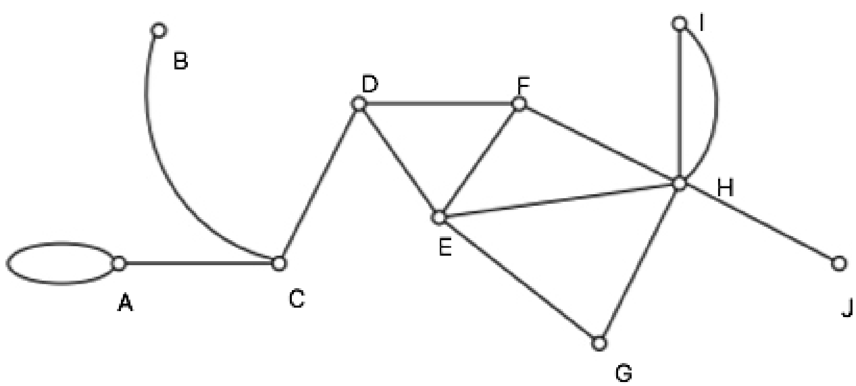
Bonner Mathematikturnier 2018 Vorbereitungsmaterial

How to get the best deal and make that flow great again

So fantastic! Believe me!

Lineare Programmierung und Optimierung auf Graphen



Bonner Mathematikturnier 2018
Vorbereitungsmaterial

Lieber Teilnehmerinnen und Teilnehmer des Bonner Mathematikturniers 2018,

Das Bonner Mathematiktunier besteht aus zwei Runden: der **Staffel** und der **Sum of Us**. Der Teil *Sum of Us* hat dieses Jahr das Thema **Lineare Programmierung und Optimierung auf Graphen**. Die vorliegenden Unterlagen sind für dich bestimmt, sodass du dich schon mal in das Thema einarbeiten kannst.

In den verschiedenen Abschnitten des Materials wird stets erst ein theoretisches Kapitel behandelt, um danach einige darauf aufbauende Übungsaufgaben anzuschließen. Die Lösungen für diese Übungsaufgaben werden vor Beginn des Turnieres auf der Internetseite
www.mathematics.uni-bonn.de/mathematik-in-bonn/schulportal/bonner-mathematikturnier-2018
zu finden sein.

In der *Sum of Us* wirst du zusammen mit deinem Team dann Aufgaben lösen, die sich auf die Themen des Vorbereitungsmaterials beziehen. Hierbei darfst du das Vorbereitungsmaterial, die Übungsaufgaben und deren Lösungen benutzen. Wegen der begrenzten Zeit raten wir dir, das Material vorher gründlich durchzuarbeiten. So sind dir die Aufgaben der *Sum of Us* schneller und besser zugänglich. Außerdem darfst du während der *Sum of Us* einen Taschenrechner verwenden, aber es darf kein graphikfähiger oder programmierbarer Taschenrechner oder einer mit CAS sein! Du solltest zum Wettbewerb unbedingt auch ein Geodreieck und Zeichenstifte mitbringen.

Die Aufgaben und das Vorbereitungsmaterial wurden von Carlijn Mulders (Studentin der Mathematik and der Radboud University Nijmegen) unter Begleitung von Wadim Zudilin erstellt. Die Übersetzung ins Deutsche wurde von Leoni Winschermann entworfen und von Carl Peter Fitting an die deutsche Situation angepasst.

Viel Erfolg!

Inhalt

1 Einleitung

So ziemlich alle Betriebe haben eine Logistikabteilung. In dieser Abteilung schaut man auf die Planung, Ausführung und Kontrolle hinsichtlich der Lagerung und des Transportes von Gütern. Dabei ist es wichtig, dass Grundstoffe und Zeit so effektiv wie möglich verwendet und so wenig wie möglich verschwendet werden. Die Logistikabteilung eines Wasserwerks kann zum Beispiel bestimmen, wie man am besten alle Haushalte mit der jeweils benötigten Wassermenge versorgen kann, und ein Vertriebszentrum kann berechnen, wie es die Produkte so schnell wie möglich an die Supermärkte liefern kann.

Wegen der großen Vielfalt von Anwendungen, gibt es auch mehrere mathematische Modelle für die unterschiedlichen Arten von logistischen Problemen. In der Realität sind diese oft sehr komplex; wenn wir hier einige Aspekte betrachten, so gehen wir bewusst von Elementarisierungen aus.

In den vorliegenden Unterlagen beschäftigen wir uns zunächst mit der Methode der *linearen Programmierung*. Diese Methode ist vor allem geeignet um zu bestimmen wie du eine begrenzte Menge Zeit, Raum, Maschinen, etc. am besten verwenden kannst. Danach betrachten wir das Prinzip der *Optimierung auf Graphen*. Graphen sind ein gutes Hilfsmittel um Wegverbindungen, Weglängen und andere Transportvorgaben schematisch darzustellen. Das Kapitel über Graphenoptimierung werden wir beginnen, indem wir zuerst den Begriff des Graphen in mathematisch zugänglicherer Form fassen und dann zwei verschiedene Methoden zur Optimierung mit Blick auf verschiedene Optimierungsziele behandeln.

2 Lineare Programmierung

2.1 Theorie

Lineare Programmierung ist eine der einfachsten Arten der Optimierung. Sie wird verwendet um eine optimale Lösung für Probleme zu finden, bei denen das als das "bestes Resultat" betrachtete Ergebnis unter Beachtung von einschränkenden Vorgaben hinsichtlich der Ressourcen erzielt werden muss. Diese Probleme stammen oft aus dem echten Leben: Anfertigen eines Dienstplans für alle Mitarbeiter einer Firma, Herausfinden wie Maschinen am effizientesten eingesetzt werden können, Beschließen, welche Gewächse ein Bauer anbauen sollte, etc. Bei dieser Methode erfasst man alle relevanten Zusammenhänge durch lineare Funktionen.

Definition: Eine *lineare Funktion* ist eine Funktion f von der Form $f(x) = ax + b$, wobei a und b konstante reelle Zahlen sind.

Anhand eines – bewusst einfachen und damit nicht wirklich realistischen – Beispiels werden wir sehen, wie das in der Praxis funktioniert. Dazu müssen wir fünf Schritte durchlaufen, die notwendig sind um die optimale Lösung für das Problem in unserem Beispiel zu finden. Dabei treten dann auch zentrale Begriffe auf, die für die allgemeine Beschreibung des Verfahrens bedeutsam sind.

Beispiel: Der neue Betrieb *Lovechoc* wird zum ersten Mal ungesüßte Bioschokolade produzieren. Man beschließt mit nur zwei Sorten zu beginnen, die wir A und B nennen. Beide Sorten bestehen ausschließlich aus Kakao und Milch:

- Zur Herstellung einer großen Platte der Sorte A braucht man 1 Liter Milch und 3 Kilo Kakao.
- Zur Herstellung einer großen Platte der Sorte B benötigt man 1 Liter Milch und 2 Kilo Kakao.

Die Platten werden an Konditoreien geliefert, die sie dann in Stücken an ihre Kunden verkaufen.

Der Betrieb hat momentan 5 Liter Milch und 12 Kilo Kakao. Glücklicherweise für die Eigentümer macht die Firma Gewinn auf jede verkaufte Schokoladenplatte:

- Eine Platte der Sorte A bringt 6 Euro Gewinn
- Eine Platte der Sorte B bringt 5 Euro Gewinn

Wir werden zunächst davon ausgehen, dass der Betrieb auch Anteile von ganzen Platten für den proportionalen Anteil des Gesamtpreises verkaufen kann: eine halbe Tafel A bringt 3 Euro Gewinn, ein Fünftel von Tafel B bringt 1 Euro, etc. Um das Ganze etwas übersichtlicher zu machen, können wir die Situation auch in eine Tabelle eintragen:

	Milch	Kakao	Gewinn
A	1	3	6
B	1	2	5
Gesamt	5	12	

Tabelle 1: Übersicht der Daten aus dem Beispiel

Frage: Wie viele Platten der Sorten A und B muss *Lovechoc* herstellen, wenn der Betrieb möglichst viel Gewinn machen möchte?

Lösung:
Schritt 1: Identifiziere die Entscheidungsvariablen und die Zielvariable
Der Einfachheit halber führen wir drei neue Variablen ein:

- Die Menge hergestellter Tafeln der Sorte A = X
- Die Menge hergestellter Tafeln der Sorte B = Y
- Der erzielte Gewinn (in Euros) = Z

Der gesamte Gewinn, Z, hängt davon ab, was produziert wird. Mit anderen Worten: Z ist abhängig von X und Y. Das bedeutet, dass X und Y unsere Entscheidungsvariablen sind und Z die Zielvariable.

Schritt 2: Aufstellen der Hauptbedingung
Da eine Tafel A einen Gewinn von 6 Euro einbringt und eine Tafel B insgesamt 5 Euro, können wir die folgende Funktion ermitteln für den Gewinn:

$$Z = 6X + 5Y$$

Diese lineare Funktion ist die Hauptbedingung, die zu unserem Beispiel gehört.

