(w')$ zu w' . Dabei ist $s(w')$ die Wurzel der Blume, denn alle Knoten in der Blüte (und somit auch w') haben denselben Startpunkt.
- Es kann beliebig viele Kanten in $E \setminus M$ zwischen B und $V \setminus B$ geben, die nicht in Pfaden vorkommen.

Wir erhalten durch Schrumpfen den Graphen $G' = (V', E')$ mit $V' = (V \setminus B) \cup \{b\}$, wobei b der neue Knoten ist, der die Blüte ersetzt. Die Kantenmenge E' ergibt sich durch Kanten in G zwischen Knoten, die nicht in der Blüte sind, und einer Kante zwischen b und jedem Knoten $v \notin B$, der in G adjazent zu einem Knoten in B ist.

Nach Schrumpfen der Blüte betrachten wir auch ein anderes Matching M' und andere Pfade, denn einige Kanten in M oder in den Pfaden gibt es in E' gar nicht. Matching M' enthält alle Kanten in M , die zwei Knoten, die nicht in B sind, verbinden. Wenn der Ursprung der Blüte in M einen Partner hat, dann enthält M' zusätzlich die Kante, die b mit diesem Partner verbindet. In jedem gefundenen Pfad werden alle Knoten in B durch den ungeraden Knoten b ersetzt. Dies erlaubt, dass alle Markierungen von Knoten in $V \setminus B$ bestehen bleiben und b selber als ungerade markiert wird (entsprechend des bereits gefundenen Pfades von $s(v)$ zu der Blüte bzw. zu der geschrumpften Blüte b).

In Abb. 5.47 sehen wir den Graphen, den wir aus dem Graphen in Abb. 5.46 durch Schrumpfen der Blüte erhalten. Die Blüte B wurde durch Knoten b ersetzt und jeder übrige Knoten ist durch eine Kante in E' mit b verbunden, wenn er in G adjazent zu mindestens einem Knoten in B ist. Die Kante in M zwischen dem Ursprung der Blüte und seinem Partner wurde durch eine Kante in M' zwischen b und demselben Partner ersetzt. Jeder Knoten, der auf einem Pfad durch eine Kante

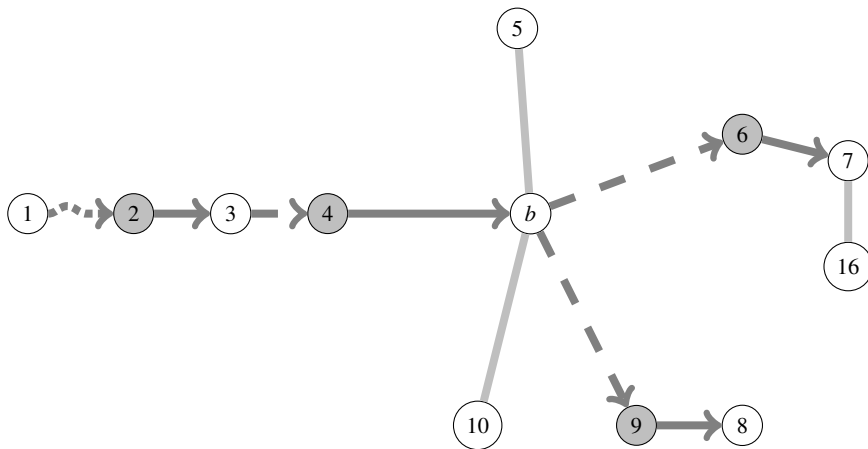


Abb. 5.47 Geschrumpfte Blüte

in $E \setminus M$ von einem Knoten in B aus erreicht wurde, wird durch eine Kante in $E' \setminus M'$ von b aus erreicht. Jeder Knoten, der zwar adjazent zu Knoten in B war, aber nicht durch einen Pfad von einem dieser Knoten aus erreicht wurde, ist adjazent zu b , wird aber nicht durch einen Pfad von b aus erreicht. Alle Markierungen von Knoten in $V \setminus B$ wurden übernommen.

Wir merken uns beim Schrumpfen, welche Knoten in B waren, sodass wir jede geschrumpfte Blüte im Nachgang wieder aufblättern können. Wir werden von diesem Vorgang als *Interpretieren* einer geschrumpften Blüte sprechen.

Zusammenfassung: Schrumpfen und Interpretieren von Blüten

Beim Schrumpfen wird eine Blüte B durch einen einzelnen Knoten b ersetzt. Dabei werden alle Kanten in M bzw. in $E \setminus M$ zwischen Knoten in B und Knoten in $V \setminus B$ durch Kanten in M' bzw. in $E' \setminus M'$ zwischen b und den Knoten in $V \setminus B$ ersetzt. Auch die Rollen der Kanten in Pfaden werden auf die entsprechenden Kanten übertragen.

Das Interpretieren einer geschrumpften Blüte ist die Umkehrung dieses Vorgangs. Um dies zu ermöglichen, merken wir uns beim Schrumpfen die betroffenen Knoten und Kanten.

Das Schrumpfen ergibt nur dann Sinn, wenn wir mit G' so weiter arbeiten können, dass wir sinnvolle Informationen für G ableiten. Dass dies tatsächlich der Fall ist, drückt das folgende Theorem aus.

Theorem 5.9. *Es gibt einen vergrößernden Pfad für Graph G' und Matching M' (mit geschrumpfter Blüte) genau dann, wenn es einen vergrößernden Pfad für G und M (mit ungeschrumpfter Blüte) gibt.*

Wir wollen hier argumentativ nur auf eine Hälfte von Theorem 5.9 eingehen. Wenn es uns gelingt, einen vergrößernden Pfad P' für G' und M' zu finden, dann

können wir aus diesem einen vergrößernden Pfad P für G und M konstruieren. Wir werden im Folgenden beschreiben, wie diese Konstruktion erfolgt. Um dies zu zeigen, unterscheiden wir zwei Fälle.

- Wenn P' nicht durch b läuft, dann enthält er ausschließlich Kanten, die auch in E sind. Die Kanten in P' , die in M' sind, sind auch in M . Daher ist P' selber ein vergrößernder Pfad für G und M .
- Wenn P' durch b läuft, dann ist b im Pfad durch eine Kante in $E' \setminus M'$ mit einem Knoten v' verbunden, denn jeder vergrößernde Pfad enthält mindestens eine Kante in $E' \setminus M'$ und Kanten aus $E' \setminus M'$ und M' wechseln sich ab.

Wir konstruieren dann P wie folgt. Wir übernehmen den Abschnitt von P' , der ein Ende von P' mit v' verbindet, ohne durch b zu laufen, direkt in P . Dieser Teil enthält abwechselnd Kanten in $E' \setminus M'$ und M' . Da beim Schrumpfen nur Kanten aus E oder M entfernt wurden, die inzident zu mindestens einem Knoten aus B sind, enthält dieser Teil von P' also auch abwechselnd Kanten in $E \setminus M$ und M . Wir ergänzen eine Kante, die in G v' mit einem Knoten in $v'' \in B$ verbindet. Es kann mehrere solche Kanten geben. Wenn dies der Fall ist, können wir eine dieser Kanten zufällig wählen. Genauso wie $\{v', b\}$ nicht in M' ist, ist $\{v', v''\}$ nicht in M .

- Wenn v'' der Ursprung von B ist, dann fügen wir den Stiel, also den Pfad von v'' zu $s(v'')$, an den bisher konstruierten Pfad an. Wenn der Stiel überhaupt Kanten enthält, ist die erste Kante in M .
- Wenn v'' nicht der Ursprung von B ist, dann gibt es einen Pfad von v'' zu $s(v'')$, der ausschließlich Kanten aus der Blume benutzt und mit einer Kante in M beginnt. Diesen fügen wir an den bisher konstruierten Pfad an.

Da der im letzten Schritt ergänzte Pfad von v'' zu $s(v'')$ abwechselnd Kanten aus M und $E \setminus M$ enthält und an dem einsamen Knoten $s(v'')$ endet, ist Pfad P ein vergrößernder Pfad für G und M .

Den zweiten oben beschriebenen Fall wollen wir anhand des Beispiels in den Abb. 5.46 und 5.47 nachvollziehen. Wir haben schon festgestellt, dass es mehrere vergrößernde Pfade für G und M in Abb. 5.46 gibt. Jeder dieser Pfade hat eine Entsprechung in Abb. 5.47. Wir wollen annehmen, dass wir den vergrößernden Pfad $(1, \dots, 2, 3, 4, b, 6, 7, 16)$ für G' und M' gefunden haben, und diesen nun als vergrößernden Pfad für G und M interpretieren müssen. In diesem Fall haben wir $v' = 6$ mit $\{6, b\} \notin M'$. Knoten $v' = 6$ hat nur v als adjazenten Knoten in B in Graph G und daher erhalten wir eindeutig $v'' = v$. Knoten $v'' = v$ ist nicht der Ursprung von B und somit müssen wir uns überlegen, wie wir den Pfad von Knoten 16 zu $v'' = v$ durch einen Pfad von v'' zu $s(v'') = 1$ vervollständigen. Die erste Kante nach v'' muss in M sein (denn die letzte Kante vor v'' war in $E \setminus M$) und daher ergänzen wir den Pfad $(v, 12, 13, 14, 15, 4, 3, 2, \dots, 1)$ und erhalten den vergrößernden Pfad $(16, 7, 6, v, 12, 13, 14, 15, 4, 3, 2, \dots, 1)$ für G und M . Analog können wir die beiden anderen vergrößernden Pfade $(10, b, 4, 3, 2, \dots, 1)$ und $(5, b, 4, 3, 2, \dots, 1)$ für G' und M' interpretieren, wobei es für letzteren noch eine Besonderheit zu beachten gibt. Abhängig davon, ob wir $v'' = 11$ oder $v'' = u$ wählen (Knoten $v' = 5$

hat zwei adjazente Knoten in B), verläuft der vergrößernde Pfad für G und M ab v'' mit dem Uhrzeigersinn oder gegen den Uhrzeigersinn in der Blüte. Wir erhalten dann im ersten Fall $(5, 11, u, v, 12, 13, 14, 15, 4, 3, 2, \dots, 1)$ und im zweiten Fall $(5, u, 11, 15, 4, 3, 2, \dots, 1)$ als vergrößernden Pfad für G und M .

Wir haben uns diese Hälfte von Theorem 5.9 genauer angeschaut, da wir durch diese Konstruktion einen vergrößernden Pfad in dem geschrumpften Graph G' in einen vergrößernden Pfad in dem Ursprungsgraph G übersetzen können. Dies wird uns im Rahmen des gesamten Verfahrens nützen und daher wollen wir die Vorgehensweise in Algorithmus 5.7 festhalten.

Algorithmus 5.7 (Vergrößernder Pfad für G und M).

Eingabe: Graph $G = (V, E)$, Matching M , Blume mit Blüte B , vergrößernder Pfad P' für G' und M'

Ausgabe: Vergrößernder Pfad P für G und M

1. Wenn P' nicht durch b läuft, dann setze $P \leftarrow P'$. STOP.
2. Identifiziere Knoten $v' \in V'$, der mit einer Kante in $E' \setminus M'$ in P' mit b verbunden ist, und beginne P mit dem Abschnitt von P' , der ein Ende von P' mit v' verbindet, ohne durch b zu laufen.
3. Identifiziere Knoten $v'' \in B$, der mit einer Kante in $E \setminus M$ mit v' verbunden ist, und verlängere P um die Kante $\{v', v''\}$.
4. Identifiziere Pfad P'' , der in der Blume von $v'' \in B$ zu der Wurzel führt und dabei nicht mit einer Kante in $E \setminus M$ beginnt, und verlängere P um diesen Pfad. STOP.

Die andere Hälfte von Theorem 5.9 besagt, dass es einen vergrößernden Pfad P' für G' und M' gibt, wenn es einen vergrößernden Pfad P für G und M gibt. Diese Hälfte haben wir hier nicht besprochen, der vollständige Beweis von Theorem 5.9 findet sich aber in Anhang C.7. Theorem 5.9 rechtfertigt dann unser Vorgehen, eine gefundene Blüte durch einen einzigen Knoten zu ersetzen und mit dem geschrumpften Graphen fortzufahren, um einen vergrößernden Pfad zu finden. Dies kann mehrfach passieren und dabei können Blüten sogar ineinander verschachtelt sein. Der Knoten b , der eine Blüte repräsentiert, kann später selber in einer Blüte sein, die wiederum geschrumpft wird. Beim Schrumpfen einer Blüte wird die Anzahl von Knoten immer um mindestens zwei reduziert (die kleinstmögliche Blüte hat drei Knoten) und daher kann dieser Schritt des Schrumpfens einer Blüte nur begrenzt oft ausgeführt werden.

Wenn wir (womöglich nach mehrmaligem Schrumpfen von Blüten) einen vergrößernden Pfad gefunden haben, müssen wir diesen unter Berücksichtigung der geschrumpften Blüten interpretieren. Dies geschieht mittels Algorithmus 5.7. Wenn dabei ineinander verschachtelte Blüten zu beachten sind, müssen wir darauf achten, dass wir die geschrumpften Blüten in umgekehrter Reihenfolge, in der sie geschrumpft wurden, interpretieren.

Zusammenfassung: Vergrößernde Pfade in geschrumpften Graphen

Unser Ziel ist, einen vergrößernden Pfad für G und M zu finden oder die Existenz eines solchen Pfades auszuschließen. Wenn wir bei der Konstruktion von Pfaden auf eine Blüte stoßen, birgt das Komplikationen, weil die Knoten in der Blüte willkürliche Markierungen haben (je nachdem wie der Pfad zu den Knoten durch die Blüte verläuft).

Wir schrumpfen die Blüte dann zu einem einzigen Knoten und suchen einen vergrößernden Pfad für den resultierenden Graphen G' und das resultierende Matching M' . Dabei wissen wir wegen Theorem 5.9, dass ein solcher genau dann existiert, wenn einer für G und M existiert.

Wenn wir also nach (womöglich mehrmaligem) Schrumpfen keinen vergrößernden Pfad finden, dann gibt es auch keinen vergrößernden Pfad für G und M . Sollten wir einen vergrößernden Pfad für G' und M' finden, dann nutzen wir die Freiheit bei der Markierung der Knoten in der (geschrumpften) Blüte, um aus diesem einen vergrößernden Pfad für G und M abzuleiten.

Nun haben wir alle Komponenten beisammen, um die vollständige Vorgehensweise zum Finden eines vergrößernden Pfades als Algorithmus 5.8 zu spezifizieren.

Algorithmus 5.8 (Vergrößernde Pfade in Graphen).

Eingabe: Graph $G = (V, E)$, Matching M

Ausgabe: Vergrößernder Pfad für G und M oder Information, dass keiner existiert

1. Setze $V' \leftarrow V$, $E' \leftarrow E$ und $M' \leftarrow M$.
2. Markiere alle einsamen Knoten in V' als ungerade Knoten und setze $s(v) \leftarrow v$ für jeden einsamen Knoten $v \in V'$.
3. Wenn es einen ungeraden Knoten $v \in V'$ und einen nicht-markierten Knoten $u \in V'$ mit $\{v, u\} \in E' \setminus M'$ gibt, dann
 - a. markiere u als gerade und den Partner u' von u als ungerade,
 - b. merke dir v als Vorgänger von u und u als Vorgänger von u' ,
 - c. setze $s(u) \leftarrow s(v)$ und $s(u') \leftarrow s(v)$ und
 - d. wiederhole Schritt 3.
4. Wenn es zwei ungerade Knoten $v, u \in V'$ mit $\{v, u\} \in E' \setminus M'$ und $s(v) \neq s(u)$ gibt, dann ist der Weg von $s(v)$ zu $s(u)$ ein vergrößernder Pfad. Interpretiere diesen Weg unter Berücksichtigung von geschrumpften Blüten mittels Algorithmus 5.7. STOP.
5. Wenn es zwei ungerade Knoten $v, u \in V'$ mit $\{v, u\} \in E' \setminus M'$ und $s(v) = s(u)$ gibt, dann ist der Weg von $s(v)$ zu $s(u)$ eine Blume. Schrumpfe die Blüte, modifiziere G' und M' entsprechend und gehe zu Schritt 3.
6. Es gibt keinen vergrößernden Weg. STOP.

An dieser Stelle soll zunächst erwähnt werden, dass es effizientere Vorgehensweisen gibt, wir in diesem Buch aber eine möglichst einfache Variante betrachten, die der Idee von Edmonds (1965b) folgt.

Das Vorgehen in Algorithmus 5.8 orientiert sich an den drei oben besprochenen Fällen. Wir prüfen wiederholt, ob wir in Schritt 3., Schritt 4. oder Schritt 5. von einem ungeraden Knoten ausgehend einen geraden Knoten (und dessen Partner) markieren können (Schritt 3.) oder direkt andere ungerade Knoten erreichen können (Schritte 4. oder 5.). In Schritt 3. markieren wir einen bislang unmarkierten Knoten als gerade und dessen Partner als ungerade. Wir halten weiterhin die Informationen über den jeweiligen Vorgänger auf dem Pfad und den Ausgangsknoten des Pfades fest. In Schritt 4. wird ein vergrößernder Pfad für G' und M' gefunden und als vergrößernder Pfad für G und M interpretiert. Sobald wir einen vergrößernden Pfad für G und M gefunden haben, endet das Verfahren. Wenn wir in Schritt 5. eine Blume gefunden haben, schrumpfen wir die Blüte und reduzieren V' , E' und M' entsprechend. Wir fahren dann mit dem geschrumpften Graphen fort. Wenn keine Bedingung für die Ausführung eines der drei Schritte erfüllt ist, dann können wir keinen bisher gefundenen Pfad verlängern und es gibt keinen vergrößernden Pfad und das Verfahren endet. Dies ist nicht offensichtlich und wir sollten dafür im Folgenden noch argumentieren.

Beispiel

Zunächst wollen wir uns allerdings ein Beispiel anschauen. In den Abb. 5.48 bis 5.60 wird dargestellt, wie wir in zwei Schritten jeweils Algorithmus 5.8 anwenden, um ein gegebenes Matching um eine Kante zu vergrößern. Abb. 5.48 stellt den vollständigen Graph $G = (V, E)$ mit den schwarz hervorgehobenen Kanten von Matching M dar.

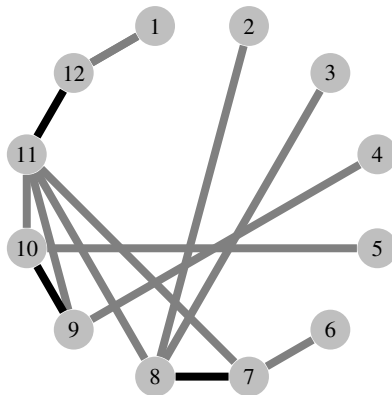


Abb. 5.48 Graph G und Matching M

Wir haben in Algorithmus 5.8 nicht festgelegt, in welcher Reihenfolge wir die Kanten des Graphen untersuchen. Der Ablauf ist daher nicht eindeutig. Initial werden alle einsamen Knoten, also die Knoten 1 bis 6, als ungerade markiert. Wir betrachten Kante $\{1, 12\} \notin M$ zuerst, stellen fest, dass Knoten 1 ungerade und Knoten 12 nicht markiert ist und markieren daher Knoten 12 als gerade und dessen Partner,

Knoten 11, als ungerade. Als nächstes betrachten wir Kante $\{10, 11\} \notin M$, stellen fest, dass Knoten 11 ungerade und Knoten 10 nicht markiert ist, und markieren daher Knoten 10 als gerade und dessen Partner, Knoten 9, als ungerade. Kante $\{9, 11\}$ verbindet nun zwei ungerade Knoten. Da $s(9) = s(11) = 1$, haben wir eine Blume gefunden. Die Blume besteht aus dem Stiel $(1, 12, 11)$ und der Blüte $\{9, 10, 11\}$, dargestellt in Abb. 5.49.

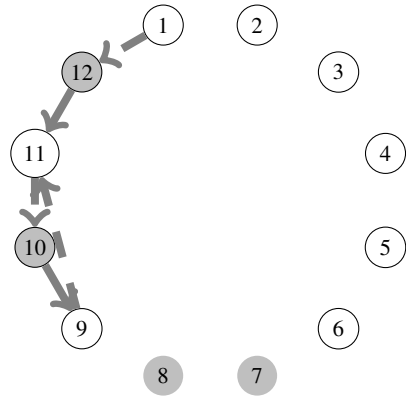


Abb. 5.49 Erste Blume

Wir schrumpfen die Blüte $\{9, 10, 11\}$ zu einem Knoten b . Die Kanten $\{4, 9\}$, $\{5, 10\}$, $\{7, 11\}$, $\{8, 11\}$ und $\{11, 12\}$ des Ausgangsgraphen werden durch die Kanten $\{4, b\}$, $\{5, b\}$, $\{7, b\}$, $\{8, b\}$ und $\{12, b\}$ ersetzt. Der geschrumpfte Graph G' und das entsprechende Matching $M' = \{\{7, 8\}, \{12, b\}\}$ sind in Abb. 5.50 dargestellt.

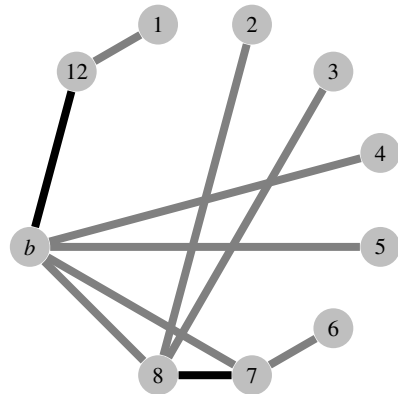


Abb. 5.50 Geschrumpfter Graph G' und Matching M'

Wir übernehmen die Markierungen der Knoten aus G in G' und markieren b als ungerade. Die einzige Kante in G , die auf einem Weg benutzt wurde und einen Knoten in der Blüte mit einem Knoten, der nicht in der Blüte ist, verbindet, ist die Kante $\{11, 12\}$. Diese wird durch die Kante $\{12, b\}$ zwischen geradem Knoten 12

und ungeradem Knoten b ersetzt. Alle Markierungen und Rollen der Kanten sind in Abb. 5.51 dargestellt.

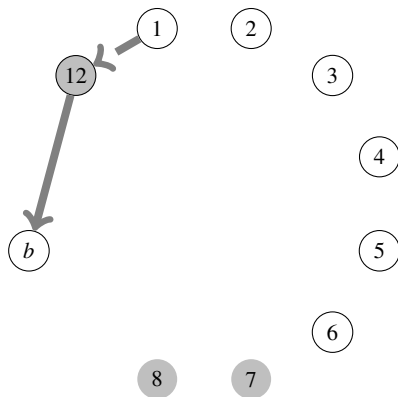


Abb. 5.51 Übernommene Markierungen und Pfade für G'

Wir fahren dann mit dem geschrumpften Graphen fort, betrachten die Kante $\{8, b\}$ und markieren Knoten 8 als gerade und Knoten 7 als ungerade. Über Kante $\{7, b\}$ finden wir eine weitere Blume, bestehend aus dem Stiel $(1, 12, b)$ und der Blüte $\{7, 8, b\}$, dargestellt in Abb. 5.52.

