Die Theorie der Graphen

Inspektor Manoris Geheimnisse

Alain Hertz

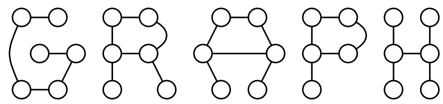
23. August 2011

Inhaltsverzeichnis

A Definitionen und Grundkonzepte

A.1 Wege, Kreise und Zusammenhang

Nehmen Sie ein Blatt Papier, wählen Sie einige Stellen, die Sie markieren, beispielsweise durch kleine Kreise, und ergänzen Sie einige Verbindungen zwischen bestimmten Paaren von Markierungen. Die von Ihnen gewählten Markierungen, diese kleinen Kreise auf Ihrem Blatt, werden als *Knoten* oder *Ecken* bezeichnet, während die Striche, die Sie gezogen haben, um einige davon zu verbinden, als *Kanten* bezeichnet werden. Diese Striche können gerade oder gebogen sein, wesentlich ist, ob zwischen zwei Knoten eine Verbindung existiert oder nicht. Folglich ist zum Beispiel die Zeichnung aus der nachfolgenden Abbildung ein Graph, der 29 Knoten und 27 Kanten enthält.



Auf diese Weise beschreibt Manori einen Graphen. Formaler ausgedrückt, ist ein Graph durch ein geordnetes Paar (V,E) definiert: V ist eine Menge von Elementen, die man als *Knoten* bezeichnet, während E eine Menge von Linien, sogenannten *Kanten*, ist, die bestimmte Knotenpaare verbinden. E ist daher eine Teilmenge des kartesischen Produkts $V \times V$. Im Allgemeinen schreibt man $\{x,y\}$ für eine Kante, welche die Knoten x und y verbindet.

Eine Kante ist folglich ein ungeordnetes Paar von Knoten. Für eine Kante, welche die Knoten x und y verbindet, kann man sowohl $\{x,y\}$ als auch $\{y,x\}$ schreiben.

Es sei erwähnt, dass eine Kante auch einen Knoten mit sich selbst verbinden kann. Dann spricht man von einer *Schleife*. Es können auch mehrere Kanten gleichzeitig existieren, die dasselbe Knotenpaar verbinden. Man spricht dann von einem *Multigraphen*.

Der Multigraph aus Abbildung A.1 auf der nächsten Seite enthält zum Beispiel vier Knoten, die als schwarze Kreise dargestellt sind und die Bezeichnungen a, b, c und d tragen. Es gibt eine Schleife um den Knoten d, und es gibt zwei parallele Kanten zwischen den Knoten a und c.

Definition A.1. Ein *Weg* in einem Graphen ist eine Folge von Kanten, durch die zwei Knoten des Graphen miteinander verbunden sind.

Wenn man zur Illustration im Graphen aus Abbildung A.1 die Kanten $\{b,d\}$ und $\{c,d\}$ nacheinander durchläuft, erhält man einen Weg, der die Knoten b und c verbindet.

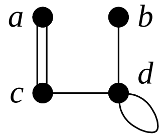


Abbildung A.1: Ein Multigraph mit den Knoten a, b, c und d.

Tatsächlich existieren noch andere Wege zwischen den Knoten b und c. Man könnte zum Beispiel die nachfolgenden Kantenzüge betrachten:

- $\{b,d\}$, $\{d,d\}$, $\{c,d\}$, $\{a,c\}$, $\{a,c\}$
- $\{b,d\}$, $\{d,d\}$, $\{d,d\}$, $\{d,d\}$, $\{d,d\}$, $\{c,d\}$.

Definition A.2. Ein Weg, der jeden Knoten nicht mehr als ein Mal besucht, heißt *elementar*.

Im Beispiel aus Abbildung A.1 ist der einzige elementare Weg, der die Knoten b und c verbindet, der Weg aus den Kanten $\{b,d\}$ und $\{c,d\}$.

Definition A.3. Ein Weg, der jede Kante nicht mehr als ein Mal durchläuft, heißt *einfach*.

Im Graphen aus Abbildung A.1 existieren nur vier einfache Wege, die b und c verbinden. Sie setzen sich aus nachfolgenden Kantenzügen zusammen:

- $\{b,d\}$, $\{c,d\}$
- $\{b,d\}$, $\{d,d\}$, $\{c,d\}$
- $\{b,d\}$, $\{c,d\}$, $\{a,c\}$, $\{a,c\}$
- $\{b,d\}$, $\{d,d\}$, $\{c,d\}$, $\{a,c\}$, $\{a,c\}$.

Anmerkungen. Die beiden letzten Wege aus dieser Aufzählung sind einfach, weil es zwischen den Knoten a und c zwei verschiedene Kanten gibt. Würde zwischen diesen beiden Knoten nur eine Kante existieren, wären die beiden letzten Wege nicht einfach, denn sie würden die Kante, die den Knoten a mit dem Knoten c verbindet, zwei Mal durchlaufen.

Alle elementaren Wege sind einfach, weil mehrere Durchläufe ein und derselben Kante zwangsläufig mehrere Besuche der Knoten dieser Kante zur Folge haben.

Definition A.4. Ein Graph heißt *zusammenhängend*, wenn zwischen allen Knotenpaaren des Graphen ein Weg existiert.

Manori definiert den Begriff *zusammenhängend* gegenüber Sébastien, indem er folgende Begriffe verwendet:

> Wenn Du Dich von jedem beliebigen Knoten des Graphen zu jedem anderen beliebigen Knoten des Graphen über Kanten des Graphen bewegen kannst, nennt man den Graphen zusammenhängend. Dieses Wort hat seinen Ursprung darin, dass jeder beliebige Punkt des Graphen mit jedem beliebigen anderen Punkt des Graphen über Kanten des Graphen zusammenhängt.

Definition A.5. Man sagt, dass zwei Knoten in einem Graphen G genau dann zu derselben *Zusammenhangskomponente* gehören, wenn in G ein Weg existiert, der sie verbindet.

So existieren in dem Graphen aus der ersten Abbildung dieses Kapitels sechs Zusammenhangskomponenten.

Man kann somit beweisen, dass ein Graph genau dann zusammenhängend ist, wenn er nicht mehr als eine Zusammenhangskomponente besitzt.

Definition A.6. Ein Weg, dessen Ausgangspunkt gleich dem Endpunkt ist, heißt *Kreis*. Wieder spricht man von einem *elementaren* Kreis, wenn darin die Knoten des Graphen nicht mehr als ein Mal besucht werden, und man spricht von einem *einfachen* Kreis, wenn keine Kante zwei Mal durchlaufen wird.

Anmerkung. Alle elementaren Kreise sind zwangsläufig einfach, da sich ja aus dem mehrfachen Durchlaufen ein und derselben Kante ein mehrfacher Besuch der Knoten dieser Kante ergibt.