Die *Hauptbedingung* ist also eine Funktion, die allen möglichen möglichen Konstellationen der Entscheidungsvariablen jeweils einen Wert zuordnet, der den Gewinn oder die Kosten repräsentiert. In diesem Beispiel soll die Zielvariable einen maximalen Wert annehmen; in anderen Beispiel könnte es um die minimalen Kosten bei der Produktion gehen: dann würde die Hauptbedingung eine zu minimalisierende Zielvariable beschreiben.

Schritt 3: Aufstellen der Nebenbedingungen
Nachdem wir nun die Hauptbedingung ermittelt haben, müssen wir jetzt noch die vorgegebenen Einschränkungen berücksichtigen und daraus lineare Nebenbedingungen formulieren.

Eine *Nebenbedingung* ist also jede Bedingungen, die die Entscheidungsvariablenariablen erfüllen müssen, um als mögliche Lösung des anstehenden Optimierungsproblems in Frage zu kommen.

Zunächst wissen wir, dass die Herstellung einer Platte A als auch einer Platte B genau 1 Liter Milch brauchen. Insgesamt haben wir aber nur 5 Liter zur Verfügung. Das gibt uns die Ungleichung

$$X + Y \leq 5$$

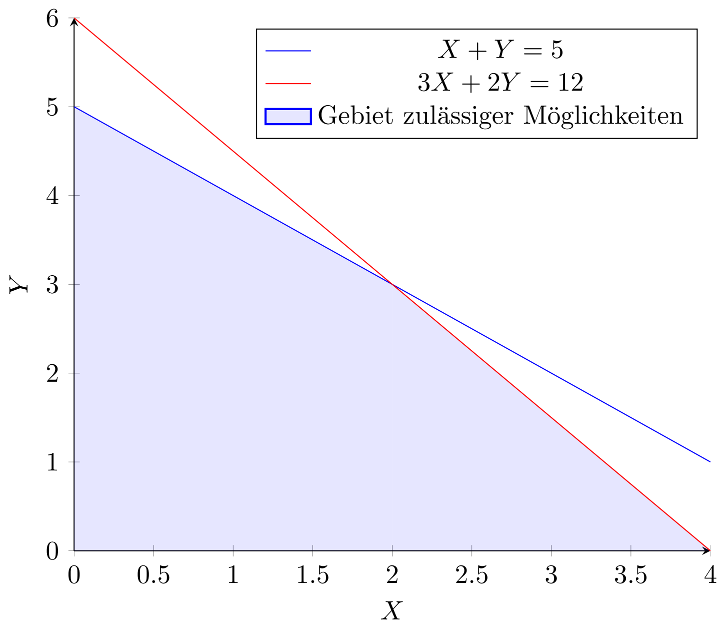
Außerdem gilt, dass wir für A pro Platte 3 Kilo Kakao und für B pro Platte 2 Kilo Kakao benötigen. Die Gesamtmenge Kakao, die uns zur Verfügung steht, ist 12 Kilo. Das gibt uns:

$$3X + 2Y \leq 12$$

Zuletzt wissen wir, dass wir keine negativen Mengen produzieren können, weshalb gilt, dass

$$X \geq 0$$
$$Y \geq 0$$

Schritt 4: Erstellen einer Graphik
Mit Hilfe unserer Nebenbedingungen wissen wir, dass X und Y nicht einfach willkürlich gewählt werden können: sie schränken die Wahl der Entscheidungsvariablen natürlich ein. Um das zu verdeutlichen, repräsentieren wie die Gleichungen $X + Y = 5$ (oder $Y = 5 - X$) und $3X + 2Y = 12$ (oder $Y = 6 - 1.5X$) in einer gemeinsamen Graphik. Wegen der in *Schritt 3* aufgestellten Ungleichungen wissen wir, dass wir nur noch das Gebiet betrachten müssen, das unter beiden Geraden liegt, über der x-Achse und rechts von der y-Achse. Einzig die x- und y-Koordinaten derjenigen Punkte, die in diesem Gebiet liegen, erfüllen die Nebenbedingungen.



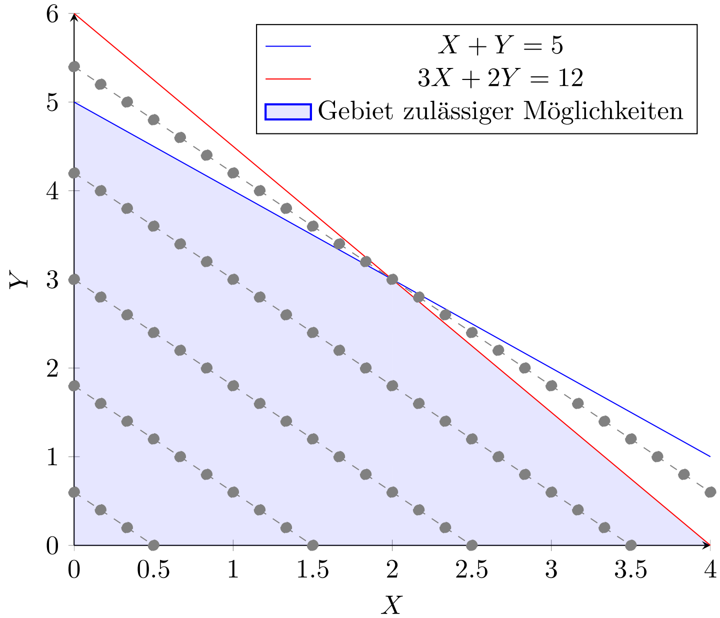
Schritt 5: Ablesen der Graphik (Option 1: es sind auch alle nichtganzzahligen Werte für die Entscheidungsvariablen erlaubt)
Mit Hilfe der oben stehenden Graphik sehen wir nicht nur das Gebiet aller zulässigen Möglichkeiten, sondern können sogar ablesen, wie viele Platten A und B *Lovechoc* jeweils herstellen muss, um den Gewinn zu maximieren.

Das Gebiet der zulässigen Werte ist als Schnittmenge von Halbebenen ein kovexes Vieleck. Dort, wo sich zwei Seiten dieses Vielecks schneiden, liegt einer der Eckpunkte des Vielecks: wir sprechen von *zulässigen Eckpunkten*.

In unserem Beispiel gibt es vier zulässige Eckpunkte, nämlich $(0, 0), (0, 5), (4, 0)$ und $(2, 3)$.

Der **Hauptsatz der linearen Optimierung** besagt nun: Wenn ein lineares Optimierungsproblem eine Lösung hat, dann ist diese entweder durch einen zulässigen Eckpunkt oder durch Punkte auf einer Randstrecke (zwischen zwei zulässigen Eckpunkten) des zulässigen Vielecks gegeben.

Der Beweis dieses Satzes ist ziemlich abstrakt und wir gehen nicht weiter darauf ein. Aber mit Hilfe einer Abbildung und unseres Beispieles wird es intuitiv deutlicher, dass der Satz gilt. Wir betrachten noch einmal den Gewinn $Z = 6X + 5Y$, den wir maximieren möchten. Wir können die Gleichung umstellen zu einer Geradengleichung $Y = \frac{1}{5}Z - \frac{6}{5}X$. In dieser Gleichung betrachten wir Z als Konstante, die wir so groß wie möglich festlegen wollen, wobei die Gerade noch Punkte mit dem Gebiet zulässiger Lösungen gemeinsam hat. Wir finden dann in der Graphik eine Schar paralleler Geraden mit der Steigung $-\frac{6}{5}$ und dem von Z abhängigen y-Achsenabschnitt $\frac{1}{5} * Z$. Das sieht dann so aus:



Je größer Z wird, desto weiter entfernt sich die graue Gerade vom Ursprung: der y-Achsenabschnitt der Gerade nimmt zu. Die Gerade mit dem größten y-Achsenabschnitt gehört demnach zum höchsten Wert für den Gewinn. Sie liegt in nur einem einzigen Punkt noch im Gebiet der zugelassenen Möglichkeiten: in einem Eckpunkt.

Beinahe immer gibt es mehrere Eckpunkte (wie eher bereits erwähnt sind es in unserem Fall vier). Die "bewegliche" Gerade unserer Hauptbedingung ist ein gutes Hilfsmittel für die Ermittlung des Eckpunktes, der tatsächlich die gesuchte Lösung gibt. Es gilt nämlich, dass der Eckpunkt, der *zuletzt* von einer Gerade berührt wird, der Eckpunkt mit der optimalen Lösung zu deinem Problem ist. Sollte ein Minimum gesucht werden, willst du Z möglichst klein finden. In dem Fall suchst du dann den Eckpunkt, der die bewegende Gerade *zuerst* berührt.

Wenn das zulässige Gebiet von einer Gerade mit derselben Steigung wie die "bewegliche" Gerade begrenzt ist, dann ist es möglich, dass zwei verschiedene Eckpunkte und ihre ganze Verbindungsstrecke die Gerade zuletzt berühren. In dem Fall gilt, dass sowohl beide Eckpunkte, als auch alle Punkte auf der Strecke zwischen den beiden Eckpunkten Lösungen darstellen.