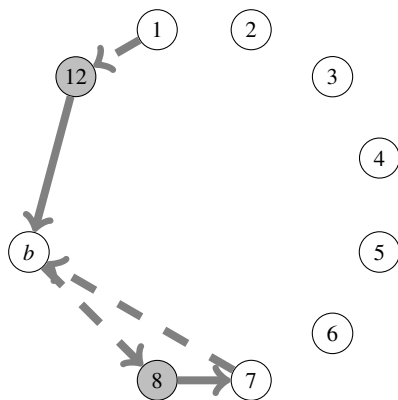


Abb. 5.52 Zweite Blume

Wir schrumpfen wiederum die Blüte, erhalten den neuen Knoten b' und markieren diesen als ungerade und erhalten Graph G'' , siehe Abb. 5.53. Das entsprechende Matching ist $M'' = \{\{12, b'\}\}$.

Wir übernehmen die Markierungen der Knoten aus G' in G'' und markieren b' als ungerade. Die einzige Kante in G , die auf einem Weg benutzt wurde und einen Knoten in der Blüte mit einem Knoten, der nicht in der Blüte ist, verbindet, ist die Kante $\{12, b\}$. Diese wird durch die Kante $\{12, b'\}$ zwischen geradem Knoten 12

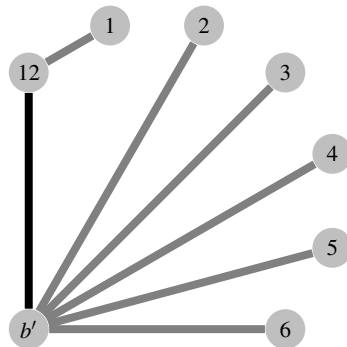


Abb. 5.53 Geschumpfter Graph G'' und Matching M''

und ungeradem Knoten b' ersetzt. Alle Markierungen und Rollen der Kanten sind in Abb. 5.54 dargestellt.

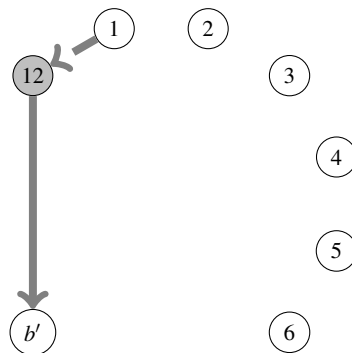


Abb. 5.54 Übernommene Markierungen und Pfade für G''

Wir können nun leicht einen vergrößernden Pfad finden. Es kommen mehrere in Frage; wir nehmen an, dass wir Kante $\{3, b'\}$ zuerst betrachten und feststellen, dass sowohl Knoten 3 als auch Knoten b' ungerade sind und dass zudem $1 = s(b') \neq s(3) = 3$ gilt. Wir haben somit einen vergrößernden Pfad $(1, 12, b', 3)$ für G'' und M'' gefunden, siehe Abb. 5.55, und müssen diesen nur noch als vergrößernden Pfad für G und M interpretieren.

Hierfür betrachten wir zunächst die geschumpfte Blüte $\{7, 8, b\}$, da diese zuletzt geschumpft wurde. Die Kante $\{3, b'\}$ ist nicht in M'' und entspricht der Kante $\{3, 8\}$ in G' (die nicht in M' ist). Wir haben also $v' = 3$ und $v'' = 8$. Der vergrößernde Pfad für G' und M' ergibt sich dann aus den beiden Teilpfaden von Knoten $s(v') = 3$ zu Knoten 8 und von Knoten 8 zu der Wurzel der Blume, also Knoten 1. Die erste Kante in dem zweiten Teilpfad muss eine Kante in M' sein (da $\{3, 8\} \notin M'$) und somit erhalten wir $(3, 8, 7, b, 12, 1)$ als vergrößernden Pfad für G' und M' . Diesen müssen wir wiederum als vergrößernden Pfad für G und M interpretieren.

Hierfür betrachten wir die geschumpfte Blüte $\{9, 10, 11\}$. Die Kante $\{7, b\}$ ist nicht in M' und entspricht der Kante $\{7, 11\}$ in G . Wir haben hier also $v' = 7$ und

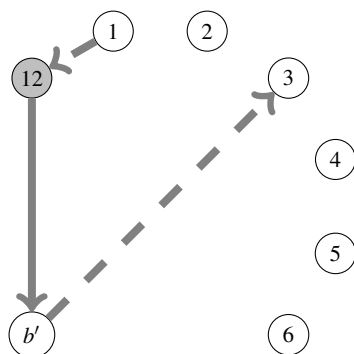


Abb. 5.55 Vergrößernder Pfad für G'' und M''

$v'' = 11$ und somit den speziellen Fall, in dem der einzige Knoten der Blüte, durch den der vergrößernde Pfad verläuft, der Ursprung der Blüte ist. Der vergrößernde Pfad für G und M ergibt sich dann aus den beiden Teilpfaden von Knoten 3 zu Knoten $v'' = 11$ und von Knoten 11 zu der Wurzel der Blume, also Knoten 1. Wir erhalten daher $(3, 8, 7, 11, 12, 1)$ als vergrößernden Pfad für G und M . Der vergrößernde Pfad für G und M ist in Abb. 5.56 dargestellt.

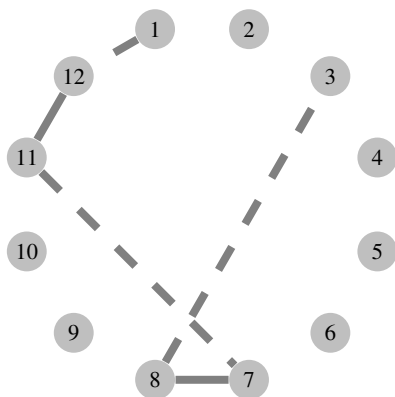
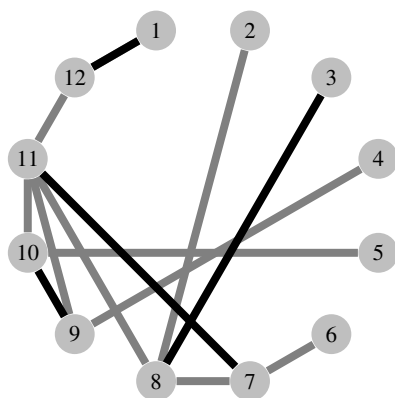
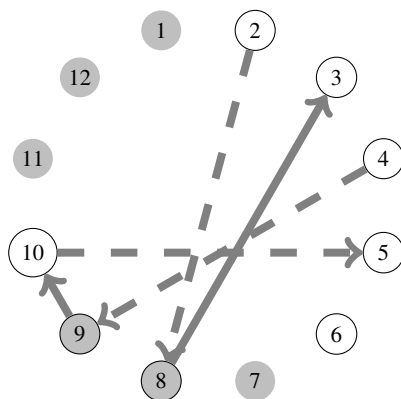


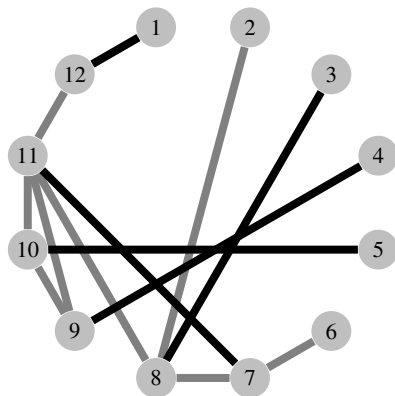
Abb. 5.56 Vergrößernder Pfad für G und M

Wir wenden diesen Pfad auf Matching M an und erhalten das Matching M''' in Abb. 5.57.

Wir nutzen erneut Algorithmus 5.8, um einen vergrößernden Pfad für G und M''' zu finden. Die einsamen Knoten 2 und 4 bis 6 werden initial als ungerade markiert. In diesem Durchlauf finden wir keine Blumen und kürzen die Beschreibung daher ab. Wir betrachten zunächst Kante $\{2, 8\}$ und markieren Knoten 8 als gerade und dessen Partner, Knoten 3, als ungerade. Als nächstes wird Kante $\{4, 9\}$ betrachtet und Knoten 9 als gerade und dessen Partner, Knoten 10, als ungerade markiert. Dann finden wir durch die Kante $\{5, 10\}$ den vergrößernden Pfad $(4, 9, 10, 5)$, siehe Abb. 5.58.

Abb. 5.57 Matching M''' Abb. 5.58 Gefundene Pfade
und Markierungen für G und
 M'''

Wir wenden diesen Pfad auf Matching M''' an und erhalten das Matching M^* in Abb. 5.59.

Abb. 5.59 Matching M^*

Wir nutzen erneut Algorithmus 5.8, um einen vergrößernden Pfad für G und M^* zu finden. Es wird allerdings keiner gefunden. Abb. 5.60 stellt die gefundenen Pfade und vorgenommenen Markierungen der Knoten dar.

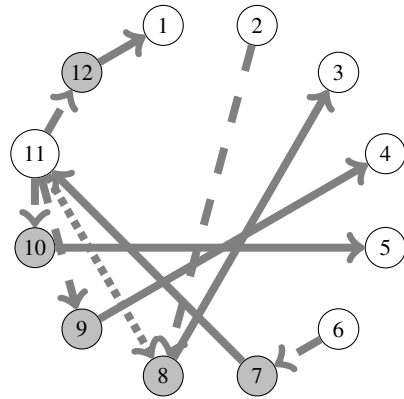


Abb. 5.60 Gefundene Pfade und Markierungen für G und M^*

Die gepunktete Kante $\{8, 11\}$ stellt dar, dass Knoten 8 auch von Knoten 11 (statt von Knoten 2 wie abgebildet) aus erreicht werden kann. An den Markierungen ändert dies nichts. Da es keinen vergrößernden Pfad für G und M^* gibt, hat Matching M^* gemäß Theorem 5.6 auf S. 315 maximale Kardinalität. In Graph G gibt es also kein perfektes Matching.

Analyse

An dieser Stelle ist noch offen, ob wir durch Anwendung von Algorithmus 5.8 Gewissheit haben, dass es tatsächlich keinen vergrößernden Pfad für G und M^* gibt. Das folgende Theorem besagt, dass Algorithmus 5.8 tatsächlich einen vergrößernden Pfad für den vorliegenden Graph und das vorliegende Matching findet, falls es einen gibt.

Theorem 5.10. *Wenn Algorithmus 5.8 keinen vergrößernden Pfad für Graph G und Matching M findet, dann gibt es keinen.*

Den Beweis führen wir in Anhang C.8. Theorem 5.10 liefert Gewissheit, dass Algorithmus 5.4 ein Matching maximaler Kardinalität liefert, wenn wir Algorithmus 5.8 in Schritt 2. verwenden, um zu prüfen, ob es einen vergrößernden Pfad für den vorliegenden Graph und das aktuelle Matching gibt, und ggf. einen zu finden.

5.3.2.2 Matchings maximalen Gewichts

Nun wollen wir uns Problem 5.4 (Matching maximalen Gewichts) zuwenden. Der Ansatz weist starke Analogien zu dem für bipartite Graphen in Abschnitt 5.3.1.2

auf. Wir repräsentieren Problem 5.4 (Matching maximalen Gewichts) in einem MO und entwickeln das entsprechende duale MO. Das Verfahren modifiziert dann sukzessiv eine primale zulässige Lösung und eine duale zulässige Lösung so, dass sie abschließend beide (für das jeweilige MO) optimal sind. Zunächst stellen wir das primale MO vor.

Modell 5.3 (Matching maximalen Gewichts (Problem 5.4)).

$$\max \sum_{\{v,u\} \in E, v < u} w_{v,u} \cdot x_{v,u} \quad (5.9)$$

$$\text{u.B.d.N.} \quad \sum_{\{v,u\} \in E, v < u} x_{v,u} + \sum_{\{u,v\} \in E, v > u} x_{u,v} \leq 1 \quad \forall v \in V \quad (5.10)$$

$$\sum_{\{v,u\} \in E, v,u \in V', v < u} x_{v,u} \leq \lfloor |V'|/2 \rfloor \quad \forall V' \subseteq V, |V'| \text{ ungerade} \quad (5.11)$$

$$x_{v,u} \geq 0 \quad \forall \{v,u\} \in E, v < u \quad (5.12)$$

Für jede Kante $\{v,u\} \in E, v < u$, wird eine Variable $x_{v,u}$ genutzt, die signalisiert, ob die Kante $\{v,u\}$ im Matching ist ($x_{v,u} = 1$) oder nicht ($x_{v,u} = 0$). Obwohl sie nicht auf binäre Werte beschränkt ist, gibt es immer optimale Lösungen, in denen diese Variable binäre Werte annimmt (dies ist nicht offensichtlich, es folgt aber aus dem im Folgenden vorgestellten Verfahren). NB (5.10) stellt sicher, dass eine Lösung einem Matching entspricht (bei binären Variablenwerten), während die Zielfunktion (5.9) das Ziel der Gewichtsmaximierung repräsentiert. NB (5.10) schließt allerdings bei kontinuierlichen Variablen nicht aus, dass ein Kreis aus einer ungeraden Anzahl von Kanten geschlossen wird und z. B. jede der entsprechenden Variablen einen Wert von 0,5 erhält. Offensichtlich repräsentiert eine solche Lösung kein Matching, aber diese Lösung kann besser sein (gemäß Zielfunktion) als jedes Matching. Dies erkennen wir einfach an einem Graph, der aus drei Knoten und drei Kanten (jeweils mit Gewicht 1) besteht. Kein Matching hat ein Gewicht von mehr als 1. Die Lösung zu Modell 5.3, in der alle Variablen einen Wert von 0,5 haben, hat aber ein gesamtes Gewicht von 1,5. NB (5.11) schließt diese Art von Lösungen aus. Daraus allein folgt nicht, dass Modell 5.3 Problem 5.4 (Matching maximalen Gewichts) akkurat repräsentiert. Wir werden dies zwar noch sehen, aber für den Moment wissen wir lediglich, dass Modell 5.3 Problem 5.4 (Matching maximalen Gewichts) oder ein relaxiertes Problem abbildet. Wir können nun das zu Modell 5.3 duale Modell 5.4 entwickeln.

Modell 5.4 (Duales Modell zu Modell 5.3).

$$\min \sum_{v \in V} \alpha_v + \sum_{V' \subseteq V, |V'| \text{ ungerade}} \left\lfloor \frac{|V'|}{2} \right\rfloor \cdot \gamma_{V'} \quad (5.13)$$

$$\begin{aligned} u.B.d.N. \quad & \alpha_v + \alpha_u + \\ & \sum_{V' \subseteq V, v, u \in V', |V'| \text{ ungerade}} \gamma_{V'} \geq w_{v,u} \quad \forall \{v, u\} \in E, v < u \end{aligned} \quad (5.14)$$

$$\alpha_v \geq 0 \quad \forall v \in V \quad (5.15)$$

$$\gamma_{V'} \geq 0 \quad \forall V' \subseteq V, |V'| \text{ ungerade} \quad (5.16)$$

Die Variablen α_v , $v \in V$, und $\gamma_{V'}$, $V' \subseteq V$ und $|V'|$ ungerade, sind die Dualvariablen, die zu den NB (5.10) und (5.11) von Modell 5.3 korrespondieren. Die Struktur von Modell 5.4 ergibt sich dann wie in Abschnitt 3.4 diskutiert.

Um die Notation ein wenig zu verkürzen, werden wir im Folgenden \mathcal{V} für die Menge von Teilmengen von V mit ungerader Kardinalität und $w_{v,u}$ für die linke Seite von (5.14) verwenden, d. h.

$$\begin{aligned} \mathcal{V} &= \{V' \mid V' \subseteq V, |V'| \text{ ungerade}\} \text{ und} \\ w_{v,u} &= \alpha_v + \alpha_u + \sum_{V' \in \mathcal{V}, v, u \in V'} \gamma_{V'}. \end{aligned}$$

In einer zulässigen dualen Lösung gilt demnach $w_{v,u} \geq w_{v,u}$ für jede Kante $\{v, u\}$, $v < u$.

Im Rahmen des folgenden Verfahrens werden eine primale zulässige Lösung (zu Modell 5.3) und eine duale zulässige Lösung (zu Modell 5.4) schrittweise so modifiziert, dass sie abschließend beide (für das jeweilige Modell) optimal sind. Wir werden wie in Abschnitt 5.3.1.2 Theorem 3.10 als Optimalitätskriterium heranziehen.

Wenn wir dieses Kriterium für die Modelle 5.3 und 5.4 konkretisieren, ergibt sich, dass zwei jeweils zulässige Lösungen \mathbf{x} und (α, γ) genau dann optimal sind, wenn

$$\begin{aligned} x_{v,u} (w_{v,u} - w_{v,u}) &= 0 \quad \text{für jedes } \{v, u\} \in E, v < u, \\ \alpha_v \left(1 - \sum_{\{v,u\} \in E, v < u} x_{v,u} - \sum_{\{u,v\} \in E, v > u} x_{u,v} \right) &= 0 \quad \text{für jedes } v \in V \text{ und} \\ \gamma_{V'} \left(\left\lfloor |V'|/2 \right\rfloor - \sum_{\{v,u\} \in E, v, u \in V', v < u} x_{v,u} \right) &= 0 \quad \text{für jedes } V' \in \mathcal{V}. \end{aligned}$$

Noch einmal wollen wir betonen, dass nach unserem bisherigen Kenntnisstand \mathbf{x} dabei nicht zwangsläufig ein Matching repräsentiert. Das heißt, wir können zunächst

nur als hinreichende Bedingung für Optimalität ableiten, dass ein Matching (die primale Lösung) und (α, γ) (die duale Lösung) optimal sind, wenn

1. für jede Kante $\{v, u\}$, $v < u$, im Matching $W_{v,u} = w_{v,u}$ gilt,
2. für jeden einsamen Knoten $v \in V$ $\alpha_v = 0$ gilt und
3. für jede Teilmenge $V' \in \mathcal{V}$ von Knoten, innerhalb derer weniger als $\lfloor |V'|/2 \rfloor$ Kanten des Matchings liegen, $\gamma_{V'} = 0$ gilt.

Nach Stand der Dinge können wir nicht ausschließen, dass jede optimale Lösung von Modell 5.3 kein Matching repräsentiert, weil es nicht-ganzzahlige Variablenwerte gibt. Dass es tatsächlich aber immer eine optimale Lösung zu Modell 5.3 gibt, die ein Matching repräsentiert, werden wir erst am Ende dieses Abschnittes erschließen können.

Wir bezeichnen eine Kante $\{v, u\}$, $v < u$, als *verfügbar* (für Matchings), wenn $W_{v,u} = w_{v,u}$. Im Laufe des Verfahrens werden Matchings nur unter Verwendung von verfügbaren Kanten konstruiert. Dadurch wird Bedingung 1. eingehalten.

Zusammenfassung: Matchings maximalen Gewichts in allg. Graphen

Wir ermitteln ein Matching maximalen Gewichts, indem wir optimale Lösungen für die zueinander dualen Modelle 5.3 und 5.4 ermitteln. Optimalität der vorliegenden Lösungen prüfen wir mittels Bedingungen 1. bis 3. Die erste dieser Bedingungen werden wir laufend einhalten, indem wir eine Kante $\{v, u\}$, $v < u$, nur dann in Matchings verwenden, wenn sie verfügbar ist, d. h. wenn $W_{v,u} = w_{v,u}$ gilt.

Überblick über das Verfahren

Das Verfahren startet mit einem Matching (und somit mit einer primalen zulässigen Lösung) und einer dualen zulässigen Lösung, die die Bedingungen 1. und 3. erfüllen. Wir werden dann die Lösungen so verändern, dass nach jedem Schritt die Bedingungen 1. und 3. erfüllt bleiben und früher oder später auch Bedingung 2. erfüllt wird. Wir starten dabei mit einem leeren Matching $M = \emptyset$ sowie $\alpha_v = \max\{w_{v',u'} \mid \{v', u'\} \in E, v' < u'\}/2$ für jedes $v \in V$ und $\gamma_{V'} = 0$ für jedes $V' \subseteq V$, $|V'|$ ungerade. Wir können recht leicht erkennen, dass beide Lösungen zulässig sind, denn jedes Matching ist primal zulässig und wir haben duale Zulässigkeit wegen $W_{v,u} = \max\{w_{v',u'} \mid \{v', u'\} \in E, v' < u'\} \geq w_{v,u}$ für jede Kante $\{v, u\}$ mit $v < u$. Zudem sind offensichtlich 1. und 3. erfüllt.

Das Verfahren läuft dann in höchstens $\lfloor |V|/2 \rfloor$ Phasen ab, die jeweils die Modifikation der primalen Lösung zur Folge haben oder zum Abbruch des Verfahrens führen. In einer Phase wird ein vergrößernder Pfad für Graph $G = (V, E')$ und das vorliegende Matching M gesucht, wobei E' die Menge der verfügbaren Kanten ist. Für die Suche nach einem solchen vergrößernden Pfad verwenden wir Algorithmus 5.8. Wenn wir einen vergrößernden Pfad finden, wenden wir ihn an und beenden damit die aktuelle Phase. Wenn wir keinen vergrößernden Pfad finden, modifi-

zieren wir die duale Lösung. Durch die Modifikation werden entweder weitere Kanten in E' aufgenommen, wodurch die bereits gefundenen Pfade verlängert werden können und ggf. letztlich ein vergrößernder Pfad gefunden wird, oder Bedingung 2. erfüllt, was zum Abbruch des Verfahrens führt, da die vorliegende primale und die vorliegende duale Lösung optimal sind.

Wenn die Phase mit einem vergrößernden Pfad endet, wenden wir diesen auf das vorliegende Matching an. Nach Anwendung des vergrößernden Pfades löschen wir die Markierung aller Knoten und geschrumpften Blüten. Die geschrumpften Blüten behalten wir allerdings bei. Wir behandeln diese in der nächsten Phase zunächst wie normale Knoten. Dadurch können geschrumpfte Blüten in späteren Phasen unmarkierte geschrumpfte Blüten oder als gerade markierte geschrumpfte Blüten sein. Dies ist in der Phase, in der sie geschrumpft werden, nicht möglich, da alle Blüten beim Schrumpfen als ungerade markiert werden. In jedem Fall werden wir im Folgenden Knoten, die direkt oder indirekt (durch in der Blüte enthaltene geschrumpfte Blüten) in einer unmarkierten, geraden oder ungeraden geschrumpften Blüte enthalten sind, als unmarkiert, gerade oder ungerade bezeichnen.