Im Beispiel aus Abbildung A.2 existiert nur ein elementarer Kreis, der jeden Knoten des Graphen besucht. Dabei handelt es sich um den Kreis aus den Kanten $\{a,b\}$, $\{b,c\}$, $\{c,d\}$, $\{d,e\}$, $\{e,f\}$ und $\{f,a\}$. Dieser Kreis ist gleichzeitig einfach.

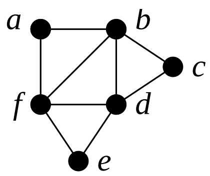
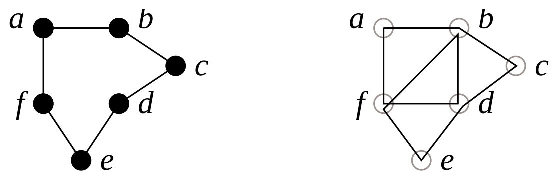


Abbildung A.2: Ein Graph mit einem elementaren Kreis.

Derselbe Graph enthält noch andere einfache Kreise, zum Beispiel den Kreis aus den Kanten $\{a,b\}$, $\{b,c\}$, $\{c,d\}$, $\{d,e\}$, $\{e,f\}$, $\{f,b\}$, $\{b,d\}$, $\{d,f\}$ und $\{f,a\}$, wie Abbildung A.3 auf der nächsten Seite illustriert.



elementarer Kreis (und daher auch einfach) — einfacher Kreis (aber nicht elementar)

Abbildung A.3: Graphen mit einfachen Kreisen.

A.2 Gerichtete Graphen

In bestimmten Fällen kann es interessant sein, die Kanten eines Graphen mit einer Orientierung oder Richtung zu versehen. So kann man anstelle einer Kante $\{x,y\}$ zwischen den Knoten x und y eine Kante haben, die von x nach y gerichtet ist, oder eine Kante, die von y nach x gerichtet ist. Man spricht dann von *gerichteten* Kanten und verwendet die Notation (x,y), um eine gerichtete Kante von x nach y zu kennzeichnen.

Manchmal hat man es auch mit *gemischten* Graphen zu tun, in denen einige (aber nicht alle) Kanten eine Orientierung oder Richtung haben.

In Kapitel 4 hat Bonvin einen gemischten Graphen gezeichnet, als er versuchte, die Ortskonfiguration des Geländespiels zu rekonstruieren. Die gerichteten Kanten veranschaulichen den kürzesten Weg, der es ermöglicht, die von Monsieur Grumbacker hinterlassenen Hinweise zu sammeln.

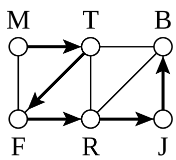


Abbildung A.4: Ein gemischter Graph.

Manori hat seinerseits gerichtete Graphen in Kapitel 8 verwendet, um zu versuchen, Familie Courtel am Dienstagmorgen länger schlafen zu lassen (siehe Abbildung A.5 auf der nächsten Seite).

Alle oben definierten Begriffe für ungerichtete Graphen können auf gerichtete Graphen übertragen werden. Man erhält so zum Beispiel die nachfolgenden Definitionen.

Definition A.7. Ein *gerichteter Weg* in einem gerichteten Graphen ist eine Folge von gerichteten Kanten über die man von einem Knoten zu einem anderen gelangen kann.

Beispiel A.1. In Kapitel 8 hat Manori einen gerichteten Weg gezeichnet, der den Knoten $A1$ mit dem Knoten $V6$ verbindet.

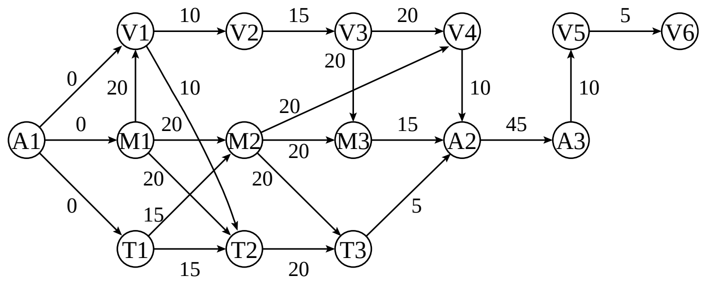


Abbildung A.5: Gerichteter Graph aus Kapitel 8.

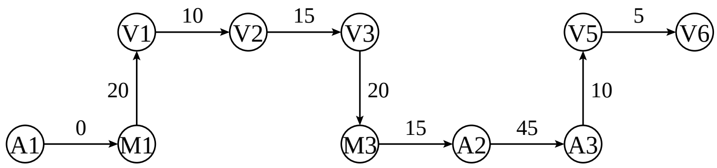


Abbildung A.6: Ein gerichteter Weg im Graphen aus Abbildung A.5.

Definition A.8. Ein gerichteter Graph ist *stark zusammenhängend*, wenn für alle Knotenpaare x, y des Graphen ein gerichteter Weg von x nach y und von y nach x existiert.

Beispiel A.2. Der gerichtete Graph, den Manori gezeichnet hat, um die Courtels länger schlafen zu lassen (siehe Abbildung A.5), ist nicht stark zusammenhängend, weil beispielweise kein gerichteter Weg von $T1$ nach $A1$ existiert. Es reicht allerdings aus, eine gerichtete Kante von $V6$ nach $A1$ einzufügen, um den Graphen stark zusammenhängend zu machen.

Definition A.9. Ein *gerichteter Kreis* in einem gerichteten Graphen ist ein Weg, in dem der Ausgangspunkt mit dem Endpunkt zusammenfällt.

Beispiel A.3. Im Graphen aus Abbildung A.7 existiert ein gerichteter Kreis, der die Knoten a, b und c verbindet.

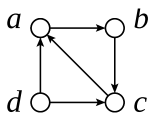


Abbildung A.7: Graph mit einem gerichteten Kreis.

Die gerichteten Wege und Kreise können elementar oder einfach sein wie die Wege und Kreise in ungerichteten Graphen. Sie sind elementar, wenn sie ein und denselben Knoten nicht zwei Mal besuchen, und sie sind einfach, wenn sie ein und dieselbe gerichtete Kante nicht zwei Mal durchlaufen.

A.3 Knotengrade

Im ersten Kapitel erklärt Manori gegenüber Sébastien:

> Wenn du einen beliebigen Graphen zeichnest, kannst du immer die Anzahl der Kanten angeben, die an jedem Knoten liegen.

Definition A.10. Die Anzahl der Kanten, die an einem Knoten liegen, ist sein *Grad*.

Im Graphen aus Abbildung A.2 auf Seite 3 haben die Knoten a, c und e daher zum Beispiel den Grad 2, während die Knoten b, d und f den Grad 4 haben.

Die Eigenschaft, anhand derer Manori aufzeigen konnte, dass es den Organisatoren nicht gelingen wird, die gewünschten Gruppen so zusammenzusetzen, dass alle Regeln befolgt werden, ist folgende.

Satz A.1. Die Summe der Knotengrade in einem Graphen ist immer eine gerade Zahl.