Nun haben wir das Vorwissen um unsere Frage zu beantworten, wie viele Platten A und B *Lovechoc* für maximalen Gewinn produzieren muss. Mit Hilfe der Graphik wissen wir nämlich, dass der Schnittpunkt der roten und blauen Gerade den besten Wert für X und Y angibt. Diesen Schnittpunkt könnten wir einfach ablesen, aber es ist genauer ihn auszurechnen. Darum bevorzugen wir immer Letzteres. Wir müssen die

Gleichungen X + Y = 5 und 3X + 2Y = 12 gleichstellen. Das geht wie folgt:

$$X + Y = 5 \longrightarrow Y = 5 - X$$

Einsetzen in $3X + 2Y = 12$ gibt:

$$3X + 2(5 - X) = 12 \longrightarrow X = 2$$

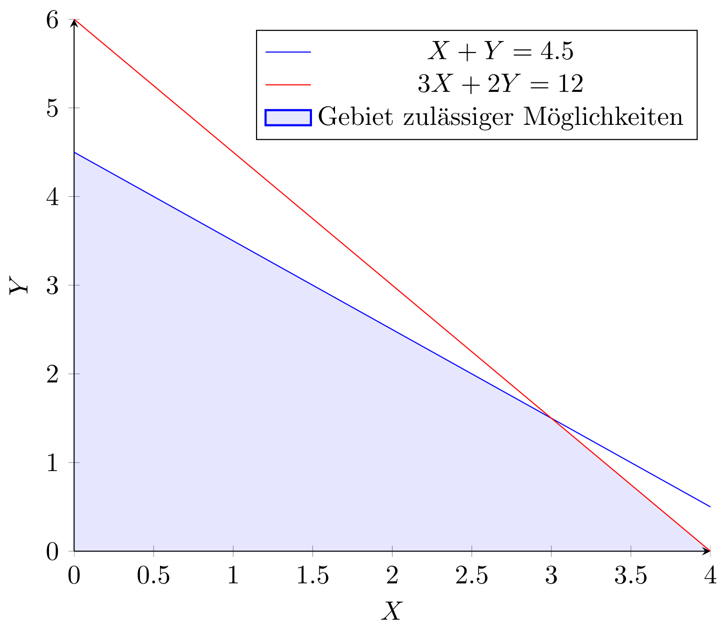
Das wiederum einsetzen in $Y = 5 - X$ gibt:

$$Y = 3$$

Kurz gesagt, *Lovechoc* muss 2 Platten der Sorte A und 3 Platten der Sorte B produzieren, um den maximalen Gewinn von $Z = 6 * 2 + 5 * 3 = 27$ Euro zu erzielen.

Schritt 5: Ablesen der Graphik (Option 2: nur ganzzahlige Werte der Entscheidungsvariablen sind zulässig)
Zwar kommen Bruchzahlen fast immer als Antworten in Frage. aber in manchen Fällen ist das auch Sachgründen ausgeschlossen. Das könnte in unserem Beispiel dann der Fall sein, wenn *Lovechoc* nur ganze Tafeln herstellen und verkaufen kann.

Wir untersuchen die veränderte Fragestellung unter der veränderten Randbedingung, dass uns statt 5 Litern Milch nur 4,5 Liter zur Verfügung stehen. Das gibt uns die folgende Graphik:



Auf dieselbe Weise wie in *Schritt 5, Option 1* könnten wir bestimmen, welcher Eckpunkt den maximalen Wert angibt: dies wäre in (3, 1.5) der Fall. Das ist nun allerdings nun keine gültige Lösung, da wir keine halbe Tafel B produzieren können. Also müssen wir unsere Strategie anpassen. Mit Hilfe der Graphik können wir alle Punkte im zulässigen Vieleck, die ganzzahlige Koordinaten haben, auflisten und jeweils ausrechnen, welchen Wert wir für die Zielvariable erhalten:

X	Y	Z = 6X + 5Y
0	0	0
0	1	5
0	2	10
0	3	15
0	4	20
1	0	6
1	1	11
1	2	16
1	3	21
2	0	12
2	1	17
2	2	22
3	0	18
3	1	23
4	0	24

Tabelle 2: Tabelle, die pro Option den Gewinn angibt

Wir sehen hier, dass der Gewinn Z maximal ist, wenn wir 4 Tafeln der Sorte A machen und 0 Tafeln B. Der Gewinn beträgt in diesem Fall 24 Euro.

Die Schritte, die wir in diesem Beispiel durchlaufen haben, bilden zusammen die Methode der linearen Programmierung. Jene Schritte können immer auf derartige Fragestellungen angewendet werden. Allerdings ist unsere Darstellung darauf begrenzt, dass wir nur zwei Entscheidungsvariablen nutzen; bei drei Entscheidungsvariablen ergäbe sich analog ein zulässiger Polyeder statt eines Vielecks, und bei noch mehr Entscheidungsvariablen muss man die direkte geometrische Anschauung verlassen und die gewohnte Koordinatengeometrie zu höherdimensionalen Strukturen ausbauen. Und dennoch befruchtet die Erfahrung mit dem hier vorgestellten Material auch das Tun in diesen abstrakteren Strukturen.

2.2 Übungsaufgaben

1. (a) Maximiere die Zielvariable z = x_1+x_2 mit den folgenden Nebenbedingungen:
   $x_1 + x_2 \leq 10$
   $2x_1 + x_2 \leq 16$
   $-x_1 + x_2 \leq 3$
   $x_1 \geq 0$, $x_2 \geq 0$
   Verwende hierzu eine Graphik.

   (b) Maximiere die Zielvariable z = $2x_1$+$5x_2$ mit den folgenden Nebenbedingungen:
   $x_1 + x_2 \leq 6$
   $6x_1 + 2x_2 \leq 24$
   $4x_1 + x_2 \leq 8$
   $x_1 \geq 0$, $x_2 \geq 0$
   Verwende hierzu eine Graphik.

   (c) Minimiere die Zielvariable z = x_1+$3x_2$ mit den folgenden Nebenbedingungen:
   $x_1 + 2x_2 \geq 8$
   $x_1 - x_2 \leq 3$
   $2x_1 + x_2 \geq 10$
   $x_1 \geq 0$, $x_2 \geq 0$
   Verwende hierzu eine Graphik.

2. Ein Bäcker produziert zwei Produkte: Grau- und Weißbrot. Für beide Brote verwendet er lediglich Wasser und Mehl. Die Tabelle gibt für jede Brotsorte an, wie viel Mehl und Wasser benötigt werden. Auch der Umsatz (= Betrag der Kosten plus Betrag des Gewinns) pro Brot ist angegeben.

Produkt	Mehl (kg)	Wasser (Liter)	Umsatz (Euro)
Graubrot	2	1	20,-
Weißbrot	1	1	10,-

Tabelle 3: Tabelle zu Aufgabe 2

Da der Bäcker das beste Brot backen will, verwendet er herrliches Quellwasser statt Kranenwasser. Das kauft er in Flaschen von je 1 Liter zu einem Preis von 2 Euro. Die Kilosäcke Mehl kosten ihn 4 Euro pro Stück. Insgesamt stehen dem Bäcker 45 Euro zur Verfügung, die er investieren will. Außerdem muss er im Hinterkopf behalten, dass sein Brot frisch am besten schmeckt. Er will darum – um sicher zu gehen – insgesamt nicht mehr als 6 Brote backen.
Wie kann der Bäcker dafür sorgen, dass er so viel Gewinn wie möglich macht? Wie viele Graubrote und wie viele Weißbrote muss er dann backen? Er kann nur ganze Brote backen.

3. Ein Bauer hat einen schönen Obstgarten von 110 Hektar gekauft. Er hat beschlossen Äpfel und Birnen anzubauen. Er will entscheiden, wie viel Land er für Äpfel und wie viel Hektar er für Birnen verwenden sollte. In der folgenden Tabelle stehen einige Eckdaten, die ihm hierbei helfen können:

Obstsorte	Kosten (Euro/Hektar)	Nettogewinn (Euro/Hektar)	Arbeitsstunden (/Hektar)
Äpfel	100	50	10
Birnen	200	120	30

Tabelle 4: Tabelle zu Aufgabe 3

Der Bauer hat ein Startbudget von 10000 Euro und 5 Arbeitnehmer. Alle seine Arbeiter arbeiten 8 Stunden pro Tag und sind 30 Tage verfügbar. Wie kann er seinen Obstgarten am besten einrichten, wenn er so viel Gewinn wie möglich machen will?

4. Sara hat morgen einen stressigen Tag: sie hat eine Mathearbeit und eine mündliche Englischprüfung im Kalender stehen. Heute Abend ist aber auch eine coole Party geplant, zu der sie am liebsten so bald wie möglich hin möchte. Darum ist es wichtig, dass sie ihre Zeit so effizient wie möglich einteilt.

   Für die Mathearbeit hat sie verschiedene Aufgaben, die sie zum Üben durchrechnen kann. Pro Aufgabe braucht sie 9 Minuten. Für Englisch haben die Schüler Vokabeln aufbekommen. Es dauert 5 Minuten ein Wort perfekt zu erlernen.
   Sara glaubt, dass die Anzahl, der Aufgaben die sie macht plus zweimal die Anzahl der gelernten Vokabeln mindestens 40 betragen sollte. Außerdem soll die Anzahl der gelernten Worte plus das Doppelte der Anzahl der Aufgaben mindestens 30 sein. Zuletzt will sie noch, dass das Doppelte der Anzahl der gelösten Aufgaben höchstens 3 mehr ist als die Anzahl der erlernten Englischvokabeln.