Nach höchstens $\lfloor |V|/2 \rfloor$ Phasen endet das Verfahren dann mit der Erfüllung von Bedingung 2., da die Kardinalität des initialen Matchings $M = \emptyset$ höchstens um $\lfloor |V|/2 \rfloor$ vergrößert werden kann. Was genau in einer Phase passiert und wie wir Blumen behandeln, wollen wir im Folgenden detaillieren.

Zusammenfassung: Verfahren für Matchings maximalen Gewichts in allgemeinen Graphen

Wir gehen von einem leeren Matching aus und versuchen einen vergrößernden Pfad aus verfügbaren Kanten zu ermitteln. Wenn wir einen finden, wenden wir ihn an und verkleinern so die Menge der einsamen Knoten. Wenn wir keinen finden, modifizieren wir die duale Lösung so, dass wir Optimalität beider Lösungen erkennen können oder weitere Kanten verfügbar werden und unsere Suche nach einem vergrößernden Pfad fortgesetzt werden kann.

Phasen

In jeder Phase wird ein vergrößernder Pfad aus verfügbaren Kanten für den vorliegenden (evt. schon geschrumpften) Graph und das vorliegende Matching ermittelt oder Bedingung 2. erfüllt, wodurch das gesamte Verfahren abbricht. Wir halten in jeder Phase eine Menge P von bereits gefundenen Pfaden vor, die nach Modifikation der dualen Lösung verlängert werden können. Die Menge von Pfaden wird also über eine Phase hinweg kontinuierlich weiterentwickelt. Am Anfang jeder Phase werden die in der vorherigen Phase gefundenen Pfade und Markierungen allerdings verworfen.

Die Suche nach einem vergrößernden Pfad erfolgt dann unter Modifikation des Graphen, nämlich durch Schrumpfen oder Interpretieren von Blüten, und Modi-

fikation der dualen Lösung. Für die Suche können wir Algorithmus 5.8 einsetzen, wobei wir in diesen das Modifizieren der dualen Lösung einbetten. Das Schrumpfen der Blüten verläuft dabei wie in Abschnitt 5.3.2.1 besprochen. Wenn V' die Knotenmenge der Blüte ist, initialisieren wir dabei zusätzlich die Dualvariable $\gamma_{V'}$. Die Modifikation der dualen Lösung wird immer dann angestoßen, wenn alle möglichen Pfade unter Verwendung der verfügbaren Kanten in E' entwickelt sind und sich unter diesen kein vergrößernder Pfad befindet.

Die Modifikation der dualen Lösung wird wie folgt vorgenommen. Zunächst bestimmen wir $\delta = \min \{\delta^V, \delta_1^E, \delta_2^E, \delta^B\}$ mit

$$\begin{aligned}\delta^V &= \min \{ \alpha_v \mid v \in V, v \text{ ist ungerade} \}, \\ \delta_1^E &= \min \{ W_{v,u} - w_{v,u} \mid \{v,u\} \in E \setminus E', v \text{ ist ungerade, } u \text{ ist nicht markiert} \}, \\ \delta_2^E &= \min \{ (W_{v,u} - w_{v,u}) / 2 \mid \{v,u\} \in E \setminus E', v \text{ und } u \text{ sind ungerade und} \\ &\quad \text{nicht in derselben Blüte} \} \text{ und} \\ \delta^B &= \min \{ \gamma_{V'} / 2 \mid V' \text{ ist eine gerade Blüte der obersten Ebene} \}.\end{aligned}$$

Die Werte von δ_1^E und δ_2^E stellen hier auf solche Kanten ab, die nicht in E' sind, die aber zur Verlängerung von Pfaden oder zum Finden einer Blüte genutzt werden könnten, wenn sie in E' wären. Dann modifizieren wir die duale Lösung, indem wir

- α_v für jeden ungeraden Knoten $v \in V$ um δ verringern,
- α_v für jeden geraden Knoten $v \in V$ um δ vergrößern,
- $\gamma_{|V'|}$ für jede ungerade geschrumpfte Blüte der obersten Ebene mit Knotenmenge V' um $2 \cdot \delta$ vergrößern und
- $\gamma_{|V'|}$ für jede gerade geschrumpfte Blüte der obersten Ebene mit Knotenmenge V' um $2 \cdot \delta$ verringern.

Alle übrigen Dualvariablen, die zu unmarkierten Knoten oder Blüten oder zu Blüten, die in anderen Blüten enthalten sind, gehören, behalten ihren bisherigen Wert. Wenn $\delta = \delta^B$, dann erhält die Dualvariable zu mindestens einer geraden geschrumpften Blüte den Wert null. Wir interpretieren solche Blüten dann, d. h., wir ersetzen den Knoten, der die Blüte repräsentiert, durch die Blüte mit den entsprechenden Kanten. Wir müssen außerdem Pfade, die durch diese geschrumpfte Blüte führten, durch die Kanten in der Blüte abbilden. Mit den Überlegungen aus Abschnitt 5.3.2.1 können wir erkennen, dass dies immer möglich ist. Knoten in der Blüte, die nicht durch einen gefundenen Pfad erreicht werden, sind unmarkiert. Nach Modifikation der dualen Variablen wird versucht, die bisher gefundenen Pfade zu verlängern oder eine Blüte zu finden (was möglich ist, wenn neue Kanten in E' aufgenommen wurden).

Zusammenfassung: Phasen

In einer Phase wird die duale Lösung (in womöglich mehreren Schritten) so modifiziert, dass entweder die Kardinalität des vorliegenden Matchings mittels eines vergrößernden Pfades erhöht wird oder Bedingung 2. (neben den Bedingungen 1. und 3.) erfüllt wird und somit das Verfahren abgebrochen werden kann, da das vorliegende Matching maximales Gewicht hat.

Pfade aus verfügbaren Kanten werden wie in Abschnitt 5.3.2.1 ermittelt. Wenn dabei Blumen gefunden werden, werden die Blüten geschrumpft. Wenn die Suche nach vergrößernden Pfaden stagniert, wird die duale Lösung so modifiziert, dass weitere Kanten verfügbar werden ($\delta = \min\{\delta_1^E, \delta_2^E\}$), eine Blüte interpretiert wird ($\delta = \delta^B$) oder Bedingung 2. erfüllt wird ($\delta = \delta^V$).

Das Verfahren lässt sich als Algorithmus 5.9 konkretisieren.

Algorithmus 5.9 (Matching maximalen Gewichts).

Eingabe: Graph $G = (V, E, w)$

Ausgabe: Matching maximalen Gewichts

1. Setze $M \leftarrow \emptyset$, $\alpha_v \leftarrow \max\{w_{v',u'} \mid \{v', u'\} \in E, v' < u'\}/2$ für jedes $v \in V$, $\gamma_{V'} \leftarrow 0$ für jedes $V' \in \mathcal{V}$, $M^g \leftarrow \emptyset$, $G^g \leftarrow G$ und Menge $E' \leftarrow \{\{v, u\} \mid \{v, u\} \in E, w_{v,u} = w_{v,u}\}$ der verfügbaren Kanten.
2. Lösche Markierungen von Knoten und geschrumpften Blüten und setze $P \leftarrow \emptyset$.
3. Falls $\alpha_v = 0$ für jeden (gemäß M) einsamen Knoten $v \in V$, STOP.
4. Verlängere die Pfade in P mit Kanten in G^g , die Kanten in E' entsprechen, zu einem vergrößernden Pfad für G^g und M^g oder zu einer Blume, falls möglich.
5. Falls in 4. ein vergrößernder Pfad für G^g und M^g gefunden wurde, wende ihn auf M^g an, aktualisiere M entsprechend und gehe zu 2.
6. Falls in 4. eine Blume gefunden wurde, schrumpfe die Blüte, initialisiere die entsprechende Dualvariable, aktualisiere G^g , M^g und P entsprechend und gehe zu 4.
7. Ansonsten modifiziere die duale Lösung
 - $\delta^V \leftarrow \min\{\alpha_v \mid v \in V, v \text{ ist ungerade}\}$ ($\delta^V \leftarrow \infty$, wenn es keinen ungeraden Knoten gibt)
 - $\delta_1^E \leftarrow \min\{w_{v,u} - w_{v,u} \mid \{v, u\} \in E \setminus E', v < u \text{ ist ungerade, } u \text{ ist nicht markiert (oder umgekehrt)}\}$ ($\delta_1^E \leftarrow \infty$, wenn es keine solche Kante gibt)
 - $\delta_2^E \leftarrow \min\{(w_{v,u} - w_{v,u})/2 \mid \{v, u\} \in E \setminus E', v < u \text{ und } u \text{ sind ungerade und nicht in derselben Blüte}\}$ ($\delta_2^E \leftarrow \infty$, wenn es keine solche Kante gibt)
 - $\delta^B \leftarrow \min\{\gamma_{V'}/2 \mid V' \text{ ist eine gerade Blüte der obersten Ebene}\}$ ($\delta^B \leftarrow \infty$, wenn es keine gerade Blüte der obersten Ebene gibt)
 - $\delta \leftarrow \min\{\delta^V, \delta_1^E, \delta_2^E, \delta^B\}$
 - $\alpha_v \leftarrow \alpha - \delta$ für jeden ungeraden Knoten $v \in V$
 - $\alpha_v \leftarrow \alpha + \delta$ für jeden geraden Knoten $v \in V$
 - $\gamma_{|V'|} \leftarrow \gamma_{|V'|} + 2 \cdot \delta$ für jede ungerade geschrumpfte Blüte der obersten Ebene mit Knotenmenge V'

- $\gamma_{|V'|} \leftarrow \gamma_{|V'|} - 2 \cdot \delta$ für jede gerade geschrumpfte Blüte der obersten Ebene mit Knotenmenge V'
- Falls $\delta = \delta^B$, interpretiere jede gerade geschrumpfte Blüte der obersten Ebene mit Knotenmenge V' und $\gamma_{|V'|} = 0$ und aktualisiere G' , M' und P entsprechend.
- Aktualisiere E' und gehe zu 3.

Graph G^g und Matching M^g stellen den geschrumpften Graph und das entsprechend geschrumpfte Matching dar. In Schritt 1. werden einmalige Initialisierungen vorgenommen. Die Initialisierung zu Beginn jeder Phase wird in Schritt 2. ausgeführt. Hier werden zwar Markierungen gelöscht, aber geschrumpfte Blüten beibehalten. Schritt 3. prüft, ob das Abbruchkriterium erfüllt ist und somit mit M ein Matching maximalen Gewichts vorliegt. In Schritt 4. werden die bisher gefundenen Pfade in P verlängert, um Pfade bzw. Blumen zu finden. Dies kann wie in Algorithmus 5.8 gestaltet werden. Sollte ein vergrößernder Pfad für G^g und M^g gefunden werden, wird dieser angewendet und die nächste Phase angestoßen. Falls eine Blume gefunden wurde, wird die Blüte geschrumpft und in dem modifizierten Graph erneut nach Pfaden gesucht. Da in diesem Fall weder die primale noch die duale Lösung modifiziert wurde, muss das Abbruchkriterium nicht geprüft werden. Wurden weder vergrößernder Pfad noch Blume gefunden, dann wird die duale Lösung modifiziert. Ggf. werden dabei zuvor geschrumpfte Blüten interpretiert.

Beispiel

Bevor wir das Verfahren analysieren, wollen wir es an einem Beispiel veranschaulichen. Wir wenden Algorithmus 5.9 auf die Problem Instanz in Abb. 5.61 an.

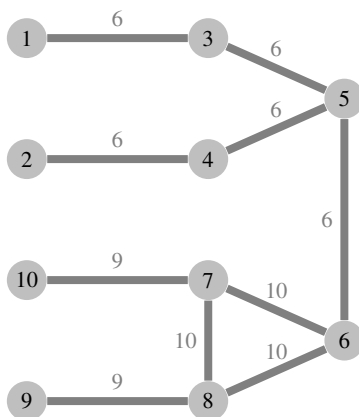


Abb. 5.61 Instanz von Problem 5.4 (Matching maximalen Gewichts)

Tabelle 5.11 stellt den Ablauf des Verfahrens dar. Die jeweiligen Teilgraphen, die gefundene Matchings und Pfade bilden, sind in Abb. 5.62 bis Abb. 5.77 illustriert.

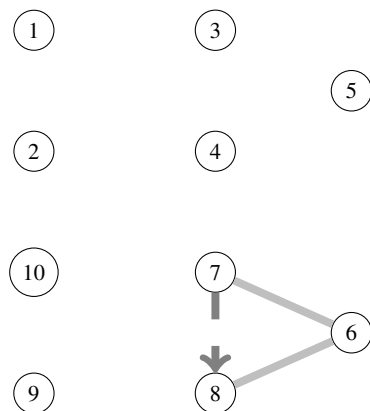
Tabelle 5.11 Ablauf von Algorithmus 5.9 für die Instanz in Abb. 5.61

	M	α										γ	E'	Abb.	Resultat
		1	2	3	4	5	6	7	8	9	10				
1	\emptyset	5	5	5	5	5	5	5	5	5	5		$\{\{6,7\}, \{6,8\}, \{7,8\}\}$	5.62, 5.63	Pfad (7, 8)
2	$\{\{7,8\}\}$	5	5	5	5	5	5	5	5	5	5		$\{\{6,7\}, \{6,8\}, \{7,8\}\}$	5.64, 5.65	Blüte $V' = \{6, 7, 8\}$
3	$\{\{7,8\}\}$	5	5	5	5	5	5	5	5	5	5	$\gamma_{\{6,7,8\}} = 0$	$\{\{6,7\}, \{6,8\}, \{7,8\}\}$	5.66	$\delta = \min\{5, \infty, \frac{3}{2}, \infty\}$
4	$\{\{7,8\}\}$	9	9	9	9	9	9	9	9	9	9	$\gamma_{\{6,7,8\}} = 1$	$\{\{6,7\}, \{6,8\}, \{7,8\}, \{7,10\}, \{8,9\}\}$	5.67, 5.68	Pfad (9, b)
5	$\{\{6,7\}, \{8,9\}\}$	9	9	9	9	9	9	9	9	9	9	$\gamma_{\{6,7,8\}} = 1$	$\{\{6,7\}, \{6,8\}, \{7,8\}, \{7,10\}, \{8,9\}\}$	5.69, 5.70	$\delta = \min\{\frac{9}{2}, \infty, \frac{3}{2}, \frac{1}{2}\}$
6	$\{\{6,7\}, \{8,9\}\}$	9	9	9	9	9	9	9	9	9	9	$\gamma_{\{6,7,8\}} = 0$	$\{\{6,7\}, \{6,8\}, \{7,8\}, \{7,10\}, \{8,9\}\}$	5.70	$\delta = \min\{4, \infty, 1, \infty\}$
7	$\{\{6,7\}, \{8,9\}\}$	4	4	4	4	4	5	5	5	4	4		$\{\{1,3\}, \{2,4\}, \{3,5\}, \{4,5\}, \{6,7\}, \{6,8\}, \{7,10\}, \{8,9\}\}$	5.71, 5.72	Pfad (3, 5)
8	$\{\{3,5\}, \{6,7\}, \{8,9\}\}$	3	3	3	3	3	4	6	6	3	3		$\{\{1,3\}, \{2,4\}, \{3,5\}, \{4,5\}, \{6,7\}, \{6,8\}, \{7,10\}, \{8,9\}\}$	5.73, 5.74	Pfad (2, 4)
9	$\{\{2,4\}, \{3,5\}, \{6,7\}, \{8,9\}\}$	3	3	3	3	3	4	6	6	3	3		$\{\{1,3\}, \{2,4\}, \{3,5\}, \{4,5\}, \{6,7\}, \{6,8\}, \{7,10\}, \{8,9\}\}$	5.75	$\delta = \min\{3, \infty, \frac{1}{2}, \infty\}$
10	$\{\{2,4\}, \{3,5\}, \{6,7\}, \{8,9\}\}$	5	5	7	7	5	7	13	13	5	5		$\{\{1,3\}, \{2,4\}, \{3,5\}, \{4,5\}, \{5,6\}, \{6,7\}, \{6,8\}, \{7,10\}, \{8,9\}\}$	5.76, 5.77	Pfad (1, 3, 5, 6, 7, 10)
	$\{\{1,3\}, \{2,4\}, \{5,6\}, \{7,10\}, \{8,9\}\}$	5	5	7	7	5	7	13	13	5	5				
		5	5	7	7	5	7	13	13	5	5				

In Tabelle 5.11 korrespondiert jede der nummerierten Zeilen zu einer Ausführung von Schritt 4. in Algorithmus 5.9 mit der daraus resultierende Modifikation in Schritt 5., 6. oder 7. In der letzten Zeile sind die finale primale und die finale duale Lösung abgetragen. Wir wollen nun nachvollziehen, was in den einzelnen Ausführungen von Schritt 4. und den resultierenden Modifikationen des Graphen oder einer der Lösungen geschieht. Wir initialisieren zunächst die primale und duale Lösung und erhalten ein leeres Matching und Werte für die Dualvariablen, wie in Zeile 1 eingetragen. Wir tragen dabei in der Spalte mit Titel γ explizit ab, auf welche Knotenmenge V' sich der jeweilige Wert bezieht. Die Werte der Variablen, die sich auf nicht genannte Mengen beziehen, sind null. Die Zielfunktionswerte der primalen und der dualen Lösung sind null und 50. Drei Kanten sind verfügbar. In Abb. 5.62 sind markierte Knoten mit einem schwarzen Rand dargestellt. Dabei sind ungerade Knoten weiß und gerade Knoten grau. Alle übrigen Knoten sind ohne Rand und in grau dargestellt. Alle Kanten in E' , die in keinem gefundenen Pfad sind, sind in hellgrau dargestellt. Kanten in E' , die in einem gefundenen Pfad, aber nicht im vorliegenden Matching sind, sind wie schon zuvor in dunkelgrau gestrichelt dargestellt. Kanten in E' , die in einem gefundenen Pfad und im vorliegenden Matching sind, sind wie gehabt in dunkelgrau dargestellt. Nach wie vor ist nicht festgelegt, in welcher Reihenfolge wir die Kanten des Graphen untersuchen. Der Ablauf ist daher nicht eindeutig.

Bei der ersten Ausführung von Schritt 4. sind die Kanten $\{6, 7\}$, $\{6, 8\}$ und $\{7, 8\}$ verfügbar. Wir markieren initial alle Knoten als ungerade, da alle Knoten einsam sind. Die erste betrachtete Kante $\{7, 8\}$ (zufällig unter den Kanten in E' gewählt) stellt direkt einen vergrößерnden Pfad dar, siehe Abb. 5.62.

Abb. 5.62 Vergrößерnder Pfad nach erster Ausführung von Schritt 4.



Die Kante $\{7, 8\}$ wird daher in das Matching aufgenommen und wir erhalten das Matching in Abb. 5.63. Die duale Lösung wird nicht modifiziert. Dies markiert das Ende der ersten Phase und alle Markierungen von Knoten werden entfernt.

Bei der zweiten Ausführung von Schritt 4. werden initial alle Knoten außer 7 und 8 als ungerade markiert. Die erste betrachtete Kante $\{6, 7\}$ (zufällig unter den

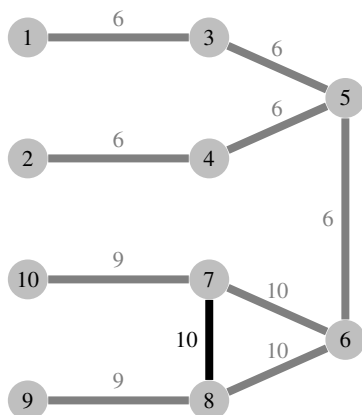


Abb. 5.63 Matching nach erster Ausführung von Schritt 4. (erste Phase)

Kanten in E' gewählt) führt zur Markierung von 7 und 8 (als Partner von 7). Die verbleibende Kante in E' schließt dann eine Blüte, siehe Abb. 5.64.

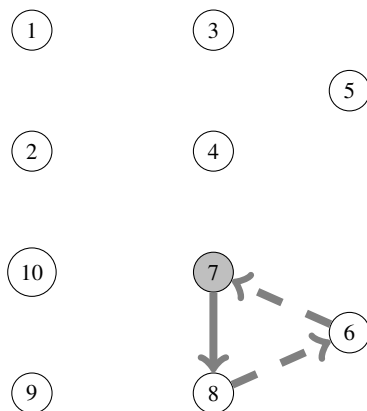


Abb. 5.64 Blüte nach zweiter Ausführung von Schritt 4.

Die Blüte wird geschrumpft, wodurch wir den Graph G^g in Abb. 5.65 erhalten. Zu der geschrumpften Blüte initialisieren wir $\gamma_{\{6,7,8\}} = 0$. Die Kanten $\{5, b\}$, $\{9, b\}$ und $\{10, b\}$ in G^g korrespondieren zu den Kanten $\{5, 6\}$, $\{8, 9\}$ und $\{7, 10\}$ in G und haben daher auch die entsprechenden Gewichte. Die Entsprechung des vorliegenden Matchings M in G^g ist leer, da die einzige Kante in M in der geschrumpften Blüte ist.

In diesem Graph G^g gibt es keine verfügbare Kante, da alle verfügbaren Kanten in der geschrumpften Blüte sind, und deshalb werden zwar bei der dritten Ausführung von Schritt 4. alle Knoten als ungerade markiert, aber kein Pfad wird gefunden, siehe Abb. 5.66.