Beweis. Wenn man die Summe über alle Knotengrade bildet, zählt man jede Kante zwei Mal. Jede Kante $\{x,y\}$ wird nämlich ein erstes Mal berücksichtigt, wenn man den Grad des Knotens x betrachtet, und ein zweites Mal, wenn man den Grad des Knotens y addiert. Diese Summe entspricht folglich dem Doppelten der Anzahl der Kanten in dem Graphen, und es handelt sich folglich um eine gerade Zahl. □

Im oben genannten Beispiel haben wir 3 Knoten vom Grad 2 und 3 Knoten vom Grad 4, was insgesamt $3 \cdot 2 + 3 \cdot 4 = 18$ ergibt, also das Doppelte von 9 – die Anzahl der Kanten in diesem Graphen.

Genau damit ist es also Manori gelungen, Sébastien davon zu überzeugen, dass die von den Organisatoren gewünschte Zusammensetzung nicht existieren kann. Tatsächlich enthält der Graph, den er betrachtet, einen Knoten pro COPS-Teilnehmer und eine Kante zwischen zwei Teilnehmern, wenn sie bereits zusammengearbeitet haben. Zu fordern, dass jeder Teilnehmer in einer 15er Gruppe mit 7 bekannten Gesichtern sein sollte, ist äquivalent zu dem Vorhaben, einen Graphen mit 15 Knoten zu konstruieren, die alle den Grad 7 haben. Da die Summe der Knotengrade dieses gewünschten Graphen $7 \cdot 15 = 105$ sein müsste, was eine ungerade Zahl ist, kann der Graph, den die Organisatoren konstruieren wollen, nicht existieren.

Andere Anwendungen

Genauso ist es unmöglich, 15 Computer so mit Kabeln zu verbinden, dass jeder Computer genau zu 7 anderen Computern eine direkte Verbindung hat. In diesem Fall sind die Computer die Knoten, und die Kabelverbindungen zwischen ihnen sind die Kanten.

Genauso ist es unmöglich, 15 Geradensegmente in der Ebene so zu legen, dass jedes Segment genau 7 andere Segmente schneidet. Die Knoten entsprechen hier den

Segmenten und die Kanten den Schnittpunkten der Segmente. Möchte man dagegen 6 Segmente haben, von denen jedes jeweils genau 3 andere Segmente schneidet, ist das möglich, beispielsweise mithilfe der 6 Segmente aus Abbildung A.8.

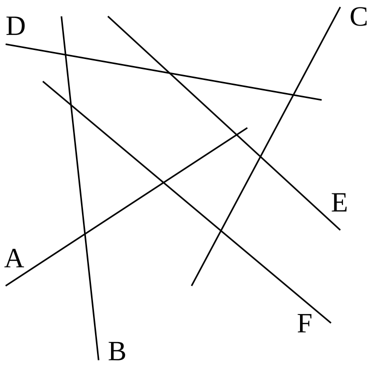


Abbildung A.8: 6 Segmente, die jeweils genau 3 andere Segmente schneiden.

Diesen Segmenten kann man den Graphen aus Abbildung A.9 zuordnen.

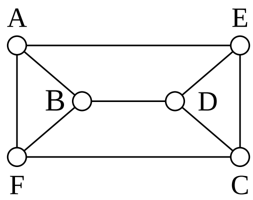


Abbildung A.9: Der Graph zu Abbildung A.8.

Es lassen sich andere, sehr einfache Eigenschaften der Knotengrade beweisen. Hier ist ein anderes Beispiel.

Satz A.2. In jedem Graphen existieren mindestens zwei Knoten mit demselben Grad.

Beweis. Betrachten wir einen beliebigen Graphen, und sei x die Anzahl seiner Knoten. Der Grad jedes Knotens ist daher eine Zahl aus der Menge $\{0, 1, \ldots, x-1\}$, weil jeder Knoten nicht mit mehr als $x-1$ anderen Knoten verbunden sein kann. Wenn alle Grade voneinander verschieden sind, muss zwangsläufig zu jedem Element y der Menge $\{0, 1, \ldots, x-1\}$ ein Knoten mit dem Grad y existieren, da $\{0, 1, \ldots, x-1\}$ x Elemente hat und der Graph x Knoten beinhaltet.

Es existiert folglich ein Knoten vom Grad 0, der mit keinem der anderen Knoten verbunden ist.

Es existiert auch ein Knoten vom Grad $x-1$, der mit allen anderen Knoten verbunden ist. Das ist aber ein Widerspruch, weil der Knoten vom Grad $x-1$ nicht mit dem Knoten vom Grad 0 (der mit keinem anderen Knoten verbunden ist) verbunden sein kann. □

Anwendung

Stellen wir uns vor, dass sich die COPS-Organisatoren entschließen, die Regeln für die Zusammensetzung der Arbeitsgruppen zu ändern. Sie könnten sich zum Beispiel entschließen, es so einzurichten, dass zwei Teilnehmer in derselben Gruppe sind, wenn sie mit unterschiedlich vielen Teilnehmern aus dieser Gruppe bekannt sind. Manori wäre nun auch hier in der Lage zu beweisen, dass eine solche Zusammensetzung unmöglich ist, weil der zu jeder Gruppe gehörende Graph Knoten mit Graden enthalten müsste, die sich alle voneinander unterscheiden, was aufgrund von Satz A.2 auf der vorherigen Seite ausgeschlossen ist.

B Planare Graphen

In Kapitel 2 ›Die Villen des Bellevue‹ des Buches definiert Manori gegenüber Sébastien planare Graphen wie folgt.

Definition B.1. Ein Graph ist *planar*, wenn man ihn auf einem Blatt Papier so zeichnen kann, dass sich seine Kanten nicht überschneiden. Eine solche Darstellung heißt *topologisch planarer Graph*.

Zum Beispiel ist der linke Graph aus Abbildung B.1 planar, weil man durch Verformung einer Kante den rechten Graphen erhalten kann, der im Grunde genommen nur eine andere Darstellung ein und desselben Graphen ist, in der sich jedoch die Kanten nicht überschneiden.

Nach der oben genannten Definition sind diese beiden Graphen planar, aber nur der rechte Graph ist topologisch planar.

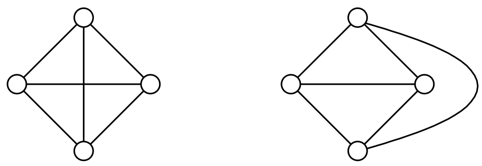


Abbildung B.1: Darstellung eines Graphen mit und ohne Kantenüberschneidung.

Definition B.2. In einem topologisch planaren Graphen nennt man die von den Kanten umschlossenen Gebiete *Flächen*.

In dem Graphen, der Courtels Villa abbildet, gibt es beispielsweise 6 Flächen, die mit $f1$ bis $f6$ bezeichnet sind (siehe Abbildung B.2 auf der nächsten Seite). Es sei betont, dass das Außengebiet eines topologisch planaren Graphen ebenfalls eine Fläche ist. In unserem Beispiel ist das die Fläche $f6$.