   Wie kann Sara dafür sorgen, dass sie möglichst schnell zu der Party kann, ohne zu wenig gelernt zu haben? Wie lange muss sie mindestens lernen?

3 Optimierung auf Graphen

3.1 Was sind Graphen?

3.1.1 Theorie

Vor allem für Firmen, die sich mit dem Transport von Produkten oder Menschen beschäftigen (Paketzusteller, DB, Wasserwerke, etc.) kann es nützlich sein Graphen zu benutzen. Um ein Gefühl dafür zu bekommen, was ein Graph ist, schauen wir uns erst die folgende Abbildung an:

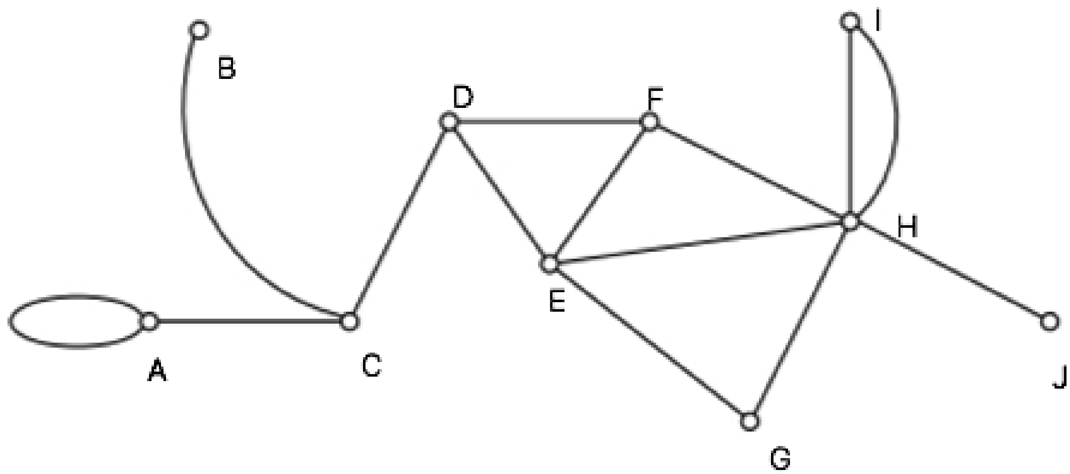


Abbildung 1: Beispiel für einen Graph

Ein Graph ist demnach eine schematische Wiedergabe von einer Menge von Punkten (zum Beispiel Häuser, Orte, Haltestellen, etc.), die durch Linien (beispielsweise Wege, Leitungen, Schienen, etc.) miteinander verbunden werden. Wir können hierzu auch eine formelle mathematische Definition geben.

Definition: Ein *Graph* G ist ein Paar (K,L), wobei K eine Menge mit endlich vielen Elementen ist und L eine Menge von Kombinationen von zwei Elementen aus K, wobei in einer solchen Kombination auch zweimal das gleiche Element von K vorkommen darf und es auf die Reihenfolge der Elemente nicht ankommt. Die Elemente aus K nennen wir *Knoten* von G und die Elemente aus L nennen wir die *Kanten* von G. Für jede Kante nennen wir die enthaltenen Knoten dann Randknoten der Kante. Die Kante verbindet also ihre Randknoten. Eine Kante, die einen Knoten mit sich selbst verbindet, heißt eine Schleife.

Wir werden noch einige Begriffe einführen, die später relevant sind.

Definition: Der *Grad* eines Knotens k ist die Anzahl der Kanten, von denen dieser Knoten k ein Randknoten ist. Schleifen werden hierbei doppelt gezählt: immerhin ist der Knoten ja zweimal Randknoten einer Schleife. Wir notieren den Grad eines Knotens mit gr(k).

Beispiel: In Abbildung 1 gilt $gr(A) = 3$, $gr(B) = 1$ und $gr(E) = 4$.

Definition: Ein *Pfad* zwischen zwei Knoten k und m ist eine Folge von Knoten, deren erstes Glied k und deren letztes Glied m ist (oder andersherum). Dabei sind je zwei aufeinanderfolgende Knoten durch eine Kante miteinander verbunden und keine Kante darf dabei doppelt vorkommen.

Beispiel: Zwei Pfade zwischen A und G in Abbildung 1 sind: A,C,D,E,G und A,C,D,F,H,I,H,E,G.

Definition: Ein Graph heißt *zusammenhängend*, wenn für je zwei verschiedene Knoten ein Pfad vom ersten zum zweiten existiert.

Der Graph in Abbildung 1 ist ein zusammenhängender Graph. Wenn die Kante (C,D) nicht im Graph wäre, dann wäre es kein zusammenhängender Graph, denn dann gäbe es keinen Pfad mehr von A nach E. Wir werden im Folgenden ausschließlich zusammenhängende Graphen betrachten.

Definition: Ein *Kreis* ist ein Pfad dessen erster Knoten gleich seinem letztenn Knoten ist.

Beispiel: In Abbildung 1 sehen wir den Kreis D,F,H,G,E,D.

Definition: Ein *Baum* ist ein zusammenhängender Graph, der keinen Kreis besitzt (siehe Abbildung 2).

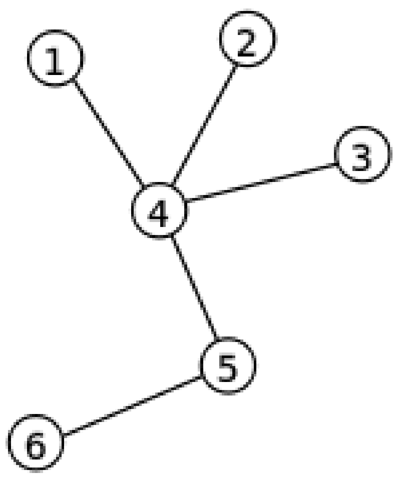


Abbildung 2: Beispiel eines Baumes

3.1.2 Übungsaufgaben

1. Gibt es einen zusammenhängenden Graphen ohne Schleifen und doppelte Kanten mit 8 Knoten und 6 Kanten? Wie viele Kanten sind mindestens notwendig, um einen Graphen zusammenhängend zu machen?

2. Wie viele verschiedene Pfade gibt es maximal, die zwei gegebene Knoten in einem Baum verbinden?

3. Der Baum in Abbildung 2 hat 6 Knoten und 5 Kanten. Wenn man eine weitere Kante zwischen den vorhandenen Knoten hinzufügt entsteht auf jeden Fall ein Kreis. Im Allgemeinen gilt, dass ein Graph mit n Knoten und mindestens n Kanten immer einen Kreis enthält. Das gibt uns den

Satz: Ein Baum mit n Knoten hat $n-1$ Kanten.
Dieser Satz kann auch formell bewiesen werden. Durchlaufe dazu die folgenden Schritte:

- Zeige, dass der Satz gilt wenn $n = 1$
- Setze nun voraus, dass du den Satz für einen Wert von n bereits als gesichert ansiehst. Dabei ist n kein von dir festzulegender Wert, sondern sollte - um seine Beliebigkeit zu sichern - als Variable behandelt werden. Beweise jetzt von dieser Annahme ausgehend, dass der Satz auch für jeden Baum mit $n+1$ Knoten gilt!

Diese Art der Beweisführung wird *vollständige Induktion* genannt. Wenn du den letzten Schritt erfolgreich abgeschlossen hast, hast du bewiesen, dass der Satz für alle natürlichen Zahlen gilt.

4. Betrachte einen Graph mit 5 Knoten, der weder Schleifen noch doppelte Kanten hat. Nimm an, dass die Grade der Knoten die folgenden sind: 4,2,2,1,1. Hat dieser Graph dann immer einen Kreis?

3.2 Der Algorithmus von Dijkstra

3.2.1 Theorie

Definition: Ein *Algorithmus* ist eine Folge von Schritten, die man nacheinander ausführen kann um zu einem bestimmten Resultat zu gelangen.

In der Graphentheorie gibt es eine Vielzahl von Algorithmen zu entdecken, von denen einige bei der Lösung von Optimierungsproblemen helfen. Einer dieser Algorithmen ist der Algorithmus von Dijkstra zur Ermittlung des kürzesten Pfades.

Definition: Ein Graph heißt *gewichtet*, wenn jeder Kante ein Gewicht zugewiesen ist: eine feste Zahl, die den Wert der Kante wiedergibt (beispielsweise ein Abstand zwischen ihren Randpunkten).
Bei einem gewichteten Graphen verstehen wir unter der Länge eines Pfades dann die Summe der Gewichte von allen Kanten, die im Pfad vorkommen.

Dijkstras Algorithmus soll die Aufgabe lösen, denn kürzeste Pfad für von einem gegebenen Ausgangsknoten zu einem gegebenen Endknoten zu bestimmen. Solche Probleme könnten etwa einen Zusteller betreffen: die Gewichte sind dann die tatsächlichen Streckenlängen von Verbindungswegen, und der Zusteller will den kürzesten Weg finden.