Nun ermitteln wir δ^V , δ_1^E , δ_2^E , δ^B und abschließend δ . Wir haben $\alpha_v = 5$ für jeden Knoten $v \in V$. Dies gilt insbesondere für die ungeraden. Es gibt in E kei-

Abb. 5.65 Geschrumpfter Graph und geschrumpftes Matching

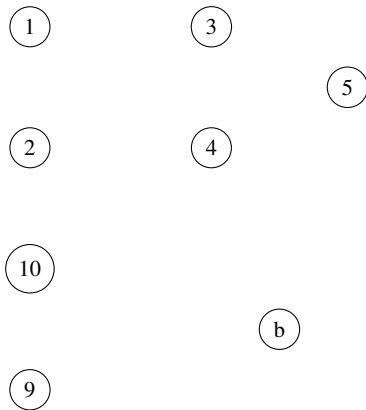
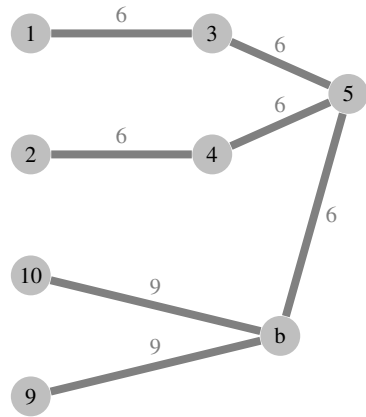


Abb. 5.66 Pfade nach dritter Ausführung von Schritt 4.

ne Kante von einem ungeraden Knoten $v \in V'$ zu einem unmarkierten Knoten $u \in V'$, daher erhalten wir $\delta_1^E = \infty$. Da alle Knoten in G^s gemäß der aktuellen Markierungen ungerade sind, wird jede Kante, die eine Entsprechung in G^s hat, bei der Bestimmung von δ_2^E berücksichtigt. Davon minimieren Kanten $\{7, 10\}$ und $\{8, 9\}$ den Term $W_{v,u} - w_{v,u}$ und wir erhalten $\delta_2^E = 1/2$. Da es keine geraden geschrumpften Blüten gibt, gilt $\delta^B = \infty$ und wir erhalten $\delta = \min\{\delta^V, \delta_1^E, \delta_2^E, \delta^B\} = \min\{5, \infty, 1/2, \infty\} = 1/2$. Wir nehmen nun eine Anpassung der dualen Lösung vor, indem wir die dualen Variablen α_1 bis α_{10} um $\delta = 1/2$ reduzieren und $\gamma_{\{6,7,8\}}$ um $2 \cdot \delta = 1$ erhöhen. Wir sehen, dass die beiden Kanten $\{7, 10\}$ und $\{8, 9\}$, die den Wert von δ bestimmt haben, in E' aufgenommen werden. Weiterhin sehen wir, dass für jede Kante $\{v, u\}$ in der Blüte der Wert von $W_{v,u}$ unverändert bleibt, denn $(\alpha_v - \delta) + (\alpha_u - \delta) + (\gamma_{\{6,7,8\}} + 2 \cdot \delta) = \alpha_v + \alpha_u + \gamma_{\{6,7,8\}}$ (wir erinnern uns, dass wir jeden Knoten in einer ungeraden geschrumpften Blüte als ungeraden Knoten ansehen).

Wir starten dann die vierte Ausführung von Schritt 4. Die erste betrachtete Kante $\{9, b\}$ (wiederum zufällig gewählt) stellt direkt einen vergrößernden Pfad dar, siehe Abb. 5.67.

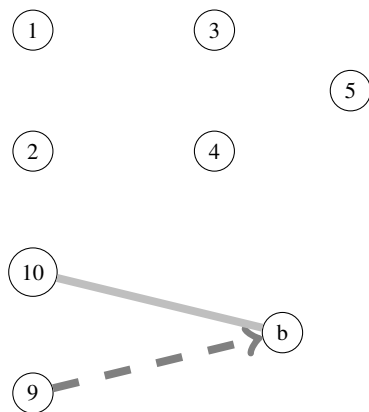


Abb. 5.67 Vergrößernder Pfad nach vierter Ausführung von Schritt 4.

Wir wenden Pfad $(9, b)$ auf G^g und M^g an und erhalten das Matching in Abb. 5.68. Wir können dabei im Hinterkopf behalten (und tragen dies auch in Tabelle 5.11 ein), dass dieses Matching zu dem Matching $M = \{\{6, 7\}, \{8, 9\}\}$ in G korrespondiert (wir können den vergrößernden Pfad $(9, b)$ für G^g und M^g wie aus Abschnitt 5.3.2.1 bekannt in den vergrößernden Pfad $(9, 8, 7, 6)$ für G und M übersetzen). Die duale Lösung wird nicht modifiziert. Dies markiert das Ende der zweiten Phase und alle Markierungen von Knoten werden entfernt.

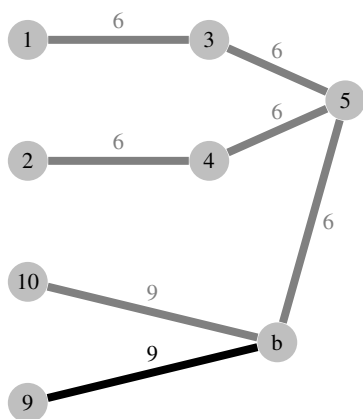


Abb. 5.68 Matching nach vierter Ausführung von Schritt 4. (zweite Phase)

Bei der fünften Ausführung von Schritt 4. finden wir nur den Pfad $(10, b, 9)$, der kein vergrößernder Pfad für G^g und M^g ist, siehe Abb. 5.69.

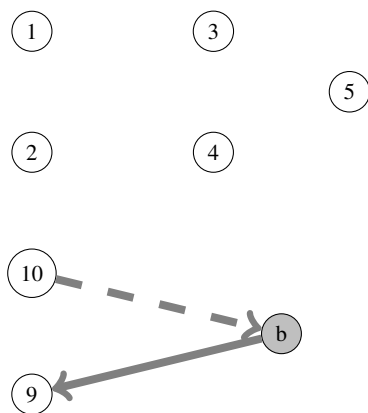


Abb. 5.69 Pfade nach fünfter Ausführung von Schritt 4.

Nun ermitteln wir δ^V , δ_1^E , δ_2^E , δ^B und abschließend δ . Es ergeben sich $\delta^V = 9/2$, $\delta_1^E = \infty$ (es gibt keinen unmarkierten Knoten), $\delta_2^E = 3/2$ (z. B. durch Kante $\{1, 3\}$), $\delta^B = \gamma_{\{6,7,8\}}/2 = 1/2$ und daher $\delta = 1/2$. Wir nehmen nun eine Anpassung der dualen Lösung vor, indem wir die dualen Variablen α_1 bis α_5 , α_9 und α_{10} um $\delta = 1/2$ reduzieren, α_6 bis α_8 um $\delta = 1/2$ erhöhen (wir erinnern uns, dass wir jeden Knoten in einer geraden geschrumpften Blüte als geraden Knoten ansehen) und $\gamma_{\{6,7,8\}}$ um $2 \cdot \delta = 1$ reduzieren. Keine Kante wird in E' aufgenommen, da $\min\{\delta_1^E, \delta_2^E\} > \delta^B$. Stattdessen wird die einzige Blüte wieder interpretiert, da $\delta = \delta^B$, und wir erhalten den Graph G^g , der dieselbe Struktur wie G hat, in Abb. 5.70. Den Pfad $(10, b, 9)$ übersetzen wir mit ähnlichen Überlegungen wie in Abschnitt 5.3.2.1 in den Pfad $(10, 7, 6, 8, 9)$ (da b eine gerade Blüte ist, muss die erste Kante im Pfad nach $\{7, 10\}$ im Matching sein) und markieren die Knoten auf diesem Pfad entsprechend. Ggf. vorher gesetzte Markierungen der übrigen Knoten in der Blüte würden gelöscht, hier gibt es aber keine Knoten in der Blüte, die nicht auch im konstruierten Pfad liegen.

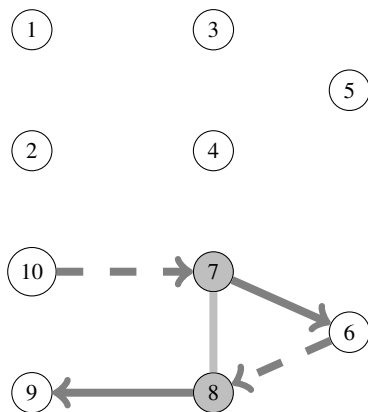


Abb. 5.70 Graph mit interpretierter Blüte

Wir wollen kurz auf Kante $\{7,8\}$ genauer eingehen. Diese Kante war vor Schrumpfen der Blüte im Matching. Dies ist nach Interpretation der Blüte nicht mehr der Fall, da durch Anwendung von dem vergrößernden Pfad $(9,b)$ für G^g und M^g nach der vierten Ausführung von Schritt 4. die Kanten $\{8,9\}$ und $\{6,7\}$ in das Matching aufgenommen und die Kante $\{7,8\}$ aus dem Matching entfernt wurde.

Da keine Kante in E' aufgenommen wurde, können wir die bisher gefundenen Pfade nicht durch die sechste Ausführung von Schritt 4. verlängern und bestimmen erneut $\alpha_v = 4$, $\delta_1^E = \infty$ (es gibt keinen unmarkierten Knoten), $\delta_2^E = 1$ (z. B. durch Kante $\{1,3\}$), $\delta^B = \infty$ (es gibt keine geschrumpften Blüten) und daher $\delta = 1$. Durch die Modifikation der dualen Lösung werden vier weitere Kanten in E' aufgenommen und Kante $\{7,8\}$ aus E' entnommen. Durch die siebte und achte Ausführung von Schritt 4. werden die vergrößernden Pfade $(3,5)$ (Abb. 5.71) und $(2,4)$ (Abb. 5.73) gefunden und das Matching in zwei Schritten (Abb. 5.72 und 5.74) erweitert. Die beiden Schritte markieren das Ende der dritten Phase und der vierten Phase.

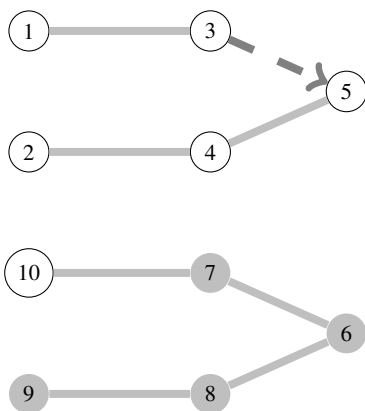


Abb. 5.71 Vergrößernder Pfad nach siebter Ausführung von Schritt 4.

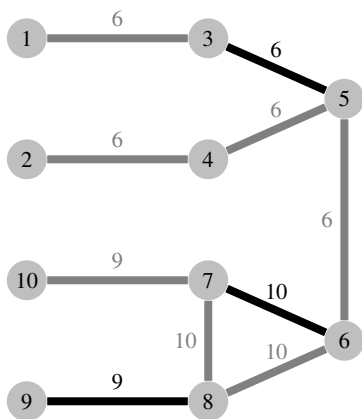


Abb. 5.72 Matching nach siebter Ausführung von Schritt 4. (dritte Phase)

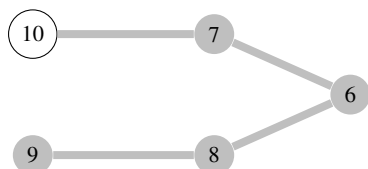
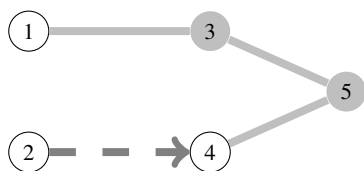


Abb. 5.73 Vergrößernder Pfad nach achter Ausführung von Schritt 4.

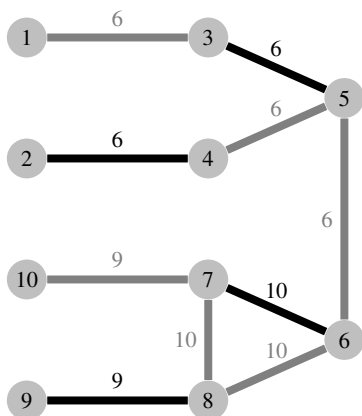


Abb. 5.74 Matching nach achter Ausführung von Schritt 4. (vierte Phase)

Durch die neunte Ausführung von Schritt 4. werden die Pfade $(1, 3, 5, 4, 2)$ und $(10, 7, 6, 8, 9)$, siehe Abb. 5.75, gefunden, die aber beide keine vergrößernden Pfade sind.

Wir bestimmen $\alpha^V = 3$, $\delta_1^E = \infty$, $\delta_2^E = 1/2$ (durch Kante $\{5, 6\}$), $\delta^B = \infty$ und daher $\delta = 1/2$. Wir nehmen die Anpassung der dualen Lösung vor, wodurch Kante $\{5, 6\}$ in E' aufgenommen wird.

Durch die zehnte Ausführung von Schritt 4. finden wir den vergrößernden Pfad $(1, 3, 5, 6, 7, 10)$, siehe Abb. 5.76. Durch Anwendung dieses Pfades erhalten wir das Matching in Abb. 5.77. Der Schritt markiert das Ende der fünften Phase.

Da kein Knoten mehr einsam ist, ist Bedingung 2. erfüllt und das Verfahren endet mit einer optimalen primalen Lösung und einer optimalen dualen Lösung, die jeweils den Zielfunktionswert 36 haben.

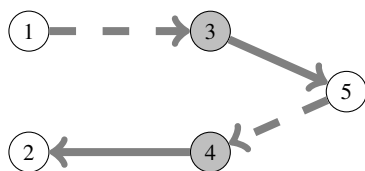


Abb. 5.75 Pfade nach neunter Ausführung von Schritt 4.

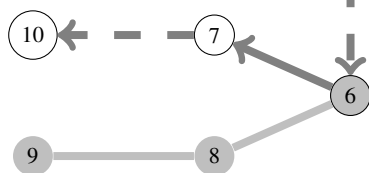
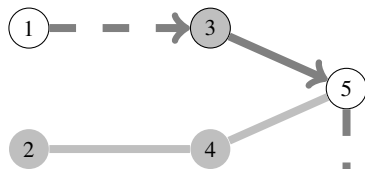
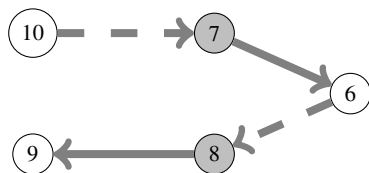


Abb. 5.76 Vergrößernder Pfad nach zehnter Ausführung von Schritt 4.

Analyse

Wir werden im Folgenden dafür argumentieren, dass die jeweiligen Lösungen zulässig sind, dass die Bedingungen 1. und 3. während des gesamten Ablaufs durch die jeweils vorliegenden Lösungen erfüllt sind und dass nach einer endlichen Anzahl von Durchläufen von Schritt 4. eine optimale Lösung ermittelt wird.

Jede betrachtete primale Lösung ist ein Matching und daher primal zulässig. Die duale Lösung bleibt durch die jeweilige Modifikation ebenfalls zulässig. Es ist relativ leicht zu erkennen, dass durch die Wahl von δ keine der Dualvariablen negativ werden kann. Um zu erkennen, dass darüber hinaus $W_{v,u} \geq w_{v,u}$ für jede Kante $\{v,u\}$ gilt, wollen wir uns die Veränderung von $W_{v,u}$ für jede Kante $\{v,u\}$ etwas genauer anschauen.

- Der Wert $W_{v,u}$ ändert sich für eine Kante $\{v,u\}$ in einer geschrumpften Blüte nicht. Die Dualvariablen α_v und α_u werden um jeweils δ vergrößert (verringert), wenn die Blüte – und somit auch die enthaltenen Knoten – gerade (ungerade) sind, während Dualvariable $\gamma_{|V'|}$ zur Blüte der obersten Ebene mit Knotenmenge

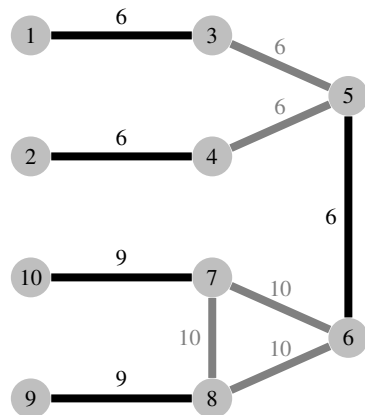


Abb. 5.77 Matching nach zehnter Ausführung von Schritt 4. (fünfte Phase)

V' , $\{v, u\} \subset V'$, (und nur für diese) um $2 \cdot \delta$ verringert (vergrößert) wird. Wenn die Blüte unmarkiert ist, dann ändert sich keine der Dualvariablen α_v , α_u und $\gamma_{V'}$.

- Für eine Kante $\{v, u\}$, die einen geraden Knoten mit einem ungeraden Knoten verbindet, verändert sich der Wert von $W_{v,u}$ nicht, denn die Veränderungen von α_v und α_u heben sich auf und keine Variable $\gamma_{V'}$ mit $\{v, u\} \subset V'$ wird verändert, da $\{v, u\}$ in keiner Blüte liegt (ansonsten hätten v und u identische Markierungen).
- Für eine Kante $\{v, u\}$, die einen geraden Knoten mit einem geraden oder unmarkierten Knoten verbindet (und die nicht in einer Blüte liegt), vergrößert sich der Wert von $W_{v,u}$ um $2 \cdot \delta$ oder δ .
- Für eine Kante $\{v, u\}$, die einen ungeraden Knoten mit einem ungeraden oder unmarkierten Knoten verbindet (und die nicht in einer Blüte liegt), verringert sich der Wert von $W_{v,u}$ um $2 \cdot \delta$ oder δ . Aufgrund der Wahl von δ_1^E und δ_2^E kann der Wert aber nicht kleiner als $w_{v,u}$ werden.

Zusammenfassung: Zulässige primale und duale Lösungen

Die initialen primalen und dualen Lösungen sind zulässig. Durch die Modifikationen werden die primalen und dualen Lösungen nicht unzulässig, falls sie vorher zulässig waren. Die jeweils vorliegenden primalen und dualen Lösungen sind also während des gesamten Verfahrens zulässig.

Darüber hinaus können wir feststellen, dass die Bedingungen 1. und 3. nicht verletzt werden (wenn sie nicht schon vorher verletzt waren). Dies können wir wie folgt erschließen.

- Der Wert $W_{v,u}$ einer Kante $\{v, u\}$ im Matching ändert sich nicht. Um dies zu sehen, unterscheiden wir drei Fälle.
 - Wenn Kante $\{v, u\}$ in keiner geschrumpften Blüte liegt und mindestens einer der Knoten v und u markiert ist, dann sind beide Knoten markiert und zwar

unterschiedlich, denn der Partner eines markierten Knotens wird unmittelbar anders markiert. Die Veränderungen von α_v und α_u heben sich auf und keine Variable $\gamma_{v'}$ mit $\{v, u\} \subset V'$ wird verändert.

- Wenn Kante $\{v, u\}$ in keiner geschrumpften Blüte ist und keiner der Knoten v und u markiert ist, dann ändern sich weder α_v noch α_u und keine Variable $\gamma_{v'}$ mit $\{v, u\} \subset V'$ wird verändert.
- Wenn Kante $\{v, u\}$ in einer geraden (ungeraden/unmarkierten) geschrumpften Blüte ist, dann sind die Knoten v und u auch gerade (ungerade/unmarkiert). Also heben sich die Änderungen von α_v , α_u und $\gamma_{v'}$, mit $\{v, u\} \subset V'$ und V' ist eine Blüte der obersten Ebene, auf. Es sollte hier betont werden, dass es höchstens eine Blüte V' der obersten Ebene geben kann, in der v und u enthalten sind.

Daher gilt für jede Kante $\{v, u\}$ im Matching nach der Modifikation der dualen Lösung $W_{v,u} = w_{v,u}$, wenn es vorher galt.

- Umgekehrt werden nur verfügbare Kanten in ein Matching aufgenommen. Daher gilt für jede Kante $\{v, u\}$ im Matching nach der Modifikation der primalen Lösung $W_{v,u} = w_{v,u}$, wenn es vorher galt.
- Der Wert der Dualvariablen $\gamma_{|V'|}$ wird nur erhöht, wenn V' die Knotenmenge einer ungeraden geschrumpften Blüte ist. In jeder Blüte liegen $\lfloor |V'|/2 \rfloor$ Kanten des Matchings. Daher gilt für jede Teilmenge $V' \in \mathcal{V}$, innerhalb derer weniger als $\lfloor |V'|/2 \rfloor$ Kanten des Matchings liegen, nach der Modifikation der dualen Lösung $\gamma_{V'} = 0$, wenn es vorher galt.
- Bei Anwendung eines vergrößernden Pfades (also bei Änderung der primalen Lösung) ändert sich die Anzahl der Matchingkanten in einer geschrumpften Blüte nicht. Somit gibt es auch nach Modifikation des Matchings $\lfloor |V'|/2 \rfloor$ Matchingkanten innerhalb der Knotenmenge $V' \in \mathcal{V}$, $\gamma_{V'} > 0$, wenn dies vorher der Fall war.

Zusammenfassung: Lösungen erfüllen Bedingungen 1. und 3.

Die initialen primalen und dualen Lösungen erfüllen Bedingungen 1. und 3. Durch die Modifikationen einer der beiden Lösungen wird weder Bedingung 1. noch Bedingung 3. verletzt, falls sie vorher erfüllt waren. Die jeweils vorliegenden primalen und dualen Lösungen erfüllen also während des gesamten Verfahrens Bedingungen 1. und 3.

Abschließend müssen wir argumentieren, dass Bedingung 2. früher oder später auch erfüllt wird. Dies passiert, wenn durch Modifikation des Matchings jeder Knoten v , $\alpha_v > 0$, gepaart wird oder durch Modifikation der dualen Lösung jeder einsame Knoten v einen Dualvariablenwert von $\alpha_v = 0$ erhält.

Bei der Modifikation der dualen Lösung tritt mindestens einer der folgenden Fälle ein (wiederum durch Wahl von δ).

- Wenn $\delta = \delta^V$, dann ist nach Modifikation der dualen Lösung auch Bedingung 2. erfüllt und das Verfahren endet mit der optimalen Lösung. Die Argumentation

ist analog zu der in Abschnitt 5.3.1.2. Die Dualvariablen zu einsamen Knoten in V haben identische Werte und die Dualvariable keines anderen Knoten in V hat einen niedrigeren Wert. Daher werden in diesem Fall alle Dualvariablenwerte von einsamen Knoten auf null reduziert. Somit wird Bedingung 2. erfüllt.

- Wenn $\delta = \min \{ \delta_1^E, \delta_2^E \}$, dann wird mindestens eine weitere Kante verfügbar und wir können einen der bisher gefundenen Pfade um diese Kante verlängern (und ggf. eine Blüte oder einen vergrößernden Pfad finden). Dieser Fall kann in einer Phase höchstens $|E|$ -mal eintreten, denn jede Kante wird in jeder Phase höchstens einmal verfügbar. Grundsätzlich ist es zwar möglich, dass eine Kante aus E' entnommen wird, dies passiert aber nicht in der Phase, in der sie aufgenommen wurde. Stattdessen wird eine Kante $\{v, u\}$ unmittelbar nachdem sie verfügbar wurde in einen Pfad oder in eine Blüte eingebunden (oder es wird ein anderer vergrößernder Pfad ohne diese Kante gefunden, was die Phase beendet). Wenn $\{v, u\}$ in einem Pfad, aber in keiner Blüte ist, dann ändert sich $W_{v,u}$ nicht, da einer der Knoten gerade und der andere ungerade ist. Da Pfade während einer Phase erhalten bleiben (ggf. werden Teile des Pfades als Bestandteil einer Blüte geschrumpft), bleibt $\{v, u\}$ während der gesamten Phase in einem Pfad oder wird als Bestandteil einer Blüte geschrumpft. Wenn $\{v, u\}$ in einer Blüte ist, wird $W_{v,u}$ ebenfalls nicht modifiziert (wie wir schon oben erschlossen haben). Blüten werden nicht in der Phase, in der sie geschrumpft werden, interpretiert, denn nur gerade Blüten werden interpretiert und eine geschrumpfte Blüte ist in der Phase, in der sie geschrumpft wird, ungerade. Da der Wert $W_{v,u}$ für jede Kante $\{v, u\}$ in einem gefundenen Pfad oder einer geschrumpften Blüte nicht modifiziert wird (s. o.), wird $\{v, u\}$ also nicht in derselben Phase wieder aus E' entnommen und kann somit nur einmal pro Phase in E' aufgenommen werden.
- Wenn $\delta = \delta^B$, dann erhält die Dualvariable zu mindestens einer geraden geschrumpften Blüte den Wert null. Wir interpretieren solche Blüten dann, d. h., wir ersetzen den Knoten, der die Blüte repräsentiert, durch die Blüte mit den entsprechenden Kanten. Dieser Fall kann in einer Phase höchstens $|V|/2$ -mal eintreten, denn nur gerade Blüten werden interpretiert. Da alle in der aktuellen Phase geschrumpften Blüten ungerade sind, können nur die schon in vorherigen Phasen geschrumpften Blüten interpretiert werden.