Sei N die Anzahl der Knoten eines topologisch planaren Graphen, K sei die Anzahl seiner Kanten und F die Anzahl seiner Flächen. Euler hat den folgenden Satz bewiesen.

Satz B.1 (Eulerscher Polyedersatz). In allen topologisch planaren zusammenhängenden Graphen gilt

$$F = K - N + 2\,.$$

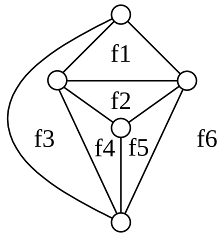


Abbildung B.2: Graph mit den inneren und äußeren Flächen $f1$ bis $f6$.

Mithilfe dieses Satzes lassen sich zahlreiche Resultate beweisen. Bezeichnet man mit P die minimale Anzahl von Kanten, die eine Fläche begrenzen, so erhält man zum Beispiel folgendes Resultat.

Satz B.2. Jeder topologisch planare zusammenhängende Graph mit N Knoten, deren kleinste Fläche[1] von P Kanten begrenzt ist, enthält höchstens $\frac{P}{P-2}(N-2)$ Kanten.

Beweis. Sei G ein topologisch planarer zusammenhängender Graph mit N Knoten, K Kanten, F Flächen und wie vorhin mit der Eigenschaft, dass der Rand jeder Fläche mindestens P Kanten enthält. Nach dem Satz von Euler gilt $F = K - N + 2$.

Wenn man die Summe über die Anzahl der Kanten bildet, die jede Fläche begrenzen, erhält man das Doppelte der Anzahl der Kanten. Jede Kante wird nämlich zwei Mal gezählt, weil sie im Rand von genau zwei Flächen auftaucht.

Da der Rand jeder Fläche mindestens P Kanten enthält, ergibt sich $2K \geq PF$. Mit dem Satz von Euler lässt sich die letzte Ungleichung wie folgt schreiben:

$$2K \geq P(K - N + 2)\,,$$

wenn man alle K nach rechts bringt, heißt das also:

$$\frac{P}{P-2}(N-2) \geq K\,.$$

□

Anhand dieser Eigenschaft, die Manori verwendet hat, um Courtel der Übertreibung zu überführen, kann man zeigen, dass ein planarer zusammenhängender Graph mit N Knoten nicht mehr als $3N - 6$ Kanten enthalten kann.

Satz B.3. Jeder planare zusammenhängende Graph mit N Knoten enthält höchstens $3(N-2)$ Kanten.

[1] Im Sinne von minimaler Anzahl von Kanten.

Beweis. Sei G ein planarer zusammenhängender Graph mit N Knoten und K Kanten. Wir betrachten eine Darstellung D von G, in der sich die Kanten nicht überschneiden. Die Darstellung D ist daher topologisch planar und hat dieselbe Anzahl von Knoten und Kanten wie der Graph G. Sei P die minimale Anzahl von Kanten auf dem Rand einer Fläche von D.

Da der Rand jeder Fläche aus mindestens 3 Kanten besteht, gilt $P \geq 3$, und mit Satz B.2 auf der vorherigen Seite erhalten wir $3(N-2) \geq \dfrac{P}{P-2}(N-2) \geq K$. □

Also kann der nachfolgende Graph, der Courtels Beschreibung seiner Villa entspricht, nicht planar sein, weil $3(N-2)$ gleich 9, K aber gleich 10 ist.

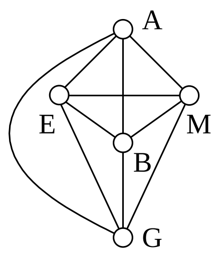


Abbildung B.3: Graph mit allen Verbindungen durch Türen oder Fenster.

Nehmen wir nun an, dass keine Fläche eines planaren Graphen ein Dreieck ist. Anders ausgedrückt, nehmen wir an, dass wenn man drei beliebige Knoten im Graphen wählt, immer mindestens eine Kante zwischen zwei dieser drei Knoten fehlt. Nun gilt $P \geq 4$, und die Schranke für die Anzahl der Kanten verändert sich, sie wird allerdings noch kleiner.

Satz B.4. Jeder planare zusammenhängende Graph mit N Knoten, der kein Dreieck enthält, hat höchstens $2(N-2)$ Kanten.

Beweis. Der Beweis ist analog zum letzten Beweis. Ist ein planarer zusammenhängender Graph G mit N Knoten, K Kanten und ohne Dreiecke gegeben, so kann man eine topologisch planare Darstellung D von G betrachten, in der sich die Kanten nicht überschneiden. Der Graph D hat dieselbe Anzahl von Knoten und Kanten wie der Graph G und enthält kein Dreieck. Die kleinste Anzahl P von Kanten auf dem Rand einer Fläche von G ist daher mindestens 4.

Satz B.2 auf der vorherigen Seite gilt für alle P, woraus wir die Ungleichung $2(N-2) \geq \dfrac{P}{P-2}(N-2) \geq K$ schließen. □

Mithilfe dieser Eigenschaft gelang es Manori zu zeigen, dass der Graph aus Abbildung B.4 auf der nächsten Seite, der den Anschlüssen der Villen an Wasser, Gas und

Strom entspricht, nicht planar sein kann. Dieser Graph enthält nämlich kein Dreieck, und es ist $2(N-2) = 8$, während $K = 9$ ist.

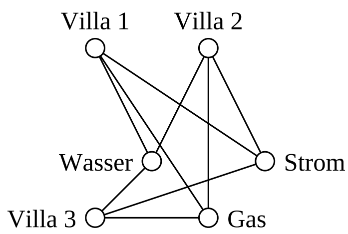


Abbildung B.4: Darstellung mit den Villen als Knoten.

Wie man sieht, ist es mitunter einfach zu zeigen, dass ein Graph nicht planar ist. Das ist allerdings nicht immer der Fall. Im nachfolgenden Beispiel gibt es $N = 6$ Knoten, $K = 11$ Kanten, und der Graph enthält Dreiecke. Wir wissen daher: Wenn dieser Graph planar ist, so muss die Ungleichung $3(N-2) \geq K$ gelten. In unserem Fall ist $3(N-2) = 12 \geq K$. Dieser Graph könnte also theoretisch planar sein. Er ist es aber nicht, denn es ist unmöglich, ihn zu zeichnen, will man alle Kantenüberschneidungen vermeiden.

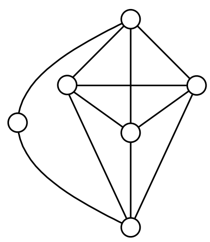


Abbildung B.5: Graph mit 6 Knoten und 11 Kanten, der nicht planar ist.

Mithilfe von Satz B.2 auf Seite 10 lassen sich zahlreiche andere Resultate beweisen. Man kann zum Beispiel beweisen, dass jeder planare Graph mindestens einen Knoten enthält, dessen Grad höchstens 5 ist.

Satz B.5. Jeder ebene Graph enthält mindestens einen Knoten, dessen Grad höchstens 5 ist.