Die Idee von Dijkstras Algorithmus besteht darin, dass wir den Knoten sogenannte Labels zuweisen und diese unter Berücksichtigung der Streckenlängen immer möglichst klein halten. Wir unterscheiden dabei **vorläufige** (hellgraue) und **permanente** (dunkelgraue) Labels. Ein vorläufiges Label gibt den bis jetzt kürzesten Abstand vom Ausgangspunkt A bis zu betreffendem Knoten an. Die Schritte, die wir danach machen, können dafür sorgen, dass das vorläufige Label noch kleiner wird. Wir erklären ein vorläufiges Label als permanent, wenn wir festgestellt haben, dass es keinen kürzeren Pfad zu diesem Knoten gibt. Bevor wir den Algorithmus in konkreten Schritten beschreiben, legen wir zunächst noch eine Sprechweise fest:

Definition: Zwei Knoten heißen *benachbart*, wenn sie miteinander durch mindestens eine Kante verbunden sind.

Den Ablauf des Algorithmus werden wir an Hand eines Beispiels behandeln.

Beispiel:
Ein Paketzusteller muss ein Päckchen ausliefern. Unten in Abbildung 3 ist die Situation in einem gewichteten Graphen schematisch dargestellt . Der Zusteller geht von Knoten A los und muss die Lieferung zu Knoten G bringen. Die Knoten sind Kreuzungen und die Zahlen an den Kanten sind die Längen der Straßenabschnitte zwischen den Kreuzungen.

<u>Frage:</u> Um Benzin zu sparen, will er die kürzeste Route nehmen. Welche ist das und wie lang ist sein Weg dann?

<u>Lösung:</u>

1. Gib Knoten A das permanente Label 0.

2. Betrachte nun alle Nachbarn von Knoten A. In unserem Beispiel sind das B und C. Diese Nachbarn bekommen als vorläufiges Label das Gewicht der Kante zwischen A und dem jeweiligen Knoten. Wähle von diesen vorläufigen Labels das kleinste und mache es permanent. Sollten mehrere Labels den kleinsten Wert haben, dann wähle einen dieser Knoten aus und mache dann sein Label permanent. In unserem Beispiel ist das Label von B mit einem Wert von 3 das kleinste und wird also permanent gemacht (siehe Abbildung 4).

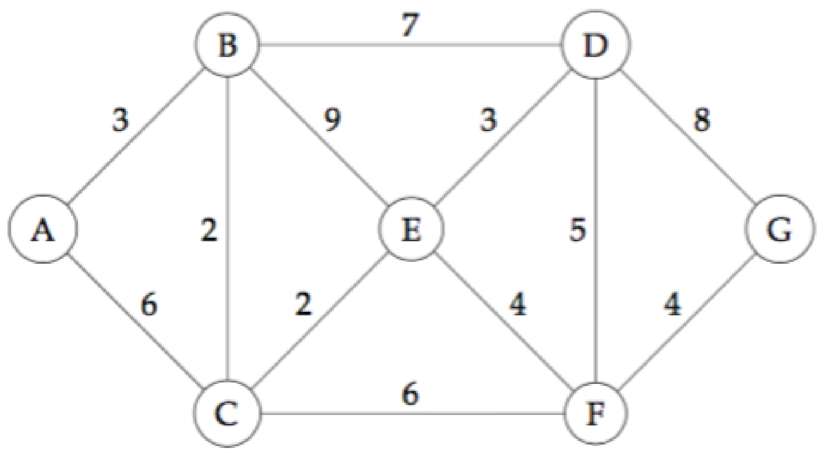


Abbildung 3: Schematische Wiedergabe Paketzusteller

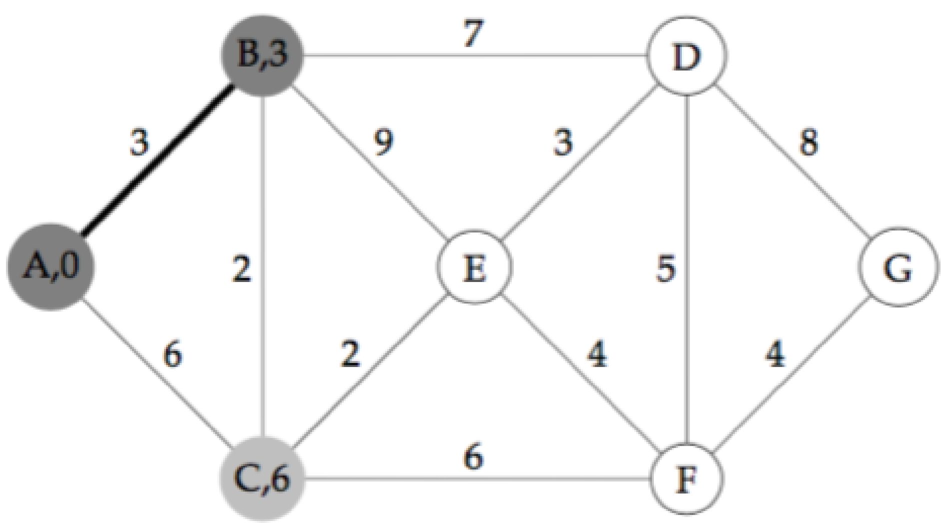


Abbildung 4: Abbildung zu Schritt 2 des Lösungswegs

3. Der Algorithmus setzt nun an jenem Knoten an, der im vorausgegangenen Schritt ein permanentes Label erhalten hat: hier ist das B. Betrachte nun die Nachbarn von B, die noch kein permanentes Label haben. In unserem Beispiel sind das C, D und E. Gib diesen Knoten nun vorläufige Labels. Um den Wert eines neuen Labels zu bestimmen, vergleichen wir zwei Zahlen:

   - Die erste Zahl ist der Wert eines vorläufigen Labels, das ihm schon zuvor zugewiesen wurde. (Wenn der Knoten noch kein vorläufiges Label erhalten hatte, so fällt diese erste Zahl einfach weg und wir schauen nur auf die zweite.)
   - Die zweite Zahl ist die Summe von dem Label des zuletzt permanent-gelabelten Knotens (in

unserm Fall B) und dem Gewicht der Kante zwischen diesem Knoten und dem betrachteten Nachbarn.

Das Minimum dieser beiden Zahlen wird als neues vorläufiges Label des Knotens gewählt. Für Knoten C muss das neue Label also das Minimum sein von 3 + 2 = 5 und 6. Demnach bekommt C das vorläufige Label 5. Wenn es Knoten gibt, die Nachbarn sind von A, aber nicht von B (in diesem Fall ist das nicht so), dann behalten sie ihr vorläufiges Label. Erkläre nun das kleinste vorläufige Label wieder als permanent In diesem Fall ist das C (siehe Abbildung 5).

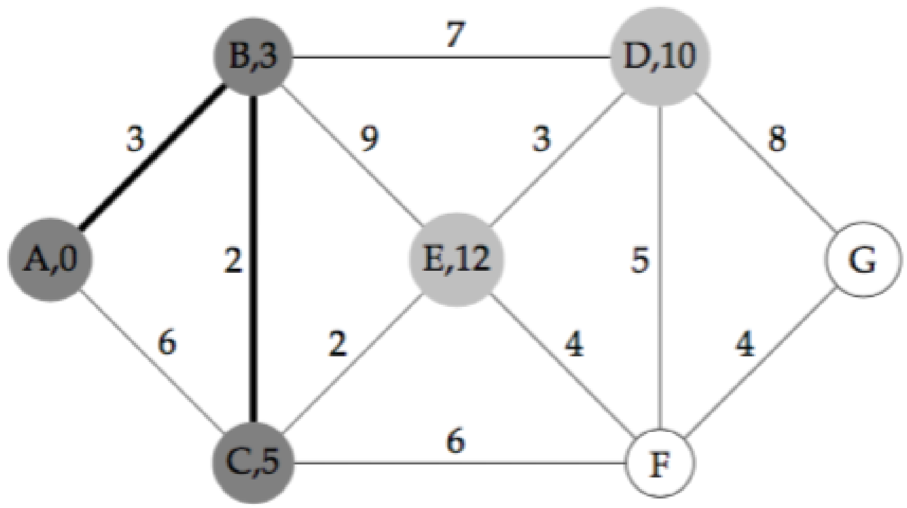


Abbildung 5: Abbildung zu Schritt 3 des Lösungswegs

4. Wiederhole Schritt 3: Gib neue vorläufige Label an die Nachbarn von C und mach das kleinste vorläufige Label im Graphen permanent (sieh Abbildung 6).

5. Wiederhole nun Schritte 3 so lange, bis alle Knoten, inklusive G, ein permanentes Label haben. Das Label von G ist dann die Gesamtlänge des kürzesten Pfades von A nach G. Die tatsächliche Route ist jetzt vielleicht nicht direkt deutlich. In Abbildungen 4, 5 und 6 ist jeweils die bis dahin kurzeste Route fett gedruckt. Allerdings kann sich diese Route nach dem Permanent-machen eines neuen Knotens wieder ändern. Darum kann es einfacher sein, den ganzen Algorithmus vollständig durchzuführen und in einer Tabelle zu dokumentieren (siehe Tabelle 5). Man muss natürlich sorgfältig im Auge halten, wie jeder Knoten an sein permanentes Labeel gekommen ist und welche Kanten nun in den kürzesten Teilweg eingeflossen sind.