Da der zweite und dritte Fall jeweils nur endlich oft eintreten können, muss entweder, ggf. nach mehrmaligem Eintreten des zweiten Falls, ein vergrößernder Pfad gefunden werden, was die Phase beendet, oder der erste Fall eintreten, was nicht nur die Phase, sondern auch das Verfahren beendet.

Zusammenfassung: Phasen sind endlich

Eine Phase endet mit einer Modifikation des vorliegenden Matchings oder durch Erfüllung von Bedingung 2. Während einer Phase wird die duale Lösung womöglich mehrfach modifiziert (was zur Erfüllung von Bedingung 2. führen kann). Spätestens wenn alle in vorherigen Phasen geschrumpften Blüten interpretiert wurden und alle Kanten verfügbar gemacht wurden (was jeweils nur endlich oft passieren kann), muss ein vergrößernder Pfad gefunden werden oder bei der folgenden Modifikation der dualen Lösung $\delta = \delta^V$ gelten. In beiden Fällen endet die Phase.

Nun haben wir uns alles erschlossen, was wir benötigen, um für die Korrektheit des Verfahrens zu argumentieren. Jede Phase, die nicht zur Erfüllung von Bedingung 2. führt, führt dazu, dass die Kardinalität des vorliegenden Matchings erhöht wird. Da diese niemals verringert wird, kann dieser Fall also wiederum nur $|V|/2$ -mal eintreten. Wir erhalten also letztlich eine primale Lösung und eine duale Lösung, die die Bedingungen 1., 2. und 3. erfüllen und somit jeweils optimal für Modell 5.3 und Modell 5.4 sind.

Wir haben zuvor schon betont, dass wir aber gar nicht sicher wissen, ob Modell 5.3 tatsächlich Problem 5.4 (Matching maximalen Gewichts) oder ein relaxiertes Problem repräsentiert. In Kenntnis des Verfahrens sehen wir aber, dass die einzigen primalen Lösungen, die betrachtet werden, Matchings sind. Trotz dieser Einschränkung (im Vergleich zum LR von Modell 5.3) können wir argumentieren, dass wir Optimalität erreichen. Somit können wir nun auch schließen, dass Modell 5.3 mindestens eine ganzzahlige optimale Lösung hat. Wir fassen unsere Erkenntnisse im folgenden Theorem zusammen.

Theorem 5.11. *Algorithmus 5.9 findet in endlicher Zeit eine optimale Lösung zu Problem 5.4 (Matching maximalen Gewichts).*

Wir wollen hier nicht weiter darauf eingehen, aber Algorithmus 5.9 ermittelt nicht nur in endlicher Zeit eine optimale Lösung, sondern sogar in polynomieller Zeit. Wir haben bisher in diesem Buch der Laufzeitkomplexität von Verfahren keine besondere Bedeutung geschenkt, aber hier sollte betont werden, dass Algorithmus 5.9 (und effizientere Ausgestaltungen desselben) als Meilenstein bei der Entwicklung sogenannter effizienter Verfahren gilt.

5.4 Flüsse

In diesem Abschnitt soll es um die Bestimmung von Flüssen gehen. Zunächst wollen wir spezifizieren, was Flüsse überhaupt sind. Dabei müssen wir etwas weiter ausholen und die zugrunde liegende Graphenstruktur definieren.

Definition 5.7 (Flussnetzwerk). Ein Flussnetzwerk $(V, E, (w, c), s, t)$ ist ein gerichteter, (ggf. doppelt) gewichteter Graph $G = (V, E, (w, c))$ mit $w_{i,j} \geq 0$ für jede Kante $(i, j) \in E$, eine ausgewiesene Quelle $s \in V$ und eine ausgewiesene Senke $t \in V$, $s \neq t$.

Die Notation $G = (V, E, (w, c))$ haben wir in Abschnitt 2.2 nicht eingeführt. Sie soll hier dafür stehen, dass sowohl (V, E, w) als auch (V, E, c) gewichtete Graphen sind, d. h., jeder Kante in E werden zwei numerische Werte zugeordnet. Wir interpretieren dabei $w_{i,j}$ als die Kapazität einer Kante $(i, j) \in E$ und $c_{i,j}$ als Kosten einer Kante $(i, j) \in E$. Nur für das in Abschnitt 5.4.2 behandelte Problem sind die Kosten relevant. Wir werden dieses Gewicht daher in Abschnitt 5.4.1 nicht darstellen und betrachten dort einen (einfach) gewichteten Graph. Ein Flussnetzwerk stellt eine Infrastruktur aus Transportrelationen dar. Eine gerichtete Kante $(i, j) \in E$ repräsentiert dabei die Möglichkeit, Mengen von i nach j zu versenden. Die Kapazität einer Kante stellt die Höchstmenge, die über die Kante versendet werden kann, dar und die Kosten einer Kante stellen die Kosten pro über diese Kante versendete Einheit dar.

Abb. 5.78 stellt ein Flussnetzwerk F dar. An jeder Kante $(i, j) \in E$ sind Kapazität und Kosten als Tupel $(w_{i,j}, c_{i,j})$ abgetragen.

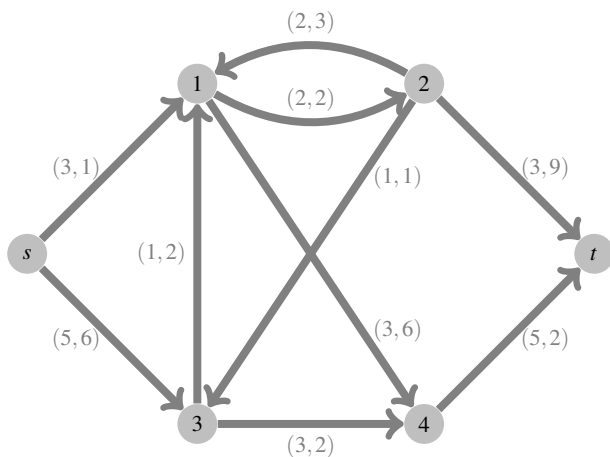


Abb. 5.78 Flussnetzwerk F

Definition 5.8 (Fluss). Ein Fluss in einem Flussnetzwerk $(V, E, (w, c), s, t)$ ist eine Funktion f , die jeder Kante $(i, j) \in E$ einen numerischen Wert $f_{i,j}$ zuordnet, sodass

1. $0 \leq f_{i,j} \leq w_{i,j}$ für jede Kante $(i, j) \in E$ und
2. $\sum_{(j,i) \in E} f_{j,i} = \sum_{(i,j) \in E} f_{i,j}$ für jeden Knoten $i \in V \setminus \{s, t\}$.

Wir können einen Fluss als Transportmenge für jede Kante des Flussnetzwerkes interpretieren. Dabei darf über keine Kante mehr als ihre Kapazität fließen und alle Knoten außer s und t dienen ausschließlich zur Weiterleitung von eingehenden Transportmengen. D. h., von einem Knoten in $V \setminus \{s, t\}$ aus wird genau die

Menge versendet (womöglich über mehrere Kanten), die bei dem Knoten eingeht (womöglich über mehrere Kanten). Für die Knoten s und t hingegen gilt diese Einschränkung nicht und daher kann von s mehr versendet werden als bei s eingeht. Wenn dies der Fall ist, muss dieser Überschuss bei t aufgefangen werden, denn alle anderen Knoten können ja nur weiterleiten. D. h., bei der Senke t „versickert“ die Menge, die bei der Quelle s entsteht. Wir erweitern die Notation in Abb. 5.78 zur Darstellung eines Flusses und tragen an jeder Kante $(i, j) \in E$ in Abb. 5.79 das Tripel $(f_{i,j}, w_{i,j}, c_{i,j})$ ab.

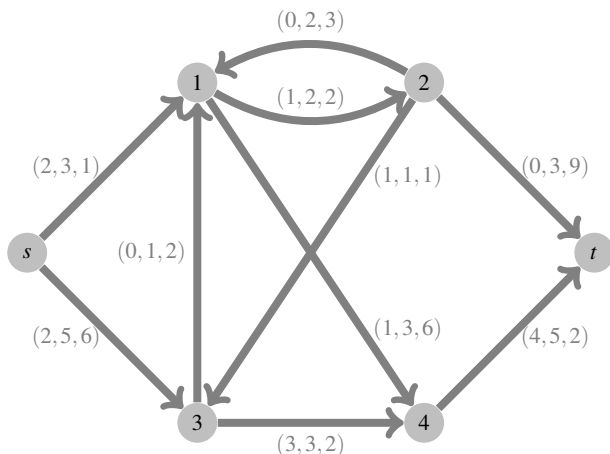


Abb. 5.79 Fluss in Flussnetzwerk F in Abb. 5.78

Die bei Knoten 1 eingehende Gesamtmenge wird auf die zwei ausgehende Kanten $(1,2)$ und $(1,4)$ verteilt. Analog werden bei Knoten 3 die Mengen, die über die Kanten $(2,3)$ und $(s,3)$ eingeht, gebündelt und über die einzelne ausgehende Kante $(3,4)$ weitergeleitet. Wir können leicht feststellen, dass keine Kapazitätsbeschränkung verletzt wird und an den Knoten 1, 2, 3 und 4 dieselbe Menge ankommt, wie abfließt.

Definition 5.9 (Wert eines Flusses). Der Wert $v(f)$ eines Flusses f in einem Flussnetzwerk $(V, E, (w, c), s, t)$ ist $v(f) = \sum_{(s,j) \in E} f_{s,j} - \sum_{(j,s) \in E} f_{j,s} = \sum_{(j,t) \in E} f_{j,t} - \sum_{(t,j) \in E} f_{t,j}$.

Der Wert eines Flusses beschreibt die Menge, die von s (nicht zwangsläufig direkt) nach t gesendet wird. Da alle Knoten außer s und t nur weiterleiten, muss ein Überschuss an Versand von s , d. h. $\sum_{(s,j) \in E} f_{s,j} - \sum_{(j,s) \in E} f_{j,s} > 0$, zwangsläufig zu einem Überschuss an Empfang bei t , d. h. $\sum_{(j,t) \in E} f_{j,t} - \sum_{(t,j) \in E} f_{t,j} > 0$, führen. Nach Definition 5.9 kann ein Fluss einen negativen Wert haben, nämlich dann, wenn t einen Überschuss an Versand hat. Der Fluss in Abb. 5.79 hat einen Wert von vier. In dem zugrunde liegenden Flussnetzwerk gibt es keinen Fluss mit negativem Wert,

was wir leicht daran erkennen, dass s bzw. t keine eingehenden bzw. ausgehenden Kanten hat.

Definition 5.10 (Kosten eines Flusses). Die Kosten $c(f)$ eines Flusses f in einem Flussnetzwerk $(V, E, (w, c), s, t)$ betragen $c(f) = \sum_{(i,j) \in E} c_{i,j} \cdot f_{i,j}$.

Die Kosten eines Flusses ergeben sich als die Summe der auf den einzelnen Kanten entstehenden Kosten. Die Kosten des Flusses in Abb. 5.79 sind 37.

Zusammenfassung: Flussnetzwerke und Flüsse

Ein Flussnetzwerk ist ein gerichteter Graph, der ein Transportnetzwerk abbildet. Die Kanten stellen dabei Transportwege zwischen Orten dar und haben Kapazitäten und ggf. Kosten. Es gibt eine Quelle und eine Senke, alle übrigen Knoten dienen lediglich der Weiterleitung von empfangenen Transportmengen.

Ein Fluss ist eine Zuordnung von Transportmengen zu jeder Kante, sodass Kapazitäten nicht überschritten werden und Transportmengen nur an der Quelle oder Senke entstehen oder verschwinden.

Der Wert eines Flusses ist die Menge, die insgesamt von der Quelle zur Senke transportiert wird. Die Kosten eines Flusses sind die insgesamt durch die Transportmengen entstehenden Kosten auf den Kanten.

Bevor wir uns mit konkreten Problemen und Verfahren beschäftigen, wollen wir ein Konzept einführen, das fundamental für die Verfahren in den Abschnitten 5.4.1 und 5.4.2 ist. Ein *wertsteigernder Pfad* (im Englischen „Augmenting Path“) hilft uns, einen bestehenden Fluss f in einen Fluss f' mit höherem Wert, d. h. $v(f') > v(f)$, zu transformieren.

Ein wertsteigernder Pfad ist nicht zwangsläufig ein Pfad in dem vorliegenden Flussnetzwerk, sondern in einem Hilfsgraph, dem *Residualgraph*, den wir für ein vorliegendes Flussnetzwerk F und einen Fluss f in diesem Flussnetzwerk konstruieren. In diesem Residualgraph gibt es sogenannte *Vorwärtskanten* und *Rückwärtskanten*. Diese repräsentieren Kanten in F , auf denen die Transportmenge in f vergrößert bzw. verringert werden kann. Die beiden Mengen bezeichnen wir mit E^v bzw. E^r . Vorwärtskante $(i, j)^v$ ist in E^v , wenn $(i, j) \in E$ und $w_{i,j} - f_{i,j} > 0$, d. h. wenn es Potential gibt, die Transportmenge auf dieser Kante zu vergrößern. Rückwärtskante $(i, j)^r$ ist in E^r , wenn $(j, i) \in E$ und $f_{j,i} > 0$, d. h. wenn es Potential gibt, die Transportmenge auf (j, i) zu verringern. Die Entsprechung von $(j, i) \in E$ im Residualgraph in Form einer Rückwärtskante hat also eine andere Orientierung als (j, i) .

Definition 5.11 (Residualgraph). Ein Residualgraph $G^r(F, f)$ für ein Flussnetzwerk $F = (V, E, (w, c), s, t)$ und Fluss f in F ist ein gerichteter und (ggf. doppelt) gewichteter Graph $G^r(F, f) = (V, E', (w', c'))$, wobei

- $(i, j) \in E'$ mit Gewichten $(w'_{i,j}, c'_{i,j}) = (w_{i,j} - f_{i,j}, c_{i,j})$, wenn $(i, j)^v \in E^v$ und $(i, j)^r \notin E^r$,
- $(i, j) \in E'$ mit Gewichten $(w'_{i,j}, c'_{i,j}) = (f_{j,i}, -c_{j,i})$, wenn $(i, j)^r \in E^r$ und $(i, j)^v \notin E^v$,
- $(i, j) \in E'$ mit Gewichten $(w'_{i,j}, c'_{i,j}) = (w_{i,j} - f_{i,j}, c_{i,j})$, wenn $(i, j)^v \in E^v$ und $(i, j)^r \in E^r$ und $c_{i,j} \leq -c_{j,i}$, und
- $(i, j) \in E'$ mit Gewichten $(w'_{i,j}, c'_{i,j}) = (f_{j,i}, -c_{j,i})$, wenn $(i, j)^r \in E^r$ und $(i, j)^v \in E^v$ und $c_{i,j} > -c_{j,i}$.

Wir wollen von dem ersten Gewicht einer Kante im Residualgraph als ihr *Potential* sprechen. Das zweite Gewicht bezeichnen wir die *Kosten* der Kante. Definition 5.11 wirkt recht kompliziert. Deshalb wollen wir sie näher erläutern und unterscheiden dabei drei Fälle, von denen der letzte die beiden letzten Punkte in der Definition abdeckt.

- Wenn es Vorwärtskante $(i, j)^v$ in E^v , aber keine Rückwärtskante $(i, j)^r$ in E^r gibt, dann hat Kante $(i, j) \in E'$ relevante Attribute, die Eigenschaften der Vorwärtskante beschreiben. Das Potential der Kante beschreibt, um wieviel die Transportmenge auf der Kante (i, j) in Fluss f vergrößert werden kann, nämlich um $w_{i,j} - f_{i,j}$. Die durch eine Vergrößerung der Transportmenge entstehenden Kosten betragen $c_{i,j}$ pro Einheit.
- Wenn es Rückwärtskante $(i, j)^r$ in E^r , aber keine Vorwärtskante $(i, j)^v$ in E^v gibt, dann hat Kante $(i, j) \in E'$ relevante Attribute, die Eigenschaften der Rückwärtskante beschreiben. Das Potential der Kante beschreibt, um wieviel die Transportmenge auf der Kante (j, i) in Fluss f verringert werden kann, nämlich um $f_{j,i}$. Die durch eine Verringerung der Transportmenge entstehenden Kosten betragen $-c_{j,i}$ pro Einheit, d. h. wir verringern die Kosten falls $c_{j,i} > 0$.
- Wenn es sowohl Vorwärtskante $(i, j)^v$ in E^v als auch Rückwärtskante $(i, j)^r$ in E^r gibt, dann hat Kante $(i, j) \in E'$ die Attribute der Kante mit niedrigeren Kosten pro Einheit der assoziierten Veränderung (Vergrößerung der Transportmenge zu Kosten von $c_{i,j}$ für $(i, j)^v$ und Verringerung der Transportmenge zu Kosten von $-c_{j,i}$ für $(i, j)^r$).

Diese etwas umständlich wirkende Definition erlaubt ein einheitliches Verständnis eines Residualgraphen in den Abschnitten 5.4.1 und 5.4.2. Im Fall eines einfach gewichteten Flussnetzwerkes $F = (V, E, w, s, t)$ vereinfacht sich die Definition, da die Kosten der Kanten $(i, j)^v \in E^v$ und $(i, j)^r \in E^r$ nicht gegeneinander abgewogen werden müssen. Wir werden später sehen, dass wir in diesem Fall einfach immer die relevanten Attribute von $(i, j)^v \in E^v$ übernehmen, wenn es Vorwärtskante $(i, j)^v$ gibt (dies ist in Einklang mit Definition 5.11).

Abb. 5.80 stellt den Residualgraph $G^r(F, f)$ für Flussnetzwerk F und Fluss f in Abb. 5.79 dar. Hier treten die drei oben erläuterten Fälle auf.

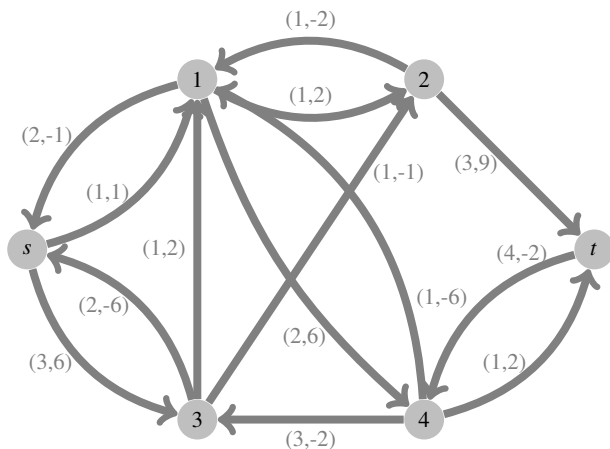


Abb. 5.80 Residualgraph $G^r(F, f)$ für F und f in Abb. 5.79

- Kante $(3, 1)$ in F hat eine Entsprechung in $G^r(F, f)$ mit gleicher Orientierung und Gewichten $(w'_{3,1}, c'_{3,1}) = (w_{3,1} - f_{3,1}, c_{3,1}) = (1, 2)$, da es Vorwärtskante $(3, 1)^v$ (wegen $f_{3,1} < w_{3,1}$), aber nicht Rückwärtskante $(3, 1)^r$ gibt (wegen $f_{3,1} = 0$).
- Kante $(2, 3)$ in F hat eine Entsprechung in $G^r(F, f)$ mit umgekehrter Orientierung und Gewichten $(w'_{3,2}, c'_{3,2}) = (f_{2,3}, -c_{2,3}) = (1, -1)$, da es keine Vorwärtskante $(2, 3)^v$ (wegen $f_{2,3} = w_{2,3}$), aber Rückwärtskante $(2, 3)^r$ gibt (wegen $f_{2,3} > 0$).
- Da $(1, 2) \in E$ und $f_{1,2} > 0$ sowie $(2, 1) \in E$ und $f_{2,1} < w_{2,1}$, erhalten wir $(2, 1)^r \in E^r$ sowie $(2, 1)^v \in E^v$. Wie in Definition 5.11 festgelegt, setzen wir dann $(w'_{2,1}, c'_{2,1}) = (f_{1,2}, -c_{1,2})$, da die Kosten, die mit $(2, 1)^r$ assoziiert sind, niedriger als die zu $(2, 1)^v$ sind.

Weiterhin sehen wir, dass eine Kante in F mehrere Entsprechungen in $G^r(F, f)$ haben kann. Kante $(1, 4)$ in F wird durch $(1, 4)^v$ mit Gewichten $(w'_{1,4}, c'_{1,4}) = (w_{1,4} - f_{1,4}, c_{1,4}) = (2, 6)$ und durch $(4, 1)^r$ mit Gewichten $(w'_{4,1}, c'_{4,1}) = (f_{1,4}, -c_{1,4}) = (1, -6)$ repräsentiert, da sowohl $f_{1,4} < w_{1,4}$ als auch $f_{1,4} > 0$ gilt.

Definition 5.12 (Wertsteigernder Pfad). Ein wertsteigernder Pfad für Flussnetzwerk $F = (V, E, (w, c), s, t)$ und Fluss f in F ist ein Pfad von s nach t in dem entsprechenden Residualgraphen $G^r(F, f)$.

In Abb. 5.80 können wir durch „scharfes Hinsehen“ mehrere wertsteigernde Pfade erkennen: $(s, 1, 2, t)$, $(s, 1, 4, t)$, $(s, 1, 4, 3, 2, t)$, $(s, 3, 2, t)$, $(s, 3, 1, 2, t)$ und $(s, 3, 1, 4, t)$.