Beweis. Sei G ein planarer Graph. Sei G' eine Zusammenhangskomponente von G. (Bedenken Sie, dass $G' = G$ ist, wenn G zusammenhängend ist.) Bezeichnet man die Knotenzahlen und Kantenzahlen von G' mit N und K, so erhalten wir wegen Satz B.3 auf Seite 10 eine Ungleichung, die sich wie folgt umschreiben lässt:

$$3N \geq K + 6\,. \tag{*}$$

Wir wollen den Beweis durch Widerspruch führen. Wir nehmen daher an, dass alle Knoten von G (und folglich auch von G') einen Grad von mindestens 6 haben. Wir wollen zeigen, dass dies zwangsläufig auf einen Widerspruch führt. Wir nehmen also an, dass die Summe der Knotengrade von G' mindestens $6N$ ist.

Wir haben bereits festgestellt, dass die Summe der Knotengrade eines Graphen gleich dem Doppelten der Kantenzahl ist. Es gilt daher $2K \geq 6N$, das heißt,

$$K \geq 3N\,. \tag{**}$$

Wenn wir die beiden Ungleichungen (*) und (**) verknüpfen, so erhalten wir $K \geq K+6$, was offensichtlich unmöglich ist. □

Betrachten wir erneut den Graphen aus Kapitel 1, in dem es einen Knoten pro Teilnehmer der COPS gibt und zwei Teilnehmer durch eine Kante verbunden sind, wenn sie bereits zusammengearbeitet haben. Wenn die Organisatoren der COPS ihre Arbeitsgruppen schließlich so festlegen, dass jeder Teilnehmer in der Vergangenheit mit genau 6 anderen Mitgliedern der Gruppe zusammengearbeitet hat, dann wissen wir, dass der zu einer Gruppe gehörende Graph nicht planar ist, weil der Grad aller Knoten 6 ist.

Populär wurden die Graphen durch den Schotten Francis Guthrie, der 1852 die Frage aufwarf, ob es möglich sei, alle geographischen Karten mithilfe von vier Farben eindeutig einzufärben, ohne dass zwei direkt aneinander grenzende Länder dieselbe Farbe haben. Stellt man jedes Land als einen Knoten dar und verbindet man zwei Knoten genau dann durch eine Kante, wenn die beiden Länder der Knoten direkt aneinander grenzen, so erhält man einen planaren Graphen. Francis Guthrie wollte also wissen, ob es immer möglich ist, die Knoten eines planaren Graphen mithilfe von vier Farben eindeutig einzufärben, unter der einzigen Bedingung, dass die Enden jeder Kante verschiedene Farben haben sollen. (Das ist äquivalent zu der Forderung, dass die Länder mit einer gemeinsamen Grenze verschiedene Farben haben).

Zur Illustration folgt hier eine Vier-Färbung einer imaginären geographischen Karte. Der zugehörige Graph ist gefärbt wie in Abbildung B.7 auf der nächsten Seite.

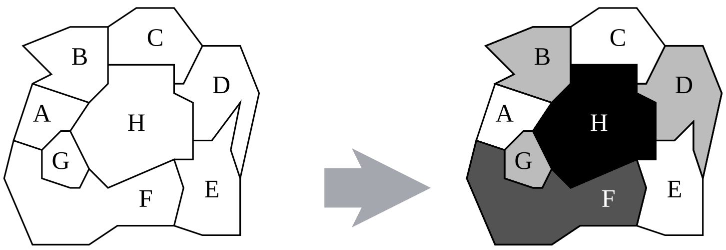


Abbildung B.6: Eine Vier-Färbung einer imaginären geographischen Karte.

Es ist leicht, jeden beliebigen planaren Graphen G zu färben, indem man maximal 6 Farben verwendet (anstelle von 4), unter der einzigen Bedingung, dass die Enden

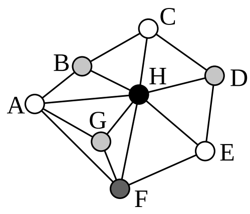


Abbildung B.7: Der zu einer Vier-Färbung gehörende Graph.

jeder Kante verschiedene Farben haben sollen. Dazu kann man zunächst die Knoten nach folgendem Verfahren ordnen (wobei N die Anzahl der Knoten des Graphen G ist).

> Betrachte den Graphen $G' = G$ und initialisiere einen Zähler i mit 0.
>
> Solange G' nicht leer ist, gehe wie folgt vor.
>
> Wähle einen Knoten mit dem kleinsten Grad in G' und bringe ihn an die Stelle $N-i$.
>
> Entferne diesen Knoten aus G' und erhöhe den Zähler i um eins.

Zur Illustration könnte man in dem oben betrachteten Graphen im ersten Schritt B, C, E oder G wählen, weil sie alle den minimalen Grad 3 haben. Wir wählen den Knoten B, der folglich an 8. Stelle ist.

In dem neuen Graphen, der sich ergibt, wenn wir den Knoten B entfernen, hat nur C den minimalen Grad 2. Wir bringen ihn daher an die 7. Stelle. Dann kann sofort D an die 6. Stelle gebracht werden, und anschließend E an die 5. Stelle. Es bleiben 4 Knoten vom Grad 3. Wir können uns zum Beispiel entscheiden, den Knoten H an die 4. Stelle zu bringen, dann F an die 3. Stelle, G an die 2. Stelle und schließlich A an die 1. Stelle. Wir erhalten somit eine Liste, die A, G, F, H, E, D, C und B in dieser Reihenfolge enthält. Dieses Verfahren illustriert Abbildung B.8 auf der nächsten Seite.

Wir können nun die Knoten des Graphen in der durch die Liste gegebenen Reihenfolge färben. Wenn wir einen Knoten färben, dann ist es per Konstruktion jedes Mal derjenige mit dem minimalen Grad im Graphen aus den Knoten, die bereits gefärbt wurden. Unter der Voraussetzung, dass G planar ist, wissen wir aus Satz B.5 auf Seite 12, dass dieser Grad zwangsläufig kleiner als 6 ist. Mit anderen Worten: Wenn wir einen Knoten färben, ist dieser mit höchstens 5 Knoten verbunden, die bereits gefärbt wurden. Es ist folglich immer eine der 6 Farben verfügbar.