Schritt	Vorläufige Labels	Permanent gemacht	Pfad
1		A,0	A
2	C,6	B,3	AB
3	E,12 D,10	C,5	ABC
4	F,11 D,10	E,7	ABCE

Tabelle 5: Tabelle zum Beispiel

Manchmal gibt es mehrere Lösungen. Wenn der Zusteller zum Beispiel die kürzeste Route von A

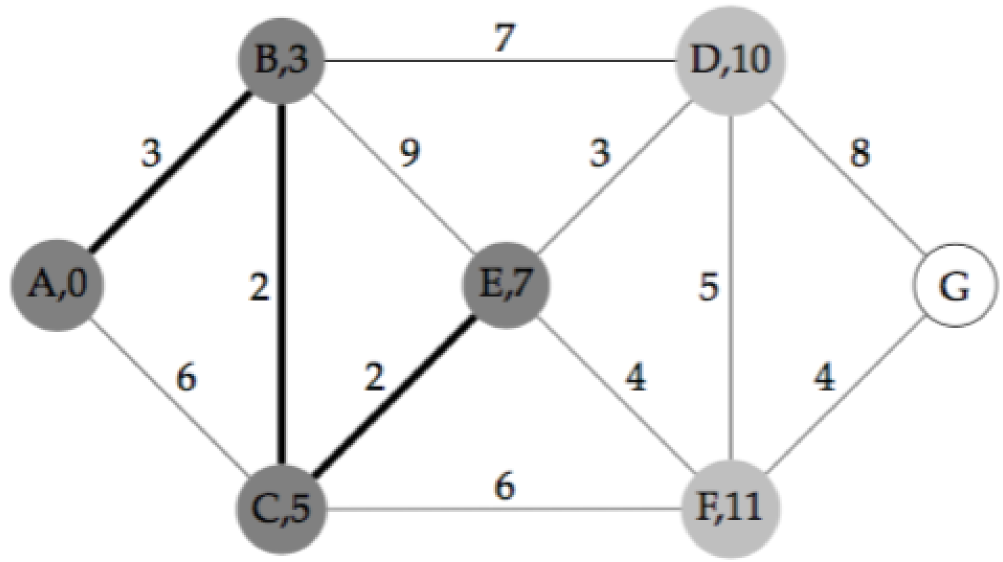


Abbildung 6: Abbildung zu Schritt 4 des Lösungswegs

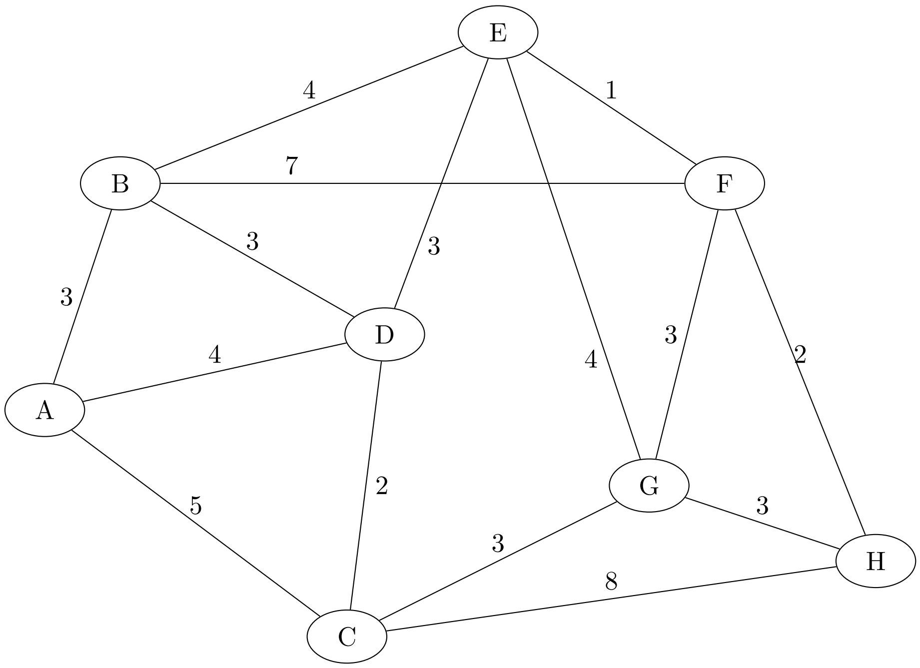
nach D wissen wollte, dann wären sowohl der Pfad A,B,D als auch der Pfad A,B,C,E,D mit jeweils einer Länge von 10 optimale Lösungen.

Bei Übungsaufgabe 1 kannst du selbst die letzten Schritte für dieses Beispiel ausführen und damit die kürzeste Route für den Paketzusteller finden.

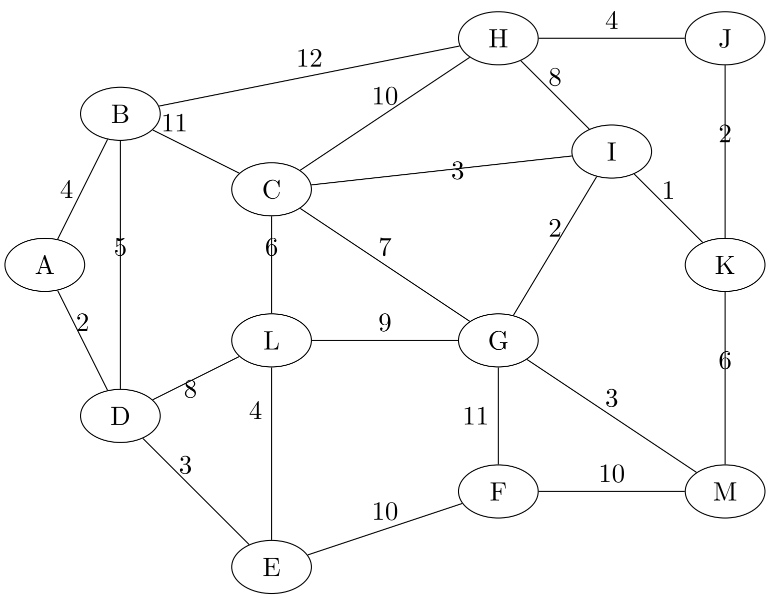
3.2.2 Übungsaufgaben

1. Rechne das Beispiel zu Ende. Finde den kürzesten Pfad und den zugehörigen Abstand von A nach G.

2. Finde für unten stehenden Graphen den kürzesten Pfad von A nach H und die zugehörige Länge.



3. Finde im unten stehendem Graphen den kürzesten Pfad von A nach M.



3.3 Maximaler Fluss, minimaler Schnitt

3.3.1 Theorie

In diesem Kapitel spezialisieren wir uns auf eine bestimmte Klasse von Graphen.

Definition: Ein *gerichteter* Graph ist ein Graph, bei dem die Kanten eine feste Orientierung haben. Kanten werden nun also durch geordnete Paare von Knoten beschrieben. Jede Kante hat nun also einen Anfangs- und einen Endpunkt. Die Richtung der Kante symbolisieren wir durch Pfeile.

Definition: Ein *(Transport-) Netzwerk* ist ein gerichteter Graph mit den folgenden Eigenschaften:

- Es gibt genau einen Knoten, zu dem keine Kanten führen. Dieser Knoten heißt *Quelle*.
- Es gibt genau einen Knoten, von dem keine Kanten wegführen. Dieser Knoten heißt *Senke*.
- Der Graph ist ein gewichteter Graph. Dabei stehen die Gewichte für die *Kapazitäten* der Kanten: die maximale Menge, die durch eine Kante fließen kann.

Ein Beispiel für ein Transportnetzwerk ist gegeben in Abbildung 7.

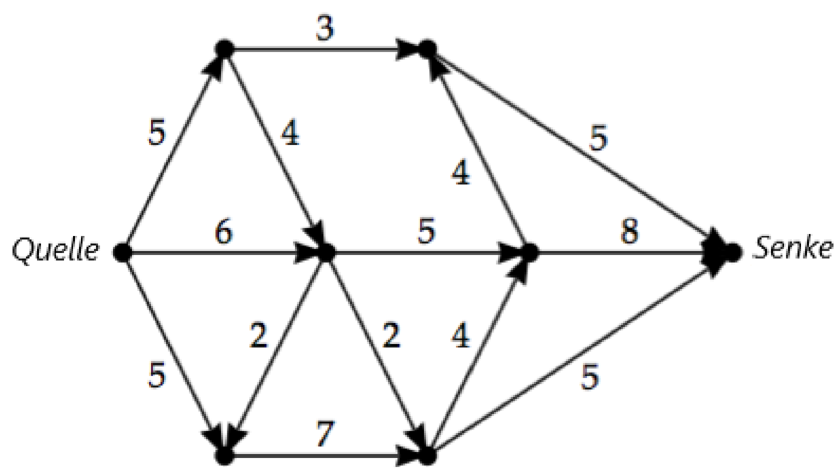


Abbildung 7: Schematische Wiedergabe eines Netzwerks. Für jede Kante ist die Kapazität gegeben.