Ein wertsteigernder Pfad P für Flussnetzwerk $F = (V, E, (w, c), s, t)$ und Fluss f in F erlaubt uns unmittelbar, den Fluss f so zu modifizieren, dass der erhaltene Fluss einen höheren Wert hat. Ein wertsteigernder Pfad P hat hierfür ein Potential $\delta^{\max}(P)$ in Höhe des minimalen Potentials unter den in ihm enthaltenen Kanten. Wir können Fluss f modifizieren, indem wir auf jeder Kante in F , die einer Kante

des Pfades in E' entspricht, die Transportmenge um $\delta \leq \delta^{\max}(P)$ verringern und auf jeder Kante in F , die einer Kante des Pfades in E^v entspricht, die Transportmenge um δ vergrößern.

Wir wollen uns vergewissern, dass wir durch diese Modifikation tatsächlich einen Fluss f' erhalten. Bedingung 1. aus Definition 5.8 wird aufgrund der Wahl von $\delta \leq \delta^{\max}(P)$ und der Bestimmung der Potentiale der Kanten im Residualgraph $G^r(F, f)$ eingehalten. Da der Pfad bei s startet und bei t endet, enthält er zudem für jeden Knoten in $V \setminus \{s, t\}$ genau eine eingehende und genau eine ausgehende Kante in $G^r(F, f)$ oder gar keine inzidente Kante in $G^r(F, f)$. Für einen Knoten in $V \setminus \{s, t\}$, der durch den Pfad nicht besucht wird, bleibt Bedingung 2. offensichtlich erfüllt. Für jeden anderen Knoten $i \in V \setminus \{s, t\}$, der durch den Pfad besucht wird, können wir vier Fälle unterscheiden.

- Knoten i wird durch zwei Kanten $(j, i) \in E'$ und $(i, k) \in E'$, die den Kanten $(j, i)^r$ und $(i, k)^r$ in E^r entsprechen, erreicht und verlassen. Dann werden die Transportmengen auf den Kanten $(k, i) \in E$ und $(i, j) \in E$ jeweils um δ verringert.
- Knoten i wird durch zwei Kanten $(j, i) \in E'$ und $(i, k) \in E'$, die den Kanten $(j, i)^v$ und $(i, k)^v$ in E^v entsprechen, erreicht und verlassen. Dann werden die Transportmengen auf den Kanten $(j, i) \in E$ und $(i, k) \in E$ jeweils um δ vergrößert.
- Knoten i wird durch zwei Kanten $(j, i) \in E'$ und $(i, k) \in E'$, die den Kanten $(j, i)^r \in E^r$ und $(i, k)^v \in E^v$ entsprechen, erreicht und verlassen. Dann wird die Transportmenge auf der Kante $(i, j) \in E$ um δ verringert und die Transportmenge auf der Kante $(i, k) \in E$ um δ vergrößert.
- Knoten i wird durch zwei Kanten $(j, i) \in E'$ und $(i, k) \in E'$, die Kanten $(j, i)^v \in E^v$ und $(i, k)^r \in E^r$ entsprechen, erreicht und verlassen. Dann wird die Transportmenge auf der Kante $(j, i) \in E$ um δ vergrößert und die Transportmenge auf der Kante $(k, i) \in E$ um δ verringert.

In jedem der vier Fälle bleiben die Transportmengen auf allen übrigen zu i inzidenten Kanten in F unverändert und daher ist leicht zu erkennen, dass Bedingung 2. in Definition 5.8 durch f' erfüllt wird.

In Abb. 5.81 wird Fluss f' dargestellt, den wir durch Anwendung des wertsteigernden Pfades $(s, 1, 4, 3, 2, t)$ mit $\delta = \min\{1, 2, 3, 1, 3\} = 1$ auf den Fluss in Abb. 5.79 erhalten.

Jeder der oben beschriebenen vier Fälle trifft hier auf einen Knoten zu und wir können daher beispielhaft nachvollziehen, wieso Bedingung 2. durch f' erfüllt wird. Zudem wurden die Potentiale der Kanten im Residualgraph in Abb. 5.80 so bestimmt, dass bei entsprechender Vergrößerung oder Verringerung der Transportmengen keine Transportmenge negativ wird und keine Kantenkapazität verletzt wird, wenn $\delta = 1 \leq \delta^{\max}(P)$ gewählt wird. Die Kosten steigen entsprechend der gesamten Kosten des wertsteigernden Pfades in $G^r(F, f)$ um 13.

Diesen Mechanismus zur Vergrößerung des Wertes eines Flusses unter Verwendung wertsteigernder Pfade werden wir in den Abschnitten 5.4.1 und 5.4.2 verwenden, um die dort behandelten Probleme zu lösen.

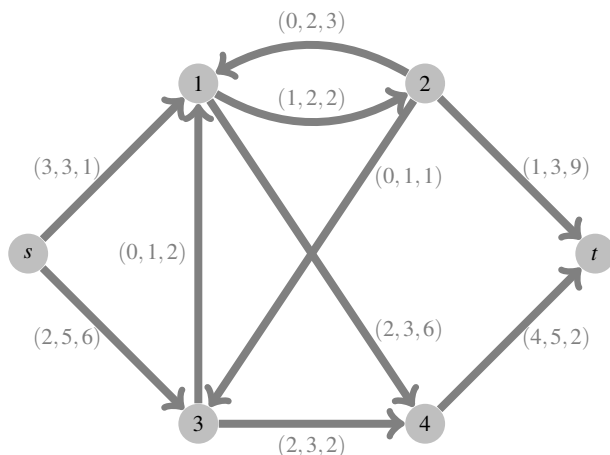


Abb. 5.81 Fluss f' in Flussnetzwerk F in Abb. 5.78

Zusammenfassung: Residualgraph und wertsteigernde Pfade

Ein Residualgraph zeigt für ein Flussnetzwerk F und einen Fluss f Möglichkeiten auf, einer Kante eine größere oder eine kleinere Transportmenge als in f zuzuweisen. Kanten in E^v stehen dabei für eine Vergrößerung und Kanten in E^r für eine Verringerung der Transportmenge. Die Potentiale und Kosten der Kanten entsprechen dabei der größtmöglichen zulässigen Veränderung und dem Kostenfaktor der Veränderung.

Die Kantenmenge in einem Pfad des Residualgraphs von s nach t , also in einem wertsteigernden Pfad, ist eine Menge von Kanten, auf denen die Transportmengen simultan so verändert werden können, dass die Veränderung wiederum einen Fluss erzeugt. Dabei müssen die Transportmengen auf Kanten in E^r verringert und auf Kanten in E^v vergrößert werden. Die betraglich größte mögliche Veränderung entspricht dabei dem kleinsten Potential einer Kante im Pfad.

5.4.1 Maximale Flüsse und Minimale Schnitte

Wir werden uns in diesem Abschnitt mit der Lösung zweier Probleme befassen, die im Kontext von Flussnetzwerken betrachtet werden können. Das erste Problem lässt sich recht leicht dadurch motivieren, dass wir eine möglichst große Menge durch das Netzwerk von der Quelle zur Senke transportieren wollen.

Problem 5.5 (Maximaler Fluss).**Eingabe:** Flussnetzwerk $F = (V, E, w, s, t)$ **LR:** Menge aller Flüsse in F **ZB:** –**ZS:** Wert des Flusses ist zu maximieren.**Frage:** Was ist eine optimale Lösung?

Eine Lösung zu Problem 5.5 (Maximaler Fluss) ist immer dann interessant, wenn wir ein zu transportierendes Gut betrachten, bei dem es erstrebenswert ist, es zur Senke zu bringen. Wenn dies nicht der Fall ist, d. h., wir wollen gerade verhindern, dass das Gut die Senke erreicht, dann ergibt sich eine scheinbar vollständig andere Perspektive. Um zu verhindern, dass das Gut von der Quelle zur Senke gelangt, können Transportverbindungen aus dem Netzwerk entnommen (oder blockiert oder zerstört) werden. Dies soll so geschehen, dass es keinen Pfad von s nach t gibt, denn dann kann kein Transport von der Quelle zur Senke mehr erfolgen. Wir betrachten dabei einen sogenannten Schnitt eines Flussnetzwerks $F = (V, E, w, s, t)$. Schrijver (2002) gibt einen Überblick über die Betrachtung der beiden Probleme in einem historischen Kontext. Wir werden den Schnitt im Folgenden nicht als eine Menge von (zu entnehmenden) Kanten beschreiben (was naheliegend wäre), sondern durch die Menge von Knoten, die von s erreichbar ist. Diese Menge impliziert dann die zu entnehmenden Kanten.

Definition 5.13 (Schnitt). Ein Schnitt eines Flussnetzwerkes (V, E, w, s, t) ist eine Teilmenge V' von Knoten, sodass $s \in V'$ und $t \notin V'$.

Wir interpretieren diese Teilmenge V' als diejenigen Knoten, die auf Pfaden von s aus erreicht werden können, und die Komplementärmenge $V \setminus V'$ als diejenigen Knoten, die nicht auf Pfaden von s aus erreicht werden können. In einem sinnvollen Flussnetzwerk ist t zunächst von s aus erreichbar. D. h., $t \notin V'$ ist nur zu erreichen, wenn wir Kanten aus dem Flussnetzwerk entfernen. Dass die Knoten in V' erreicht werden und die in $V \setminus V'$ nicht, erreichen wir, indem wir alle Kanten, die von einem Knoten in V' zu einem Knoten in $V \setminus V'$ führen, entfernen. Die Menge dieser Kanten wird also durch die Menge V' als zu entfernen impliziert.

Definition 5.14 (Kapazität eines Schnittes). Die Kapazität $c(V')$ eines Schnittes V' eines Flussnetzwerkes (V, E, w, s, t) ist $c(V') = \sum_{(i,j) \in E, i \in V', j \notin V'} w_{i,j}$.

Das folgende Problem motiviert sich durch die Zielsetzung, Kanten aus dem Netzwerk so zu entfernen, dass es keinen Pfad von s nach t gibt und dafür möglichst wenig Aufwand betrieben wird. Wenn wir den Aufwand für das Entfernen einer Kante mit ihrer Kapazität gleichsetzen, ergibt sich Problem 5.6 (Minimaler Schnitt).

Problem 5.6 (Minimaler Schnitt).**Eingabe:** Flussnetzwerk $F = (V, E, w, s, t)$ **LR:** Schnitte**ZB:** –**ZS:** Die Kapazität des Schnittes ist zu minimieren.**Frage:** Was ist eine optimale Lösung?

Zwischen Problem 5.5 (Maximaler Fluss) und Problem 5.6 (Minimaler Schnitt) gibt es eine sehr enge Beziehung. Wir können relativ leicht argumentieren, dass die Kapazität jedes Schnittes nicht kleiner sein kann als der Wert jedes Flusses in einem Flussnetzwerk. Um dies zu sehen, betrachten wir einen willkürlichen Schnitt V' und betrachten alle Kanten, die von einem Knoten in V' zu einem Knoten in $V \setminus V'$ verlaufen. Der Wert eines Flusses, also die gesamte von s nach t transportierte Menge, muss von V' nach $V \setminus V'$ fließen, da $s \in V'$ und $t \notin V'$. Von V' nach $V \setminus V'$ kann aber nicht mehr als $c(V')$ fließen. Tatsächlich ist die Beziehung zwischen den beiden Problemen aber sogar noch enger, wie das folgende Theorem formuliert.

Theorem 5.12. *Für ein Flussnetzwerk entspricht der maximale Wert eines Flusses der minimalen Kapazität eines Schnittes.*

In Anhang C.9 führen wir einen Beweis, der auf Vorwissen aus Abschnitt 3.4 aufbaut. Im Folgenden stellen wir ein Verfahren vor, das beide Probleme löst. Aus der Korrektheit dieses Verfahrens folgt Theorem 5.12 ebenfalls, wie wir später sehen werden.

Salopp formuliert: Maximale Flüsse und (minimale) Schnitte

Ein maximaler Fluss ist ein Fluss mit einer maximalen von s nach t versendeten Transportmenge. Ein Schnitt entspricht einer Kantenmenge, sodass ohne diese Kanten nichts von s nach t versendet werden kann. Die Kapazität eines Schnitts ist die gesamte Kapazität dieser Kanten und ein minimaler Schnitt ist ein Schnitt mit minimaler Kapazität. In einem beliebigen Flussnetzwerk entspricht der maximale Wert unter allen Flüssen der minimalen Kapazität unter allen Schnitten.

Überblick über das Verfahren

Das Verfahren, das in diesem Abschnitt vorgestellt wird, geht auf Ford Jr. und Fulkerson (1956) zurück. Es betrachtet iterativ jeweils einen aktuellen Fluss f und prüft, ob es einen wertsteigernden Pfad für das Flussnetzwerk $F = (V, E, w, s, t)$ und den aktuellen Fluss f gibt. Wenn es keinen gibt, hat der aktuelle Fluss maximalen Wert. Ansonsten modifizieren wir Fluss f mittels des wertsteigernden Pfads, erhalten einen neuen aktuellen Fluss mit höherem Wert und wiederholen den Schritt.

Initial kann dabei der Fluss betrachtet werden, bei dem alle Transportmengen null sind.

Es bleibt die Frage, wie wir ermitteln, ob es einen wertsteigernden Pfad gibt. Wir können hierfür in jeder Iteration den Residualgraph $G^r(F, f) = (V, E', w')$ für F und den aktuellen Fluss f in F konstruieren. Dann können wir z. B. Algorithmus 5.1 verwenden (siehe Abschnitt 5.1.1), um einen Pfad von s nach t in $G^r(F, f)$ zu finden, der die wenigsten Kanten enthält (hierfür weisen wir jeder Kante eine Länge von eins zu). Wir suchen zwar gar nicht zwangsläufig solch einen speziellen Pfad, aber solange es überhaupt einen Pfad von s nach t gibt, gibt es auch einen mit einer minimalen Anzahl an Kanten und wir können diesen mit Algorithmus 5.1 ermitteln.

Zusammenfassung: Verfahren von Ford und Fulkerson

Wir starten mit einem beliebigen Fluss. In jeder Iteration prüfen wir, ob es einen wertsteigernden Pfad gibt, und modifizieren den vorliegenden Fluss entsprechend, wenn wir einen solchen Pfad finden. Die Prüfung können wir mittels Verfahren zum Finden kürzester Wege in Graphen durchführen. Wenn kein solcher Pfad existiert, dann ist der vorliegende Fluss maximal und das Verfahren endet.

Das gesamte Verfahren lässt sich als Algorithmus 5.10 konkretisieren.

Algorithmus 5.10 (Ford & Fulkerson).

Eingabe: Flussnetzwerk $F = (V, E, w, s, t)$

Ausgabe: Fluss maximalen Wertes

1. Setze $f_{i,j} \leftarrow 0$ für jede Kante $(i, j) \in E$.
2. Konstruiere Residualgraph $G^r(F, f)$.
3. Prüfe mittels Algorithmus 5.1, ob es einen Pfad von s nach t in $G^r(F, f)$ gibt (und finde ggf. einen).
4. Wenn es keinen Pfad von s nach t gibt, hat f maximalen Wert. STOP.
5. Wenn es einen Pfad P von s nach t gibt, modifiziere f unter Verwendung von P (nutze dabei das volle Potential von P) und gehe zu 2.

Beispiel

Bevor wir das Verfahren analysieren, wollen wir es an einem Beispiel veranschaulichen. Wir wenden Algorithmus 5.10 auf das Flussnetzwerk $F = (V, E, w, s, t)$ in Abb. 5.82 an.

Den Ablauf des Verfahrens stellen wir mittels Abb. 5.83 bis Abb. 5.91 dar. Wir bestimmen einen ersten Fluss f^0 , indem wir $f_{i,j}^0 \leftarrow 0$ für alle $(i, j) \in E$ setzen. Der Residualgraph $G^r(F, f^0)$ für F und f^0 in der ersten Iteration entspricht dem Flussnetzwerk F in Abb. 5.82, da auf jeder Kante die volle Kapazität verbleibt und keine Transportmenge reduziert werden kann. Es fällt nicht schwer, einen Pfad von s nach t in $G^r(F, f^0)$ zu finden, der eine minimale Anzahl Kanten, in diesem Fall drei,

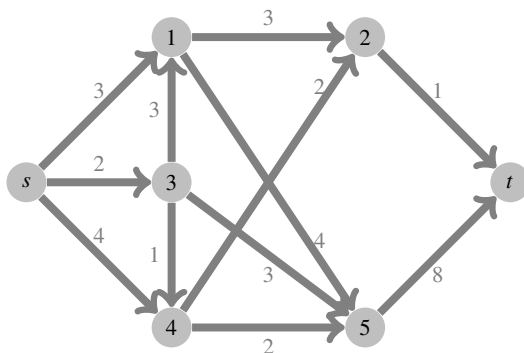


Abb. 5.82 Flussnetzwerk F
einer Instanz von Problem 5.5
(Maximaler Fluss)

enthält. Hierfür könnten wir Algorithmus 5.1 einsetzen, verlassen uns der Einfachheit halber aber auf „scharfes Hinsehen“. Der gefundene Pfad P^1 ist in Abb. 5.83 durch schwarze Kanten hervorgehoben.

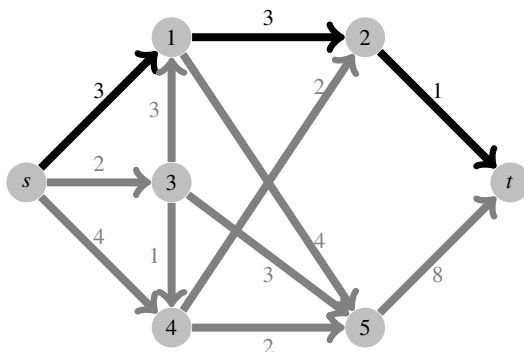


Abb. 5.83 Residualgraph
 $G^r(F, f^0)$ mit wertsteigerndem Pfad P^1

Durch Anwendung von P^1 modifizieren wir f^0 und erhalten den in Abb. 5.84 dargestellten Fluss f^1 . In jedem Tupel steht der erste Eintrag für die Transportmenge und der zweite Eintrag für die Kapazität der Kante.

In der zweiten Iteration ergibt sich ein Residualgraph, der sich in der Struktur von F unterscheidet, siehe Abb. 5.85. Kante $(2, t)$ ist voll ausgelastet und hat somit keine verbleibende Kapazität mehr. Darum wird sie durch eine einzelne Kante in umgekehrter Orientierung in $G^r(F, f^1)$ repräsentiert. Die Kanten $(s, 1)$ und $(1, 2)$ haben sowohl verbleibende Kapazität als auch eine positive Transportmenge und somit finden sich jeweils zwei entsprechende Kanten in $G^r(F, f^1)$. Dabei hat die jeweils umgekehrt orientierte Kante ein Potential in Höhe der Transportmenge (weil die Transportmenge auf null reduziert werden könnte) und die jeweils gleich orientierte Kante ein Potential in Höhe der verbleibenden Kapazität, weil die Transportmenge um diese vergrößert werden kann. Alle übrigen Kanten haben keine positive Trans-

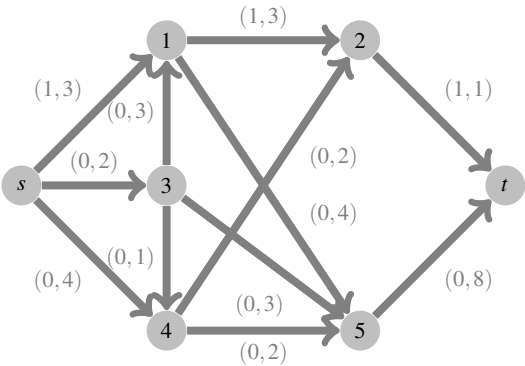


Abb. 5.84 Fluss f^1

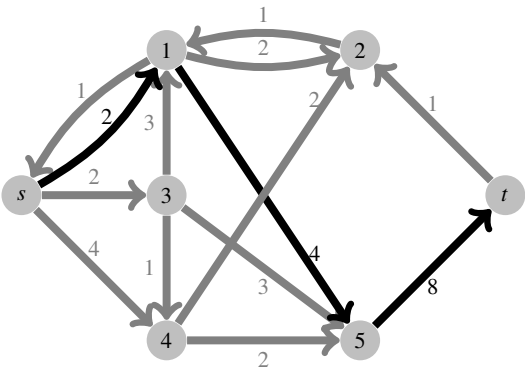


Abb. 5.85 Residualgraph $G^r(F, f^1)$ mit wertsteigerndem Pfad P^2

portmenge und sind daher nur mit derselben Orientierung und der vollen Kapazität als Potential in $G^r(F, f^1)$ repräsentiert.

Ein Pfad P^2 von s nach t mit minimaler Anzahl Kanten ist in Abb. 5.85 durch die schwarzen Kanten hervorgehoben. Durch Anwendung von P^2 modifizieren wir f^1 und erhalten den in Abb. 5.86 dargestellten Fluss f^2 .

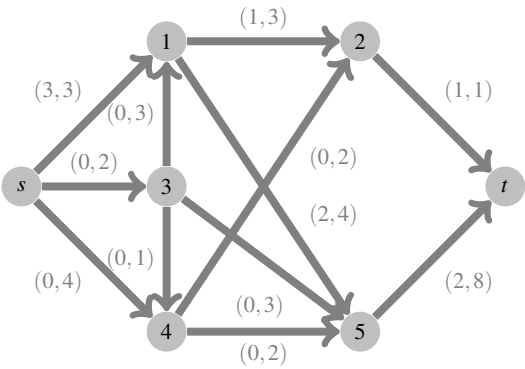


Abb. 5.86 Fluss f^2

In der dritten Iteration ergeben sich Residualgraph $G^r(F, f^2)$ und der wertsteigernde Pfad P^3 in Abb. 5.87.

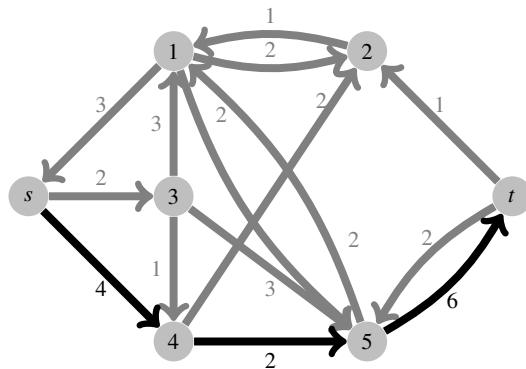


Abb. 5.87 Residualgraph $G^r(F, f^2)$ mit wertsteigerndem Pfad P^3

Durch Anwendung von P^3 modifizieren wir f^2 und erhalten den in Abb. 5.88 dargestellten Fluss f^3 .