Entscheiden wir uns zum Beispiel, nach Möglichkeit die Farbe Weiß zu verwenden, das helle Grau als zweite Wahl, das dunkle Grau als dritte Wahl und das Schwarz als vierte Wahl, so erhalten wir, wenn wir die Knoten in der Reihenfolge A, G, F, H, E, D, C und B färben, folgendes Muster. Zunächst weisen wir dem Knoten A das Weiß zu; dann dem Knoten G das helle Grau, dem Knoten F das dunkle Grau und dem Knoten

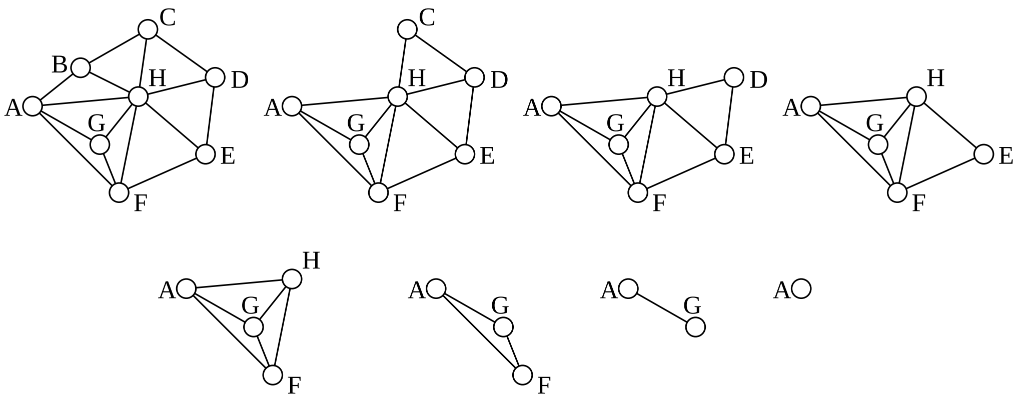


Abbildung B.8: Verfahren zur Ordnung der Knoten.

H das Schwarz. Wenn wir beim Knoten *E* angekommen sind, können wir wieder das Weiß oder das helle Grau verwenden, und wir weisen dem Knoten *E* daher das Weiß zu. Wir fahren auf diese Art und Weise fort, bis wir die Färbung erhalten, die wir schon weiter vorn angegeben haben.

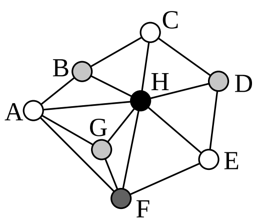


Abbildung B.9: Der zu einer Vier-Färbung gehörende Graph.

Ein bisschen (aber nicht viel) schwieriger ist es, einen planaren Graphen mit höchstens 5 Farben eindeutig zu färben. Aber erst seit 1976 weiß man dank Kenneth Appel und Wolfgang Haken, dass stets vier Farben ausreichen. Das zu zeigen, erweist sich aber als ziemlich schwierig.

C Intervallgraphen

Betrachten wir eine Menge von Aufgaben, die jeweils einen genauen Anfangs- und Endzeitpunkt haben. Angenommen, man möchte Angestellte mit der Ausführung dieser Aufgaben betrauen, aber kein Angestellter kann zwei Aufgaben auf einmal bewältigen. Um diese Situation zu modellieren, genügt es einen Graphen zu betrachten, in dem jede Aufgabe durch einen Knoten dargestellt wird und zwei Knoten durch eine Kante verbunden sind, wenn sich die entsprechenden Aufgaben zeitlich überschneiden. Der so konstruierte Graph ist ein *Intervallgraph*.

Definition C.1. Ein *Intervallgraph* ist ein Graph von Intervallüberschneidungen auf einer Geraden. Ist also eine Menge $I = \{I_1, \ldots, I_n\}$ von Intervallen auf einer Geraden gegeben, so ordnet man ihr den Intervallgraphen $G = (N,K)$ mit $N = \{1, \ldots, n\}$ zu, und zwei Knoten x und y sind genau dann durch eine Kante verbunden, wenn $I_x \cap I_y \neq \emptyset$ ist.

Beispiel C.1. Abbildung C.1 zeigt ein Beispiel für eine Menge von Aufgaben und den entsprechenden Intervallgraphen.

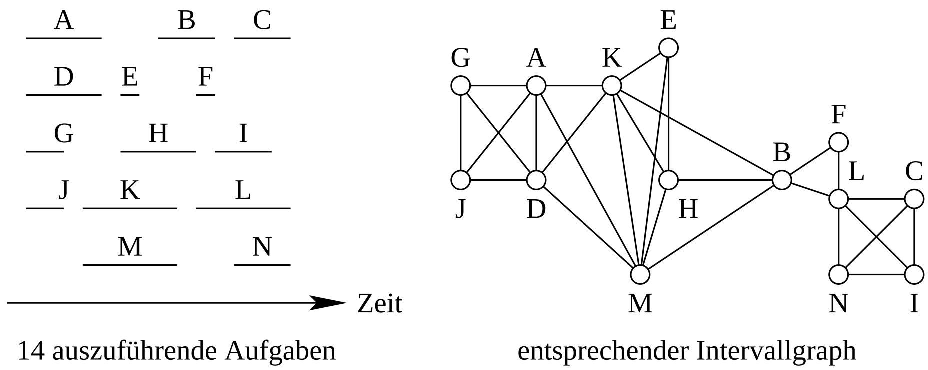


Abbildung C.1: Eine Menge von Aufgaben und der entsprechende Intervallgraph.

In der Sache mit dem Diebstahl aus dem Kantonsarchiv fasst Costello die Aussagen der Tatverdächtigen mithilfe der Tabelle aus Abbildung C.2 auf der nächsten Seite zusammen, in der ein Kreuz in einem Feld dafür steht, dass die Person in der entsprechenden Zeile ausgesagt hat, der Person in der entsprechenden Spalte begegnet zu sein.

Donnerstag, 9. April 2009

	T	B	E	S	L	G	M
T			x	x	x		
B							
E	x			x			
S	x		x		x	x	x
L	x			x			x
G				x			x
M				x	x	x	

Freitag, 10. April 2009

	T	B	E	S	L	G	M
T		x			x	x	
B	x			x		x	
E				x	x	x	
S		x	x			x	
L	x		x	x			
G	x	x	x	x			
M							

Abbildung C.2: Tabellen mit Kreuzen zu den Aussagen der Tatverdächtigen.

Wenn jeder Tatverdächtige nicht mehr als ein Mal pro Tag in das Kantonsarchiv gegangen ist, hat jeder eine gewisse Zeitspanne an diesem Ort verbracht. Folglich kann man den Aussagen einen Intervallgraphen zuordnen, indem man einen Knoten pro Tatverdächtigen anlegt und zwei Knoten miteinander verbindet, wenn sich die ihnen zugeordneten Intervalle zeitlich überschneiden, anders ausgedrückt, wenn sich diese beiden Tatverdächtigen begegnet sind (wenn also in Costellos Tabelle ein Kreuz steht). Auf dieser Basis konstruiert Manori die nachfolgenden Graphen, die im Prinzip zwei Intervallgraphen entsprechen, einem für die Intervalle vom Donnerstag und dem anderen für die Kreuze am Freitag.

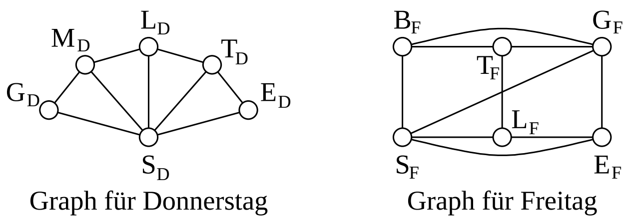


Abbildung C.3: Die beiden Intervallgraphen für Donnerstag und Freitag.