Definition: Ein *Fluss* eines Netzwerks bestimmt die Menge, die tatsächlich durch ein Netzwerk fließt. Ein Fluss wird durch eine 'Umgewichtung' der Kanten eines Netzwerks beschrieben und ist damit formal selbst ein Netzwerk; dieses muss die folgenden Eigenschaften haben:

- Der Fluss in jeder Kante darf höchstens gleich der Kapazität der Kante sein.
- In jedem Knoten (außer der Quelle und Senke) muss die Menge, die hineinfließt (also die Summe der Gewichte, die an Kanten stehen, deren Endpunkt der betreffende Knoten ist), dieselbe sein wie die Menge, die aus dem Knoten herausfließt (also die Summe der Gewichte, die an Kanten stehen, deren Anfangspunkt der betreffende Knoten ist). Es darf sich unterwegs also kein Wasser aufstauen.
- In jeder Kante muss die Strömungsrichtung gleich der der Kante zugehörigen Orientierung sein. Kein Gewicht des Flusses ist also negativ.

Ein möglicher Fluss durch das Netzwerk in Abbildung 7 ist gegeben in Abbildung 8. Die insgesamt durch den Fluss transportierte Menge ist hier gleich $3 + 5 + 5 = 13$.

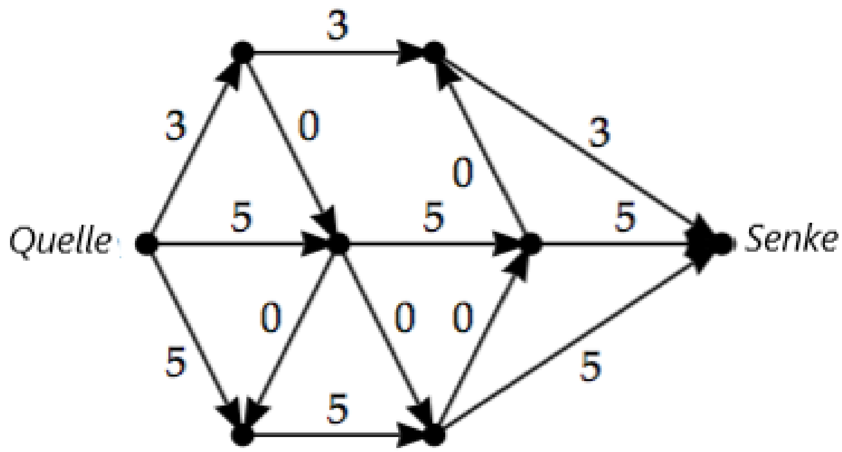


Abbildung 8: Schematische Wiedergabe eines möglichen Flusses durch das Netzwerk

Du kannst selbst kontrollieren, dass der Fluss alle nötigen Eigenschaften hat.

Auch jetzt richten wir unsere Aufmerksamkeit wieder auf die Optimierung solcher Flüsse. Beispielsweise für ein Wasserwerk ist es nämlich wichtig, die Zuleitungsrohre so effizient möglich zu nutzen. Anhand des Beispiels aus Abbildung 7 werden wir zeigen wie der maximale Fluss bestimmt werden kann.

Beispiel: Ein Wasserwerk will so viel Wasser wie möglich in eine Wohnsiedlung leiten. Nimm an, dass die Kapazität der Menge Wasser, das maximal pro Minute durch ein entsprechendes Rohr geleitet werden kann, den in Abbildung 7 gegebenen Werten entspricht.

Frage: Wie viel Wasser kann maximal pro Minute durch das Netzwerk fließen?

Lösung:

1. Nehmen wir den Fluss in Abbildung 8 als Ausgangspunkt. Dieser Fluss nutzt noch nicht die volle Kapazität der Kanäle (siehe Abbildung 9).

2. Verwende die noch ungenutzten Kapazitäten indem du einen *flussvermehrenden* Pfad von der Quelle zur Senke konstruierst (unter Berücksichtigung der Anforderungen an einen Fluss). Ein Beispiel für einen solchen flussvermehrenden Pfad ist gegeben in Abbildung 10. Diesen zusätzlichen Fluss kann man nun zu dem erst gefundenen Fluss addieren ohne die Kapazität zu überschreiten.

3. Suche auf diese Weise so lange flussvermehrende Pfade, bis es keine mehr gibt.

4. Wenn auf diese Weise keine flussvermehrende Pfade mehr zu finden sind, können wir möglicherweise noch immer den Fluss vergrößern. Hierzu betrachten wir ein weiteres Beispiel:
   Wir sehen in Abbildung 11 links ein einfaches Netzwerk. In der mittleren Abbildung ist ein mögicher Fluss eingezeichnet mit einem Wert von 5. Wenn wir die noch ungenutzten Kapazitäten betrachten (s. Abbildung ganz rechts), dann sehen wir, dass beide Kanten mit Kapazität 3 bereits voll ausgenutzt sind. Hier kann also nicht noch mehr durchströmen als es jetzt der Fall ist. Kurz gesagt: es kann kein flussvermehrender Pfad mehr gefunden werden. Dennoch kann man leicht einen Fluss mit Wert 6 finden (siehst du wie?).

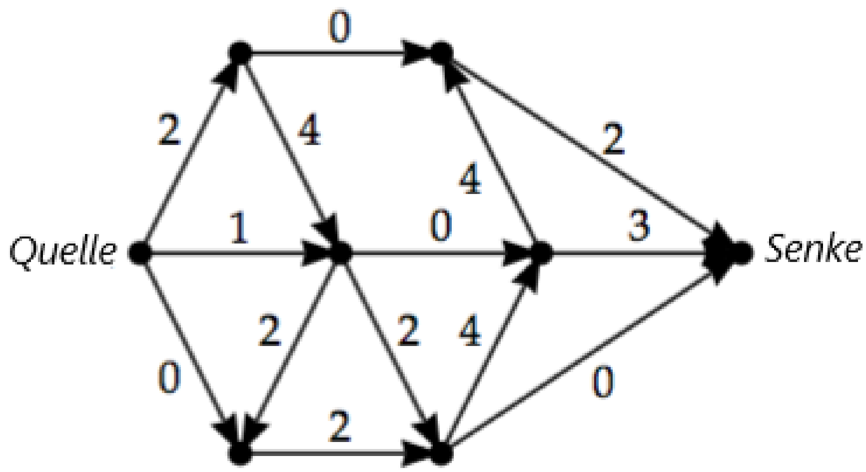


Abbildung 9: Schematische Wiedergabe der übergebliebenen Kapazitäten

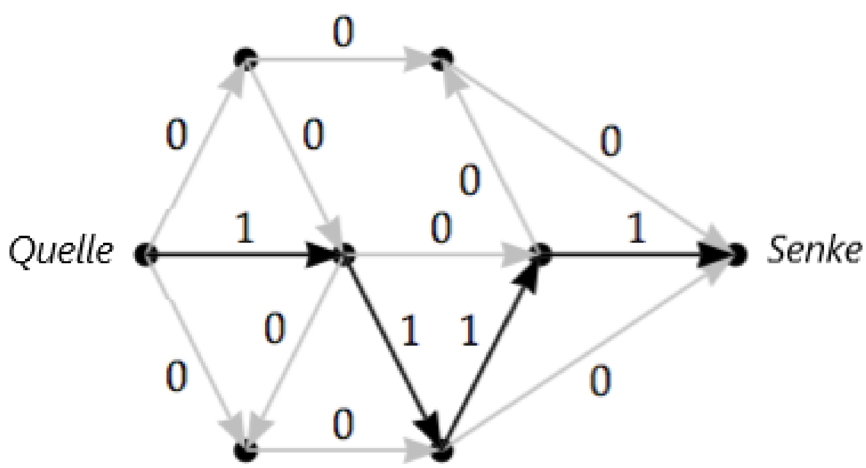


Abbildung 10: Beispiel eines flussvermehrenden Pfades

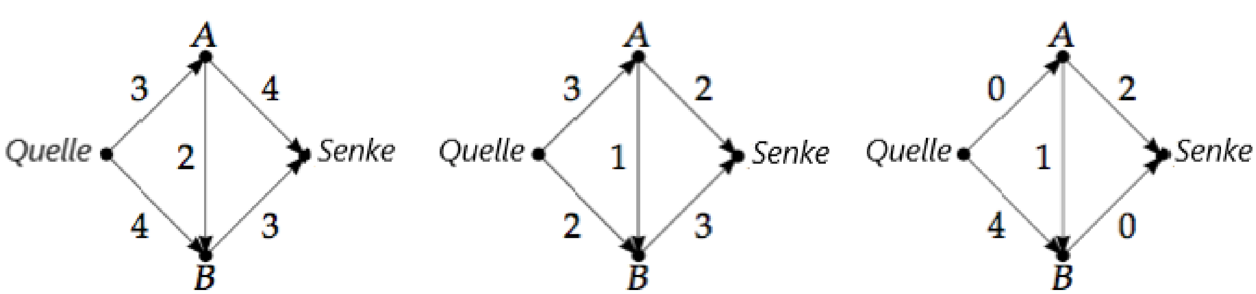


Abbildung 11: Links: die Kapazität. Mitte: möglicher Fluss. Rechts: übrige Kapazität

Der Grund ist, dass der Fluss mit Wert 1 durch die Kante (A,B) nicht clever gewählt ist. Hingegen bleibe die Kanten Quelle-B und A-Senke sehr unausgelastet- Um das zu ändern, könnten wir einen Fluss mit Wert 1 durch den Pfad Quelle-B-A-Senke senden wollen. Auf der Kante (B,A) verliefe aber entgegen der vorgegebenen Orientierung. Die Folge wäre, dass wir nun auf diesem Pfad einen Fluss von 1 mit der Orientierung und einen Fluss von 1 entgegen der Orientierung hätten. Beide würden einander aufheben. Bei den Kanten Quelle-A und B-Senke würde der erdachte Fluss hingegen der Orientierung der Kanten folgen, und darum können wir nun 1 addieren zu den Flussgewichten von Quelle-B und A-Senke, das Flussgewicht von A-B aber auf 0 reduzieren. Wir haben einen maximalen Fluss von 6 gefunden.