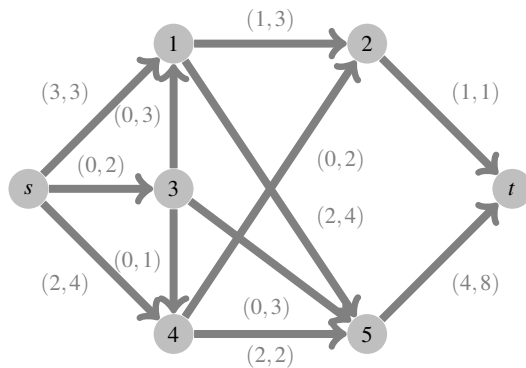


Abb. 5.88 Fluss f^3

In der vierten Iteration ergeben sich Residualgraph $G^r(F, f^3)$, der wertsteigernde Pfad P^4 und modifizierter Fluss f^4 , wie in Abb. 5.89 und 5.90 dargestellt.

In der fünften Iteration ergeben sich der Residualgraph $G^r(F, f^4)$ und der wertsteigernde Pfad P^5 , wie in Abb. 5.91 dargestellt. Der wertsteigernde Pfad P^5 enthält zwei Kanten aus E^r , nämlich $(2, 1)$ und $(1, 3)$.

Durch Anwendung von P^5 modifizieren wir f^4 und erhalten den in Abb. 5.92 dargestellten Fluss f^5 .

In der sechsten Iteration ergibt sich der Residualgraph $G^r(F, f^5)$ in Abb. 5.93. Die Knoten 2 und 4 können auf Pfaden von s aus erreicht werden; die Senke t kann nicht erreicht werden.

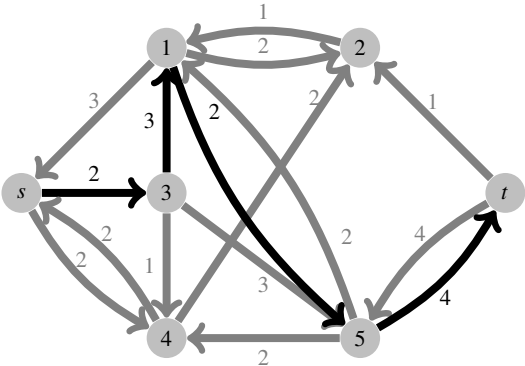


Abb. 5.89 Residualgraph $G^r(F, f^3)$ mit wertsteigerndem Pfad P^4

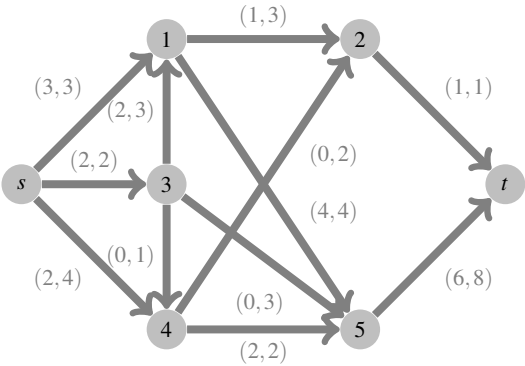


Abb. 5.90 Fluss f^4

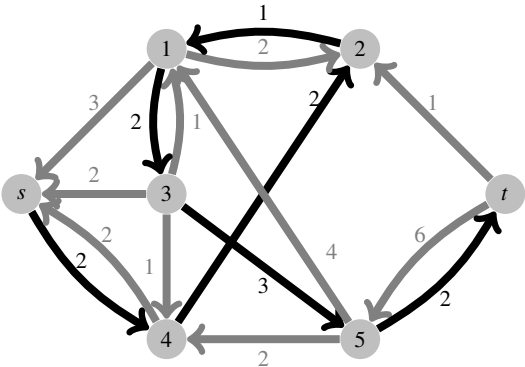
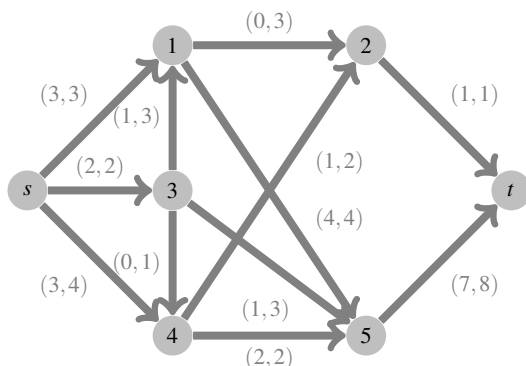
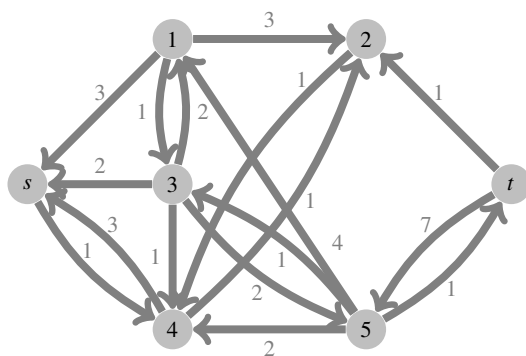


Abb. 5.91 Residualgraph $G^r(F, f^4)$ mit wertsteigerndem Pfad P^5

Algorithmus 5.10 endet und der vorliegende Fluss hat maximalen Wert. Dies können wir erkennen, wenn wir den Schnitt betrachten, der die Knoten enthält, zu denen im Residualgraph in Abb. 5.93 Pfade von s führen. Dieser Schnitt $V' = \{s, 2, 4\}$ hat eine Kapazität von $w_{s,1} + w_{s,3} + w_{2,t} + w_{4,5} = 8$ und der vorliegende Fluss f^5 hat einen Wert von ebenfalls $f_{s,1} + f_{s,3} + f_{s,4} = f_{2,t} + f_{5,t} = 8$. Da, wie oben argumentiert, kein Fluss einen höheren Wert haben kann als die minimale Ka-

Abb. 5.92 Fluss f^5 Abb. 5.93 Residualgraph
 $G'(F, f^5)$

pazität eines Schnitts, hat der vorliegende Fluss f^5 maximalen Wert und Schnitt V' minimale Kapazität.

Analyse

Nachdem wir gesehen haben, dass Algorithmus 5.10 die Problem Instanz in Abb. 5.82 von Problem 5.5 (Maximaler Fluss) und Problem 5.6 (Minimaler Schnitt) löst, besagt Theorem 5.13, dass er dies mit einer kleinen Einschränkung tatsächlich für jede Instanz von Problem 5.5 (Maximaler Fluss) und Problem 5.6 (Minimaler Schnitt) leistet.

Theorem 5.13. *Wenn alle Kantengewichte rational sind, bestimmt Algorithmus 5.10 einen Fluss maximalen Werts und die Menge von Knoten, die im finalen Residualgraph von s aus erreichbar sind, ist ein Schnitt minimaler Kapazität.*

Den Beweis führen wir in Anhang C.10. Auch wenn die Einschränkung auf rationale Kantengewichte etwas verwundern mag, ist es tatsächlich so, dass Algorithmus 5.10 womöglich nicht terminiert, wenn es nicht-rationale Kantengewichte gibt. Algorithmus 5.10 konvergiert dann zwar gegen einen bestimmten Wert von Flüssen,

aber dieser Wert mag deutlich unter dem maximalen Wert unter allen Flüssen liegen. Zwick (1995) präsentiert sehr kleine Netzwerke mit sechs Knoten und acht Kanten, in denen dies passiert.

Die hier vorgestellte Auffassung eines Flusses kann leicht dahingehend erweitert werden, dass wir mehrere Quellen und mehrere Senken berücksichtigen. Wir können trotzdem das hier vorgestellte Verfahren nutzen, um dieses verallgemeinerte Problem zu lösen, indem wir es auf Problem 5.5 (Maximaler Fluss) (mit genau einer Quelle und genau einer Senke) zurückführen. Dies können wir erreichen, indem wir eine Super-Quelle einführen, von der aus eine Kante mit unbegrenzter Kapazität zu jeder Quelle führt. Analog führen wir eine Super-Senke ein, zu der eine Kante mit unbegrenzter Kapazität von jeder Senke führt.

Das Prinzip ist in Abb. 5.94 veranschaulicht. Im ursprünglichen Flussnetzwerk gibt es die drei Quellen s_1 , s_2 und s_3 , die zwei Senken t_1 und t_2 , die weiteren Knoten 1 bis 5 und die durchgezogenen Kanten zwischen diesen Knoten. Wir fügen dann

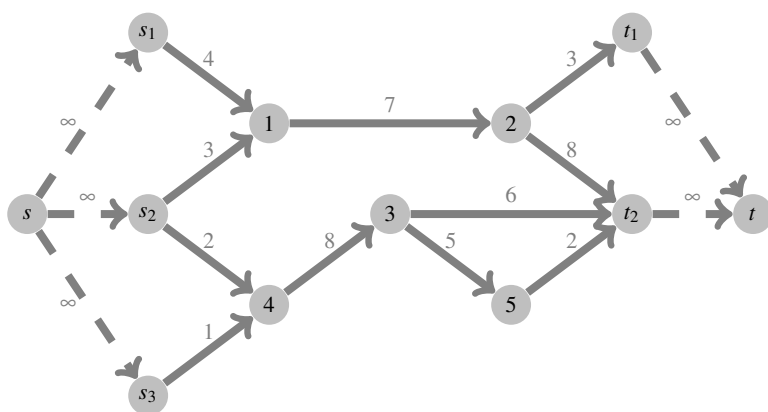


Abb. 5.94 Super-Quelle und Super-Senke

eine Super-Quelle s und eine Super-Senke t ein, die über die gestrichelten Kanten mit den Quellen und Senken verbunden sind. Diese neuen Kanten haben eine unbegrenzte Kapazität und schränken dadurch Flüsse im erweiterten Flussnetzwerk nicht ein. Daher entsprechen die Transportmengen auf Kanten des ursprünglichen Flussnetzwerks einem Fluss mit mehreren Quellen und Senken.

5.4.2 Flüsse minimaler Kosten

Wir werden uns in diesem Abschnitt mit der Lösung eines weiteren Problems befassen, das sich für Flussnetzwerke ergibt. Bei diesem Problem geht es darum, einen Fluss mit einem bestimmten Wert zu ermitteln, der unter allen Flüssen dieses Wer-

tes minimale Kosten hat. Der vorgegebene Wert muss nicht zwangsläufig, kann aber der maximale Wert unter allen Flüssen sein.

Problem 5.7 (Fluss minimaler Kosten).

Eingabe: Flussnetzwerk $F = (V, E, (w, c), s, t)$, Wert b

LR: Menge aller Flüsse in F

ZB: Fluss muss Wert b haben.

ZS: Kosten des Flusses sind zu minimieren.

Frage: Was ist eine optimale Lösung?

Die Motivation von Problem 5.7 (Fluss minimaler Kosten) fällt nicht schwer. Wenn wir vorab wissen, welche Menge wir insgesamt von der Quelle zur Senke transportieren müssen, wollen wir dies zu den geringsten Kosten erreichen.

Wir werden im weiteren Verlauf zwei Verfahren besprechen. Beide ermitteln optimale Lösungen für das Problem (eins davon mit leichten Einschränkungen) und insofern könnten wir es auch bei einem der beiden belassen. Da in beiden Verfahren aber unterschiedliche intuitiv motivierbare Ansätze verfolgt werden, die auf ähnlichen Prinzipien beruhen und sich leicht durch Anwendung von Verfahren, die in vorherigen Abschnitten erläutert werden, ausgestalten lassen, wollen wir beide besprechen.

Ein wichtiges Konzept, das wir für beide Verfahren benötigen, ist ein Kreis negativer Kosten im Residualgraph für ein Flussnetzwerk und einen Fluss.

Theorem 5.14. *Ein Fluss f in Flussnetzwerk $F = (V, E, (w, c), s, t)$ ist genau dann kostenminimal unter allen Flüssen mit Wert $v(f)$, wenn es keinen Kreis negativer Kosten in dem Residualgraph $G^r(F, f)$ gibt.*

Es ist relativ leicht zu erkennen, dass wir von Fluss f einen Fluss gleichen Wertes ableiten können, wenn wir einen solchen Kreis kennen. Wir wenden diesen Kreis an, wie wir wertsteigernde Pfade angewendet haben, und erkennen relativ leicht, dass wir einen Fluss gleichen Wertes erhalten. Die Überlegungen hierzu sind dieselben wie zu wertsteigernden Pfaden, siehe Definition 5.12 auf S. 377 und die dort folgenden Erläuterungen. Wenn die Summe der Kosten der Kanten im Kreis negativ ist, dann sind die Kosten des neuen Flusses geringer als $c(f)$. Ausführlicher werden wir hierauf in Anhang C.11 eingehen und zudem auch zeigen, dass f tatsächlich kostenminimal ist, wenn kein Kreis negativer Kosten in $G^r(F, f)$ existiert. Diese zentrale Erkenntnis wird in den Abschnitten 5.4.2.1 und 5.4.2.2 genutzt, um zwei Verfahren zur Lösung von Problem 5.7 (Fluss minimaler Kosten) zu motivieren bzw. deren Korrektheit zu zeigen.

Auch hier können wir die Auffassung eines Flusses dahingehend erweitern, dass wir mehrere Quellen und mehrere Senken berücksichtigen. Wir können aber auch dann die im Folgenden diskutierten Verfahren nutzen, indem wir es auf Problem 5.7 (Fluss minimaler Kosten) (mit genau einer Quelle und genau einer Senke) zurückführen. Dies können wir wiederum durch Einführung einer Super-Quelle und einer Super-Senke erreichen. Von der Super-Quelle führt eine Kante mit unbegrenz-

ter Kapazität und Kosten von null zu jeder Quelle und von jeder Senke führt eine Kante mit unbegrenzter Kapazität und Kosten von null zur Super-Senke.

5.4.2.1 Eliminierung negativer Kreise

Das Verfahren, das in diesem Abschnitt vorgestellt wird, geht auf Klein (1967) zurück. Es konstruiert zunächst eine zulässige Lösung und modifiziert diese sukzessiv, um die Kosten zu reduzieren.

Überblick über das Verfahren

Zunächst wird ein Fluss mit Wert b ermittelt. Dies kann z. B. durch Anwendung von Algorithmus 5.10, vorgestellt in Abschnitt 5.4.1, erfolgen, wobei das Verfahren abgebrochen wird, sobald ein Wert von b erreicht ist. Nun wird der Residualgraph $G^r(F, f)$ für F und den aktuellen Fluss f konstruiert und geprüft, ob es einen Kreis negativer Kosten in $G^r(F, f)$ gibt. Dies kann durch Anwendung von Algorithmus 5.2, vorgestellt in Abschnitt 5.1.2, erreicht werden, wobei die Kosten als Längen der Kanten in $G^r(F, f)$ interpretiert werden. Sollte ein Kreis negativer Kosten gefunden werden, wird dieser auf den aktuellen Fluss f angewendet. Wir erhalten dann einen Fluss f' mit demselben Wert und niedrigeren Kosten und wiederholen den Schritt mit f' als aktuellem Fluss. Sollte kein Kreis negativer Kosten existieren, ist der aktuelle Fluss f kostenminimal.

Salopp formuliert: Eliminierung negativer Kreise

Wir konstruieren einen Fluss mit Wert b , ohne auf die Kosten zu achten. Anschließend reduzieren wir iterativ die Kosten durch Anwendung negativer Kreise im Residualgraph bis es keinen solchen Kreis mehr gibt.

Das gesamte Verfahren lässt sich als Algorithmus 5.11 konkretisieren.

Algorithmus 5.11 (Eliminierung negativer Kreise).

Eingabe: Flussnetzwerk $F = (V, E, (w, c), s, t)$, Wert b

Ausgabe: Fluss minimaler Kosten mit Wert b oder Information, dass keiner existiert

1. Ermittle Fluss f mit Wert b (z. B. mit Algorithmus 5.10).
2. Wenn es keinen Fluss mit Wert b gibt, gibt es keine zulässige Lösung. STOP.
3. Konstruiere Residualgraph $G^r(F, f) = (V, E', (w', c'))$.
4. Prüfe mittels Algorithmus 5.2, ob es einen Kreis negativer Kosten in $G^r(F, f)$ gibt (und finde ggf. einen).
5. Wenn es keinen Kreis negativer Kosten gibt, hat f minimale Kosten. STOP.
6. Wenn es einen Kreis K negativer Kosten gibt, modifiziere f unter Verwendung von K .
7. Gehe zu 3.

Beispiel

Bevor wir das Verfahren analysieren, wollen wir es an einem Beispiel veranschaulichen. Wir wenden Algorithmus 5.11 auf das Flussnetzwerk $F = (V, E, w, s, t)$ in Abb. 5.95 an. Der geforderte Wert des zu findenden Flusses ist $b = 4$.

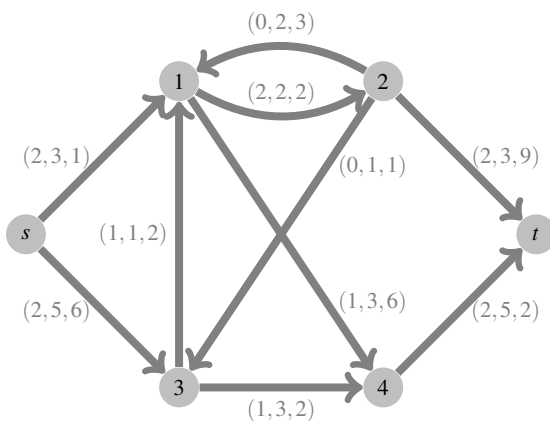


Abb. 5.95 Fluss f^0 in Flussnetzwerk F

Den Ablauf des Verfahrens stellen wir mittels Abb. 5.96 bis 5.104 dar.

Wir kürzen das Finden eines ersten Flusses f^0 mit $v(f^0) = 4$ ab und nehmen an, dass wir ihn durch „scharfes Hinsehen“ ermittelt haben. Fluss f^0 ist ebenfalls in Abb. 5.95 dargestellt. Der Residualgraph $G^r(F, f^0)$ in der ersten Iteration ist in Abb. 5.96 abgebildet. Wiederum durch „scharfes Hinsehen“ können wir den schwarz hervorgehobenen Kreis negativer Kosten erkennen. Alternativ könnten wir Algorithmus 5.2 einsetzen, um einen Kreis negativer Kosten zu finden.

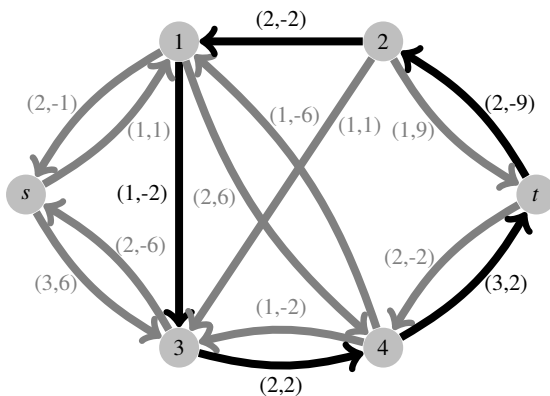


Abb. 5.96 Residualgraph $G^r(F, f^0)$ und Kreis negativer Kosten

Wir wenden den Kreis in Abb. 5.96 auf Fluss f^0 in Abb. 5.95 an und erhalten Fluss f^1 in Abb. 5.97.

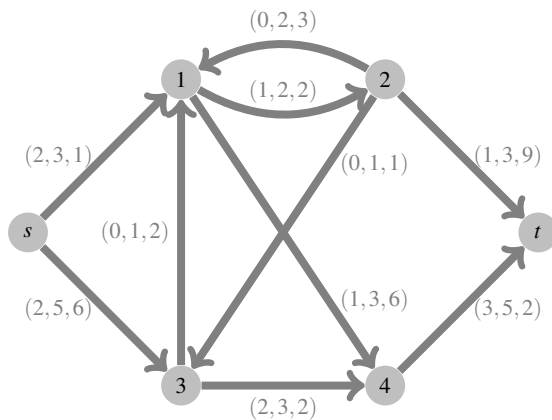


Abb. 5.97 Fluss f^1 in Flussnetzwerk F

Residualgraph $G^r(F, f^1)$ in der zweiten Iteration ist in Abb. 5.98 dargestellt. Wiederum ist ein Kreis negativer Kosten schwarz hervorgehoben.

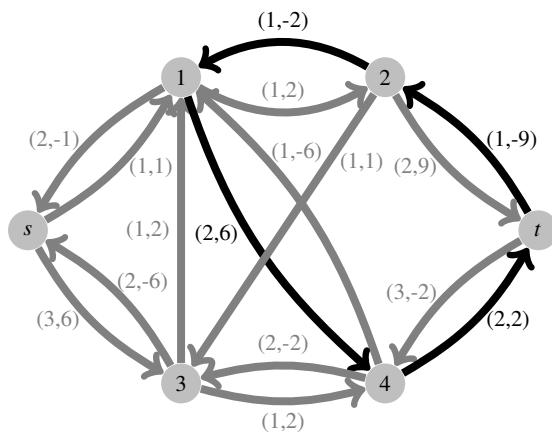


Abb. 5.98 Residualgraph $G^r(F, f^1)$ und Kreis negativer Kosten

Wir wenden den Kreis in Abb. 5.98 auf Fluss f^1 in Abb. 5.97 an und erhalten Fluss f^2 in Abb. 5.99.

In zwei weiteren Iterationen werden die Residualgraphen $G^r(F, f^2)$ und $G^r(F, f^3)$ in Abb. 5.100 und 5.102 ermittelt, wobei Kreise negativer Kosten jeweils schwarz hervorgehoben sind. Wir wenden die Kreise in Abb. 5.100 und 5.102 dann auf die Flüsse f^2 und f^3 in Abb. 5.99 und 5.101 an und erhalten die Flüsse f^3 und f^4 in Abb. 5.101 und 5.103.