Um den Schuldigen zu ermitteln, zeigt Manori unter anderem, dass der rechte Graph kein Intervallgraph ist. Man kann sich daher fragen, woran man einen Intervallgraphen erkennt. Alle Intervallgraphen weisen die nachfolgende Eigenschaft auf, und sie ergibt sich daraus, dass die Zeit linear und nicht zirkular ist.

Satz C.1. Wenn G ein Intervallgraph ist, dann existiert kein Kreis, der mehr als drei Kanten hat und keine Abkürzung besitzt.

Der Ausdruck ›keine Abkürzung‹ ist im letzten Satz sehr wichtig. Der linke Graph aus Abbildung C.3 ist beispielsweise ein Intervallgraph, obwohl ein Kreis der Länge 6 existiert, der aus den 6 Kanten auf dem Rand des Graphen besteht. Dieser Kreis weist allerdings Abkürzungen auf. Anstatt sich zum Beispiel von S_D über G_D nach M_D zu bewegen, kann man die Abkürzung über die direkte Kante zwischen S_D und M_D nehmen.

Manori hat seinen Kollegen gezeigt, dass der Graph für Freitag zwei Quadrate enthält, das heißt, zwei Kreise mit 4 Kanten ohne Abkürzung. Der Graph für Freitag ist deshalb kein Intervallgraph, was darauf hindeutet, dass einer der Tatverdächtigen am Freitag mehr als ein Mal im Kantonsarchiv gewesen ist.

Die Intervallgraphen haben noch andere interessante Eigenschaften. Die nachfolgend beschriebene Eigenschaft versetzt uns in die Lage, das Problem vom Anfang dieses Kapitels zu lösen. Wir können damit nämlich die minimale Anzahl von Angestellten bestimmen, die gebraucht werden, um alle Aufgaben termingerecht auszuführen.

Definition C.2. Ein Knoten ist *simplizial*, wenn seine Nachbarn (das sind die Knoten, mit denen er durch eine Kante verbunden ist) untereinander alle durch jeweils eine Kante verbunden sind.

Definition C.3. Ein *perfektes Eliminationschema* in einem Graphen aus n Knoten ist eine Ordnung $v_1, \ldots, v_n$ der Knoten, sodass v_i in dem Graphen simplizial ist, der nur die Knoten $v_i, \ldots, v_n$ enthält.

Mit anderen Worten: Ist $v_1, \ldots, v_n$ ein perfektes Eliminationschema, so sind alle Nachbarn von v_1 untereinander verbunden. Außerdem gilt: Entfernt man den Knoten v_1 aus dem Graphen, so sind die Nachbarn von v_2 untereinander verbunden, usw. Das heißt, entfernt man die Knoten $v_1, \ldots, v_{i-1}$ aus dem Graphen, so sind alle Nachbarn von v_i untereinander verbunden.

Satz C.2. Ein Graph ist genau dann ein Intervallgraph, wenn er ein perfektes Eliminationsschema besitzt.

Es ist sehr einfach, ein perfektes Eliminationsschema in einem Intervallgraphen zu finden. Es reicht aus, einen simplizialen Knoten auszuwählen (das heißt, einen Knoten, dessen Nachbarn alle untereinander verbunden sind) und ihn mit v_1 zu bezeichnen. Dann entfernt man diesen Knoten aus dem Graphen und sucht nach einem simplizialen Knoten in dem Restgraphen. Diesen Knoten nennt man v_2. Dieses Verfahren setzt man solange fort, bis der Restgraph leer ist.

Kommen wir auf das ursprüngliche Problem zurück, das darin bestand, Aufgaben an Angestellte zu vergeben, wobei jede Aufgabe eine genaue Anfangs- und Endzeit hat. Dieses Problem lässt sich auf die Aufgabe zurückführen, die Knoten eines Intervallgraphen unter der Bedingung zu färben, dass zwei Knoten, die durch eine Kante verbunden sind, nicht dieselbe Farbe erhalten dürfen. Mit anderen Worten: Jede Farbe entspricht einem Angestellten, und es ist verboten, einem Angestellten zwei Aufgaben zu übertragen, die sich zeitlich überschneiden. Will man die Anzahl der Angestellten minimieren, die zur Erfüllung der Aufgaben gebraucht werden, ist das äquivalent dazu, eine Färbung der Knoten mit einer minimalen Anzahl von Farben zu bestimmen.

Eine solche Färbung kann man leicht mit dem folgenden Verfahren erhalten, in dem man annimmt, dass die Farben in einer bestimmten Reihenfolge vergeben werden.

Färbung der Knoten in einem Intervallgraphen mit einer minimalen Anzahl von Farben

1. Bestimmen Sie ein perfektes Eliminationsschema $v_1, \dots, v_n$.

2. Färben Sie die einzelnen Knoten nacheinander entsprechend der umgekehrten Reihenfolge $v_n, \dots, v_i$, wobei Sie dafür immer die zuerst verfügbare Farbe wählen.

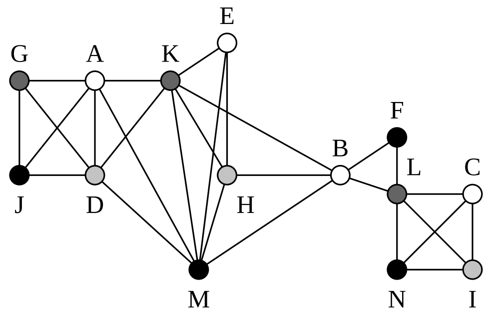
Im Problem mit den 14 Aufgaben vom Anfang dieses Kapitels sieht man beispielsweise, dass der Knoten C simplizial ist und man ihn daher als v_1 auswählen kann. Wir hätten auch G, J oder I auswählen können. Den Knoten L hätten wir dagegen nicht auswählen können, weil seine Nachbarn F und B nicht mit den anderen Nachbarn C, I und N verbunden sind. An die zweite Stelle kann man zum Beispiel den Knoten I bringen, anschließend den Knoten N an die dritte Stelle. Der Knoten L ist dann simplizial, weil seine Nachbarn in dem Restgraphen, der sich durch Entfernen der Knoten C, I und N ergeben hat, die Knoten F und B sind, die untereinander verbunden sind. Man kann daher den Knoten L an die vierte Stelle bringen. Man setzt dann in gleicher Art und Weise fort, um beispielsweise die folgende Ordnung zu erhalten:

$$C, I, N, L, F, B, E, H, K, M, A, D, G, J\,.$$

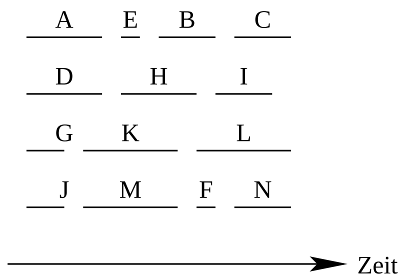
Nun kann man den Graphen färben. Angenommen, die Farben sind von dunkel nach hell geordnet. Man beginnt folglich damit, dem Knoten J die Farbe Schwarz zuzuordnen. Dann ordnet man G das dunkle Grau zu, das helle Grau dem Knoten D und dem Knoten A die Farbe Weiß. Der Knoten M, der als nächstes zu färben ist, kann die Farben Dunkelgrau oder Schwarz erhalten. Da diese letzte Farbe vorrangig ist, wählt man sie für den Knoten M. Man setzt dann in gleicher Art und Weise fort, bis man eine Färbung des gesamten Graphen erreicht hat, die im linken Teil von Abbildung C.4 auf der nächsten Seite dargestellt ist.