Mit anderen Worten: wir müssen auch flussvermehrende Pfade suchen, die teilweise entgegen der Orientierung fließen. Bei den mitlaufenden Pfeilen wird dann extra Fluss addiert, bei den entgegen gerichteten Pfeilen wird Fluss abgezogen. Natürlich dürfen für den maximalen Fluss keine negativen Flussgewichte auftreten.

In Übungsaufgabe 1 kannst du selbst diesen Schritt für das Beispiel mit dem Wasserwerk durchführen .

5. Du musst sicherstellen, dass du wirklich alle flussvermehrende Pfade gefunden hast. Hierbei können zwei Dinge helfen.

   Erstens: man kann keinen flussvermehrenden Pfad mehr finden, wenn jeder Pfad von der Quelle zur Senke entweder eine Kante in Richtung der Orientierung durchfließt, die schon ihrer Kapazität entsprechend gesättigt ist, oder eine Kante, in der der Strom schon 0 ist, in entgegengesetzter Richtung durchströmt.

   Für das zweite Hilfsmittel müssen wir uns erst noch mit einer wichtigen Definition und einem Satz auseinandersetzen.

   Definition: Ein *Schnitt* in einem Netzwerk ist eine Menge von Kanten, nach deren Entfernung kein gerichteter Pfad von der Quelle zur Senke mehr möglich ist. Die Größe eines Schnitts ist die Summe der Kapazitäten der Kanten des Schnitts.

   Wenn wir das Beispiel in Abbildung 11 betrachten, sehen wir, dass die Kanten (QUelle,A) und (B,Senke) einen möglichen Schnitt formen. Die Größe dieses Schnitts ist 6.

   Satz: In jedem Netzwerk gilt: *Der maximale Fluss ist gleich dem minimalen Schnitt.* (Der Satz wird min cut - max-flow - Satz genannt).

   Der Beweis ist etwas abstrakt, aber intuitiv ist der Zusammenhang wohl auch schon deutlich. Der minimale Schnitt schaut nämlich nach der schmalsten Engstelle für einen Fluss durch das Netzwerk, und die begrenzt genau den maximalen Fluss.

   Ein Nutzen des Satzes besteht darin, dass man kontrollieren kann, ob man den maximalen Fluss gefunden hat. Man schaut, ob man einen Schnitt derselben Größe finden kann. Wenn dein Fluss kleiner ist als der kleinstmögliche Schnitt, dann kann man noch einen flussvermehrenden Pfad finden.

   Auch nach dem minimalen Schnitt der Beispielaufgabe wird in Übungsaufgabe 1 gefragt.

Wir haben gesehen, dass die Orientierung der Kanten ziemlich praktisch ist beim Lösen solcher Optimalisierungsprobleme. Dennoch können wir auch bei ungerichteten Graphen die oben erläuterte Methode gebrauchen, wenn wir jede Kante durch zwei gerichtete Kanten (eine pro Orientierung) ersetzen . Beide bekommen die gleiche gegebene Kapazität. Im Beispiel unten siehst du wie das funktioniert. Beim Ermitteln des maximalen Stroms wirst du feststellen, dass immer nur eine der Richtungen verwendet wird (warum?).

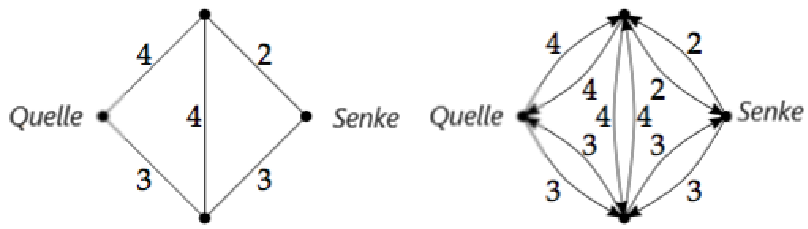


Abbildung 12: Links: ein ungerichteter Graph. Rechts: ein gerichteter Graph

3.3.2 Übungsaufgaben

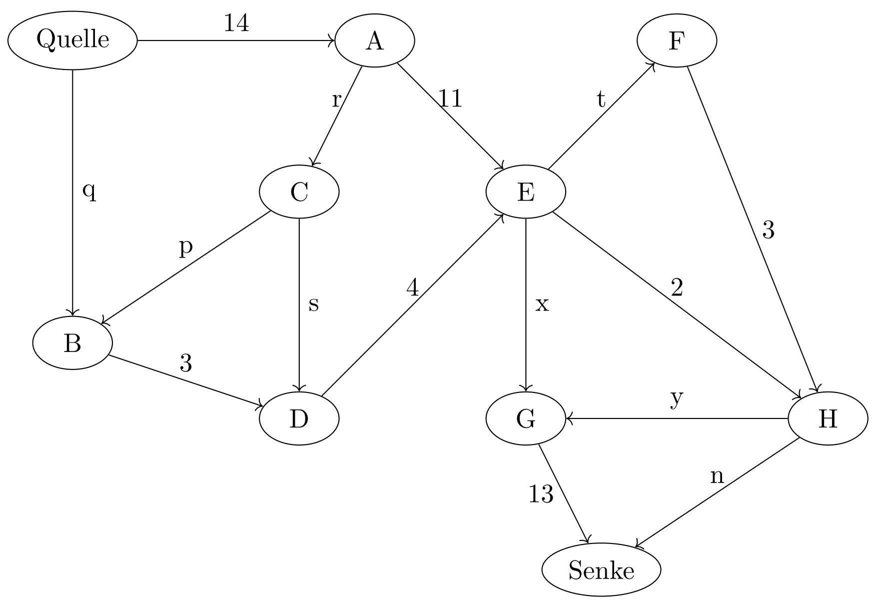
1. Stelle das Beispiel mit dem Wasserwerk fertig: finde den maximalen Fluss. Was ist die Größe dieses Flusses? Gib auch den minimalen Schnitt und die zugehörige Größe an.

2. Der untenstehende Graph zeigt einen Fluss durch ein Netzwerk, bei dem alle Kanten die Kapazität 15 besitzen.

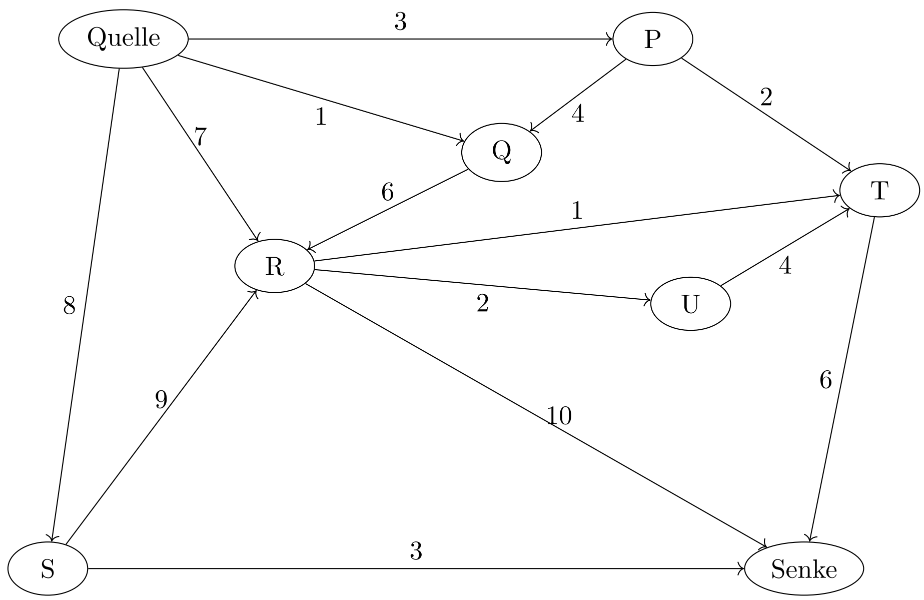
   (a) Die Zahlen, die an den Kanten stehen, geben an wie viel Wasser durch die Kante strömt. Bei einigen Kanten steht allerdings ein Buchstabe statt einer Zahl. Welche Werte müssen diese Buchstaben kriegen, damit ein richtiger Fluss durch das Netzwerk fließt?

   (b) Was ist die Größe dieses Flusses?

   (c) Finde mindestens 5 minimale Schnitte!



3. Finde den maximalen Fluss und den minimalen Schnitt für den untenstehenden Graph. Was ist die zugehörende Größe?



4. Finde den maximalen Fluss von der Quelle A zur Senke G und den minimalen Schnitt in untenstehendem Graphen. Gib beim maximalen Fluss auch die Orientierung an. Was ist die Größe dieses Flusses und des minimalen Schnitts?