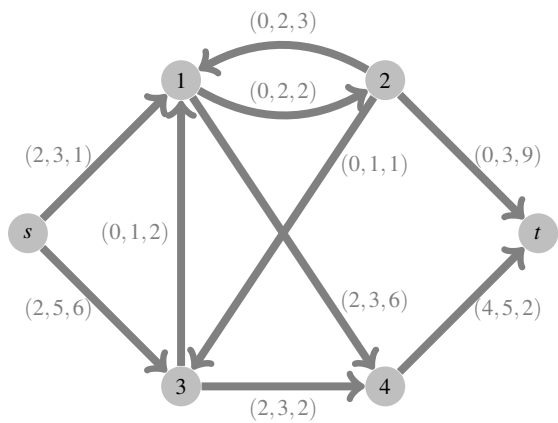


Abb. 5.99 Fluss f^2 in Flussnetzwerk F

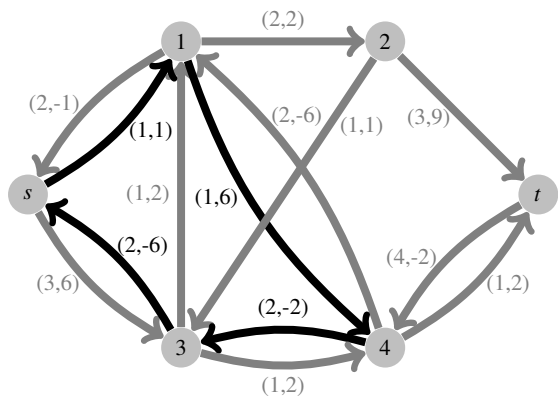


Abb. 5.100 Residualgraph $G^r(F, f^2)$ und Kreis negativer Kosten

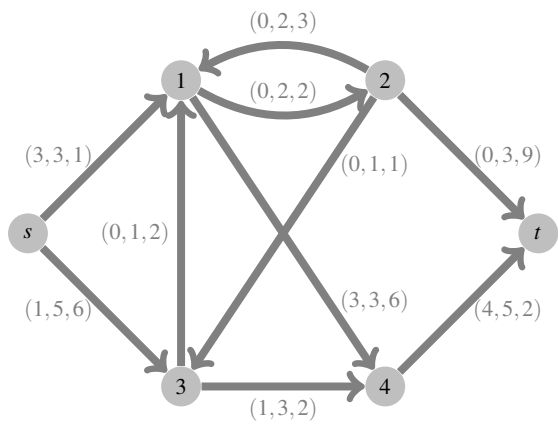


Abb. 5.101 Fluss f^3 in Flussnetzwerk F

Residualgraph $G^r(F, f^4)$ in der fünften Iteration ist in Abb. 5.104 dargestellt. Es gibt keinen Kreis negativer Kosten und somit ist f^4 ein kostenminimaler Fluss mit Wert $b = 4$.

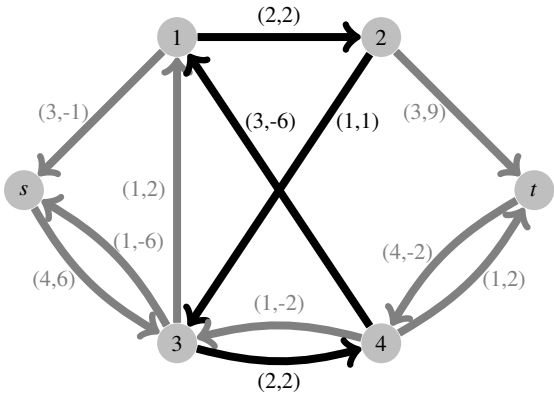


Abb. 5.102 Residualgraph $G^r(F, f^3)$ und Kreis negativer Kosten

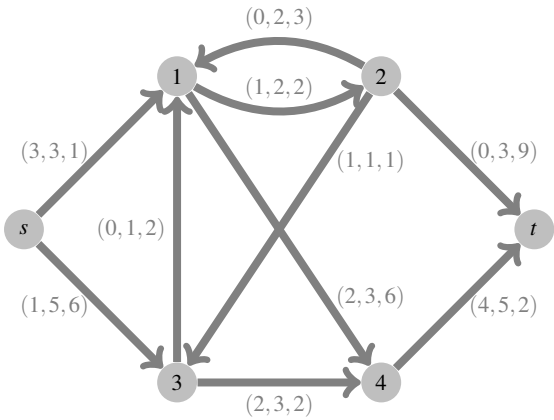


Abb. 5.103 Fluss f^4 in Flussnetzwerk F

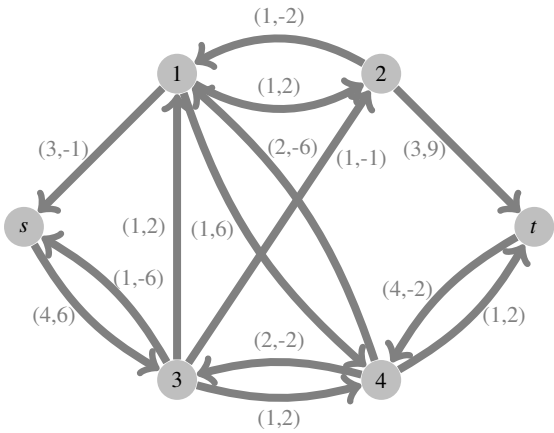


Abb. 5.104 Residualgraph $G^r(F, f^4)$

Analyse

Nachdem wir gesehen haben, dass Algorithmus 5.11 die Probleminstanz in Abb. 5.95 von Problem 5.7 (Fluss minimaler Kosten) löst, besagt Theorem 5.15, dass er dies tatsächlich für jede Instanz von Problem 5.7 (Fluss minimaler Kosten) leistet.

Theorem 5.15. *Algorithmus 5.11 bestimmt einen Fluss minimaler Kosten mit Wert b , wenn es einen solchen Fluss gibt und alle Kantengewichte rational sind.*

Den Beweis führen wir in Anhang C.12.

5.4.2.2 Versenden entlang kostenminimaler Wege

Das Verfahren, das in diesem Abschnitt vorgestellt wird, geht auf Busacker und Gowen (1960), Iri (1960) und Jewell (1958) zurück. Es ist nur anwendbar, wenn es keine negativen Kantenkosten gibt. Dies ist ein Spezialfall von Problem 5.7 (Fluss minimaler Kosten), wir werden allerdings sehen, dass wir hierbei nur solche Instanzen ausschließen müssen, in denen es Kreise negativer Kosten im Flussnetzwerk gibt. Alle übrigen Instanzen (also solche mit negativen Kantenkosten, aber ohne Kreise negativer Kosten) können wir in äquivalente Instanzen überführen, in denen es keine negativen Kantenkosten gibt.

Bevor wir uns mit dem eigentlichen Verfahren beschäftigen, wollen wir uns mit dieser Modifikation der Instanzen befassen. Wir ermitteln zunächst die minimalen Kosten unter allen Wegen von s nach i für jeden Knoten $i \in V$. Wir können zur Ermittlung dieser Wege z. B. Algorithmus 5.2, vorgestellt in Abschnitt 5.1.2, nutzen, wobei wir die Kosten einer Kante als Länge interpretieren. Da wir annehmen, dass es keine Kreise negativer Kosten gibt, gibt es entweder gar keinen Weg von s nach i oder einen kostenminimalen Pfad. Alle Knoten, zu denen es keinen Weg von s gibt, und die inzidenten Kanten entfernen wir aus dem Flussnetzwerk und erhalten so die Knoten- und Kantenmenge \bar{V} und \bar{E} . Für alle verbleibenden Knoten bezeichnen wir die minimalen Kosten unter allen Pfaden von s nach i mit c_i . Für jede Kante $(i, j) \in \bar{E}$ gilt dann $c_j - c_i \leq c_{i,j}$. Wir betrachten die modifizierten Kosten

$$\bar{c}_{i,j} = c_{i,j} + c_i - c_j \geq 0$$

für jede Kante $(i, j) \in \bar{E}$ und verwenden im Weiteren die modifizierte Instanz von Problem 5.7 (Fluss minimaler Kosten), die durch Flussnetzwerk $\bar{F} = (\bar{V}, \bar{E}, (w, \bar{c}), s, t)$ und Wert b spezifiziert ist. Ein Fluss \bar{f} in Flussnetzwerk \bar{F} mit Wert b und ein Fluss f in Flussnetzwerk F mit Wert b entsprechen sich unmittelbar, wenn $f_{i,j} = 0$ für jede Kante $(i, j) \in E \setminus \bar{E}$ und $f_{i,j} = \bar{f}_{i,j}$ jeder Kante $(i, j) \in \bar{E}$.

Theorem 5.16. *Ein Fluss \bar{f}^* mit Wert b in Flussnetzwerk $\bar{F} = (\bar{V}, \bar{E}, (w, \bar{c}), s, t)$ mit minimalen Kosten (gemäß \bar{c}) entspricht einem Fluss f^* mit Wert b in Flussnetzwerk $F = (V, E, (w, c), s, t)$ mit minimalen Kosten (gemäß c).*

Wir diskutieren ein Beispiel und führen den Beweis in Anhang C.13. In einem Verfahren zum Ermitteln eines kostenminimalen Flusses mit Wert b können wir also

zunächst die Transformation von F in \bar{F} vornehmen, wenn es keine Kreise negativer Kosten in F gibt. Anschließend ermitteln wir einen kostenminimalen Fluss \bar{f}^* mit Wert b in \bar{F} und wissen durch Theorem 5.16, dass er einem kostenminimalen Fluss f^* mit Wert b in F entspricht. Theorem 5.16 rechtfertigt daher, dass wir uns im Folgenden auf die Betrachtung von Flussnetzwerken konzentrieren, in denen keine Kante negative Kosten hat, wenn wir uns auf Flussnetzwerke ohne Kreise negativer Kosten beschränken.

Das Verfahren, das in diesem Abschnitt vorgestellt wird, konstruiert zunächst einen kostenminimalen Fluss mit Wert null und modifiziert diesen sukzessiv, um einen kostenminimalen Fluss mit größerem Wert zu erhalten. Die Modifikation endet, wenn der aktuelle Fluss einen Wert von b hat. In gewisser Weise ist die Vorgehensweise also komplementär zu der in Abschnitt 5.4.2.1, denn dort haben wir zunächst den Wert des Flusses sichergestellt und dann die Kosten reduziert.

Überblick über das Verfahren

Zunächst wird ein Fluss f mit Wert null bestimmt; dies geschieht durch Setzen von $f_{i,j} \leftarrow 0$ für jede Kante $(i,j) \in E$. In jeder Iteration wird dann der Residualgraph $G^r(F, f)$ für F und den aktuellen Fluss f konstruiert und ein Weg minimaler Kosten von s nach t in $G^r(F, f)$ ermittelt. Dies kann durch Anwendung von Algorithmus 5.2, vorgestellt in Abschnitt 5.1.2, erreicht werden, wobei die Kosten als Längen der Kanten in $G^r(F, f)$ interpretiert werden. Sollte es keinen Weg von s nach t in $G^r(F, f)$ geben, dann können wir den Wert von Fluss f nicht vergrößern, und wenn darüber hinaus $v(f) < b$, dann gibt es keinen Fluss mit Wert b in F . Ansonsten wenden wir den ermittelten Weg auf Fluss f an und erhalten Fluss f' mit $v(f') > v(f)$. Hierbei achten wir darauf, dass der Wert von f' nicht größer als b wird. Falls $v(f') = b$, ist der vorliegende Fluss f' ein kostenminimaler Fluss mit Wert b und das Verfahren endet. Ansonsten wiederholen wir den Schritt mit f' als vorliegendem Fluss.

Salopp formuliert: Versenden entlang kostenminimaler Wege

Wir starten mit dem Nullfluss und erhöhen sukzessiv den Wert des vorliegenden Flusses durch Anwendung des kostenminimalen wertsteigernden Weges bis der vorliegende Fluss Wert b hat.

Das gesamte Verfahren lässt sich als Algorithmus 5.12 konkretisieren.

Algorithmus 5.12 (Kostenminimale Wege).

Eingabe: Flussnetzwerk $F = (V, E, (w, c), s, t)$ mit $c_{i,j} \geq 0$ für jede Kante $(i, j) \in E$, Wert b

Ausgabe: Fluss minimaler Kosten mit Wert b oder Information, dass keiner existiert

1. Setze $f_{i,j} = 0$ für jede Kante $(i, j) \in E$.
2. Wenn $v(f) = b$, dann ist f ein Fluss minimaler Kosten mit Wert b . STOP.
3. Konstruiere Residualgraph $G^r(F, f) = (V, E', (w', c'))$.
4. Bestimme mittels Algorithmus 5.2 einen Weg P von s nach t in $G^r(F, f)$ mit minimalen Kosten.
5. Wenn es keinen Weg von s nach t in $G^r(F, f)$ gibt, gibt es keinen Fluss mit Wert b in F . STOP.
6. Wenn es einen Weg von s nach t in $G^r(F, f)$ gibt, modifiziere f unter Verwendung von P (ohne dabei einen Fluss mit Wert größer b zu erzeugen).
7. Gehe zu 2.

Beispiel

Bevor wir das Verfahren analysieren, wollen wir es an einem Beispiel veranschaulichen. Wir greifen das Beispiel aus Abschnitt 5.4.2.1 auf und wenden Algorithmus 5.12 auf das Flussnetzwerk $F = (V, E, (w, c), s, t)$ in Abb. 5.105 an. Der geforderte Wert des zu findenden Flusses ist $b = 4$. Den Ablauf des Verfahrens stellen wir mittels Abb. 5.106 bis 5.111 dar.

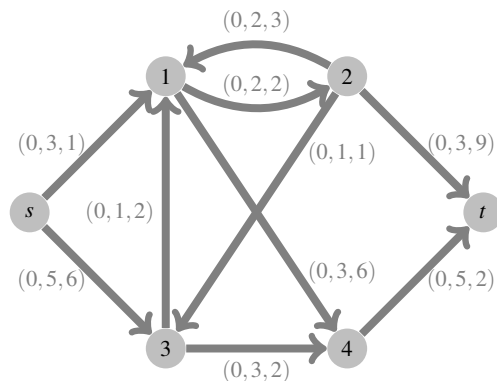


Abb. 5.105 Fluss f^0 in Flussnetzwerk F

Der erste Fluss f^0 ist in Abb. 5.105 dargestellt. Da $v(f^0) = 0 < 4$, konstruieren wir Residualgraph $G^r(F, f^0)$ in der ersten Iteration wie in Abb. 5.106 illustriert. Durch „scharfes Hinsehen“ können wir den schwarz hervorgehobenen Weg P^1 als kostenminimalen Weg von s nach t in dem Residualgraph erkennen. Alternativ könnten wir Algorithmus 5.2 einsetzen.

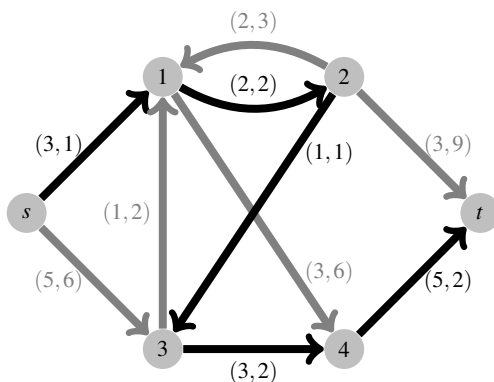


Abb. 5.106 Residualgraph $G^r(F, f^0)$ und kostenminimaler Weg P^1

Wir wenden Weg P^1 in Abb. 5.106 auf Fluss f^0 in Abb. 5.105 an und erhalten Fluss f^1 in Abb. 5.107. Fluss f^1 hat einen Wert von $v(f^1) = 1$ Kosten von $c(f^1) = 8$, was den gesamten Kosten der Kanten in Weg P^1 entspricht.

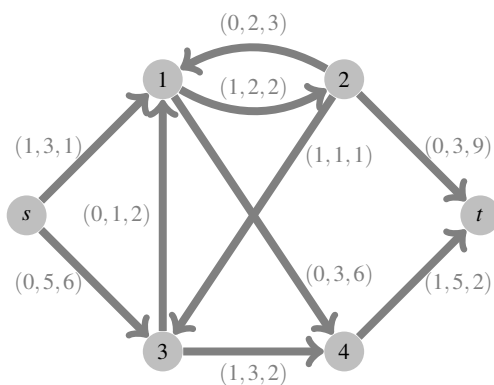


Abb. 5.107 Fluss f^1 in Flussnetzwerk F

Residualgraph $G^r(F, f^1)$ in der zweiten Iteration ist in Abb. 5.108 dargestellt. Wiederum ist ein kostenminimaler Weg P^2 von s nach t schwarz hervorgehoben.

Wir wenden Weg P^2 in Abb. 5.108 auf Fluss f^1 in Abb. 5.107 an und erhalten Fluss f^2 in Abb. 5.109. Die Kosten des Flusses f^2 sind um das Doppelte der gesamten Kosten der Kanten in Weg P^2 in Abb. 5.108 höher als die Kosten des Flusses f^1 , da wir über Weg P^2 zwei Einheiten versenden und somit $v(f^2) = 3$ und $c(f^2) = 26$ erhalten.

Residualgraph $G^r(F, f^2)$ in der dritten Iteration ist in Abb. 5.110 dargestellt. Wiederum ist ein kostenminimaler Weg P^3 von s nach t schwarz hervorgehoben.

Wir wenden Weg P^3 in Abb. 5.110 auf Fluss f^2 in Abb. 5.109 an und erhalten Fluss f^3 in Abb. 5.111.

Wir versenden bei der Anwendung von Weg P^3 nur eine Einheit über P^3 (obwohl wir zwei Einheiten versenden könnten), weil nur eine Einheit zu einem Fluss mit

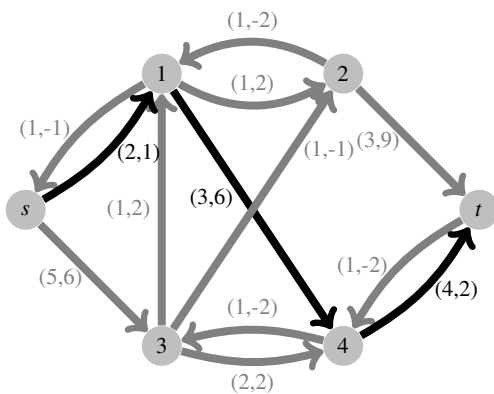


Abb. 5.108 Residualgraph $G^r(F, f^1)$ und kostenminimaler Weg P^2

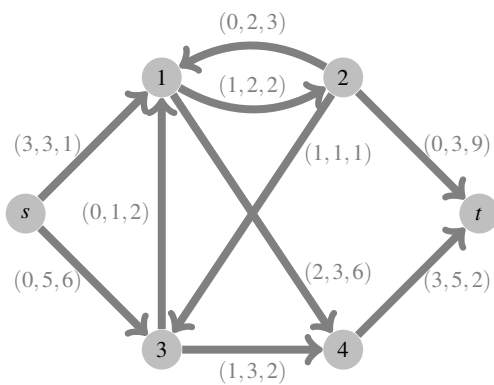


Abb. 5.109 Fluss f^2 in Flussnetzwerk F

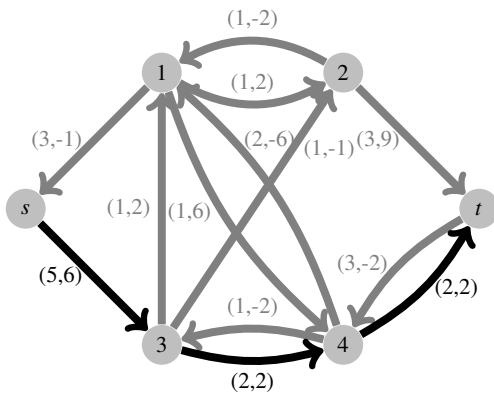


Abb. 5.110 Residualgraph $G^r(F, f^2)$ und kostenminimaler Weg P^3

Wert $b = 4$ fehlt. Fluss f^3 ist kostenminimal unter alle Flüßen mit Wert $v(f^3) = b = 4$ und hat Kosten von $c(f^3) = 36$. Da $v(f^3) = b$ endet das Verfahren.

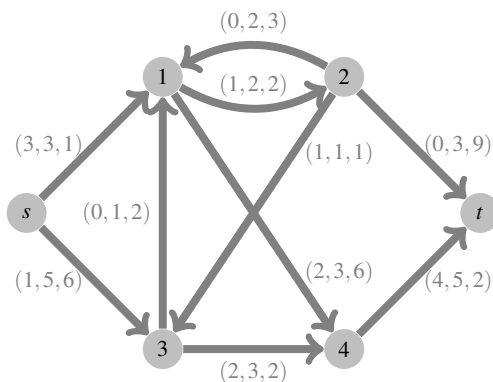


Abb. 5.111 Fluss f^3 in Flussnetzwerk F

Analyse

Nachdem wir gesehen haben, dass Algorithmus 5.12 die Problem Instanz in Abb. 5.105 von Problem 5.7 (Fluss minimaler Kosten) löst, besagt Theorem 5.17, dass er dies tatsächlich für jede Instanz von Problem 5.7 (Fluss minimaler Kosten) leistet.

Theorem 5.17. *Algorithmus 5.12 bestimmt einen Fluss minimaler Kosten mit Wert b , wenn es einen solchen Fluss gibt, alle Kantengewichte rational sind und Kantenkosten nicht negativ sind.*

Den Beweis führen wir in Anhang C.14. Dort wird auch ersichtlich, dass wir annehmen können, dass die kostenminimalen wertsteigernden Wege, die wir ermitteln, Pfade sind. Wir haben darauf bisher keinen Wert gelegt, da es für unsere Vorgehensweise keine Bedeutung hat, wollen es an dieser Stelle aber festhalten.

Literaturverzeichnis

- Berge C (1957) Two theorems in graph theory. Proceedings of the National Academy of Sciences of the United States of America 43:842–844
- Busacker R, Gowen P (1960) A procedure for determining a family of minimum-cost network flow pattern. Technical Report ORO-TP-15, Operations Research Office, The Johns Hopkins University, Bethesda, MD
- Diestel R (2017) Graphentheorie, 5. Aufl. Springer, Berlin u. a.
- Dijkstra EW (1959) A note on two problems in connexion with graphs. Numerische Mathematik 1:269–271
- Edmonds J (1965a) Matching and a polyhedron with 0,1 vertices. J of Res the Nat Bureau of Standards 69 B:125–130
- Edmonds J (1965b) Paths, trees, and flowers. Canadian Journal of Mathematics 17:449–467
- Floyd R (1962) Algorithm 97: Shortest path. Communications of the ACM 5:345

- Ford Jr LR, Fulkerson DR (1956) Maximal flow through a network. *Canadian Journal of Mathematics* 8:399–404
- Galil Z (1986) Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys* 18:23–38
- Iri M (1960) A new method for solving transportation-network problems. *Journal of the Operations Research Society of Japan* 3:27–87
- Jewell W (1958) Optimal flow through networks. Interim Technical Report No 8, Operations Research Center, MIT, Cambridge, MA
- Jungnickel D (2013) *Graphs, Networks and Algorithms*, 4. Aufl. Springer, Berlin u. a.
- Klein M (1967) A primal method for minimal cost flows with applications to the assignment and transportation problems. *Management Science* 14:205–220
- Kruskal J (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7(1):48–50
- Schrijver A (2002) On the history of the transportation and maximum flow problems. *Mathematical Programming* 91:437–445
- Wallis WD (2007) *A Beginner's Guide to Graph Theory*, 2. Aufl. Birkhäuser Boston
- Warshall S (1962) A theorem on boolean matrices. *Journal of the ACM* 9:11–12
- Zwicky U (1995) The smallest networks on which the ford-fulkerson maximum flow procedure may fail to terminate. *Theoretical Computer Science* 148:165–170