Diese Färbung verwendet vier Farben, was bedeutet, dass vier Angestellte gebraucht werden, um die 14 Aufgaben zu erfüllen. Die Zuordnung der Aufgaben an vier Angestellte ist im rechten Teil von Abbildung C.4 auf der nächsten Seite dargestellt. Jede Zeile entspricht einer Farbe und somit einem Angestellten. Beispielsweise übernimmt der durch die Farbe Weiß vertretene Angestellte die Aufgaben A, B, C und E, weil diese Knoten weiß gefärbt sind.

Im letzten Beispiel hatten wir die Aufgabe, eine Aufteilung der Knoten eines Intervallgraphen in eine minimale Anzahl von Teilmengen zu bestimmen, sodass es keine Kante gibt, die zwei Knoten ein und derselben Teilmenge verbindet. Einige Situationen



Färbung in umgekehrter Reihenfolge
des perfekten Eliminationsschemas
C, I, N, L, F, B, E, H, K, M, A, D, G, J



Die Aufgaben können von vier Angestellten
übernommen werden. Jede Zeile
entspricht der Arbeit eines Angestellten.

Abbildung C.4: Lösung des Problems, 14 Aufgaben auf möglichst wenige Angestellte zu verteilen.

können mithilfe eines Intervallgraphen modelliert werden, in dem man eine Aufteilung der Knoten in eine minimale Anzahl von Teilmengen bestimmen muss, in denen diesmal zwischen zwei Knoten ein und derselben Teilmenge keine Kante fehlt. In Farben ausgedrückt, möchte man diesmal die Knoten so färben, dass alle Knoten ein und derselben Farbe untereinander durch eine Kante verbunden sind. Es folgen drei Beispiele für solche Situationen.

Beispiel C.2. Ein Lagerhaus für Pharmaprodukte möchte einen Vorrat an Produkten $p_1, \ldots, p_n$ aufbewahren. Jedes Produkt p_i hat eine minimale Lagertemperatur m_i und eine maximale Lagertemperatur M_i. Man möchte die minimale Anzahl von Kühlschränken bestimmen, die zur Aufbewahrung der Produkte gebraucht werden.

Um dieses Problem zu lösen, konstruieren wir einen Intervallgraphen, in dem die Knoten für die Produkte stehen und zwei Produkte p_i und p_j durch eine Kante verbunden sind, wenn die Schnittmenge der Intervalle $[m_i, M_i]$ und $[m_j, M_j]$ nicht leer ist. Man möchte also Produktgruppen so erstellen, dass alle Produkte einer Gruppe untereinander durch eine Kante verbunden sind (was bedeutet, dass es eine Lagertemperatur gibt, die für alle Produkte aus dieser Gruppe passend ist).

Beispiel C.3. Die Personen $P_1, \ldots, P_n$ haben beschlossen, ihren Urlaub im Hotel Bellevue zu verbringen. Person P_i wird vom Tag A_i bis zum Tag B_i bleiben. Der Hoteldirektor hat dem Verwalter die Verantwortung für den guten Verlauf des Aufenthalts der Urlauber überlassen und ist demnach nur sehr selten im Hotel. Eigentlich begibt sich der Direktor nur ab und zu ins Hotel, um die gerade anwesenden Gäste persönlich kennenzulernen und sie auf diese Weise an das Hotel zu binden. Der Direktor möchte seine n Gäste so kennenlernen, dass die Anzahl der Tage, die er dafür im Hotel Bellevue verbringen muss, minimal ist, denn er ist ein sehr beschäftigter Mann. Vor allem liegt das Hotel in einer sehr abgeschiedenen Ecke des Landes, die schwer zu erreichen ist.

Um die Anzahl der Besuche zu minimieren, die der Direktor dem Hotel abstatten muss, um alle seine Gäste kennenzulernen, kann man einen Intervallgraphen konstruieren, in dem die Knoten die Hotelgäste sind und zwei Gäste P_i und P_j durch eine Kante verbunden sind, wenn die Schnittmenge der Intervalle $[A_i,B_i]$ und $[A_j,B_j]$ nicht leer ist. Man möchte also Gruppen so bilden, dass alle Gäste einer Gruppe untereinander durch eine Kante verbunden sind (was bedeutet, dass es einen Tag gibt, an dem alle Gäste der Gruppe im Hotel anwesend sind, und sie der Direktor folglich kennenlernen kann).

Beispiel C.4. Ein Theaterliebhaber möchte sich n Theaterstücke $P_1,\ldots,P_n$ ansehen, die demnächst in Montréal auf dem Spielplan stehen werden. Jedes Stück P_i wird in einem zusammenhängenden Zeitraum auf dem Spielplan stehen, und zwar vom Tag D_i bis zum Tag F_i. Wenn an einem Tag mehrere Theaterstücke auf dem Spielplan stehen, dann werden sie zu verschiedenen Zeiten gezeigt, damit man sie alle an demselben Tag besuchen kann. Der Theaterliebhaber wohnt eine Autostunde von Montréal entfernt und möchte unbedingt jede Nacht zu Hause schlafen. Wenn er also Theatervorstellungen an zwei aufeinanderfolgenden Tagen besucht, dann wird er sich zwei Mal nach Montréal begeben. Wie oft wird er sich mindestens nach Montréal begeben müssen, um alle Theaterstücke gesehen zu haben?

Zur Lösung dieses Problems kann man einen Intervallgraphen konstruieren, in dem die Knoten die Theaterstücke sind und zwei Theaterstücke P_i und P_j durch eine Kante verbunden sind, wenn die Schnittmenge der Intervalle $[D_i,F_i]$ und $[D_j,F_j]$ nicht leer ist. Man möchte also Gruppen so bilden, dass alle Theaterstücke einer Gruppe untereinander durch eine Kante miteinander verbunden sind (was bedeutet, dass es einen Tag gibt, an dem sie der Theaterliebhaber alle sehen kann).

Zur Lösung dieser drei Probleme muss man folglich eine Aufteilung der Knoten eines Intervallgraphen in eine minimale Anzahl von Gruppen bestimmen, sodass alle Knoten einer Gruppe untereinander durch eine Kante verbunden sind. Dazu kann man sie wie folgt auswählen. Man beginnt, indem man die Knoten nach der unteren Grenze des zugehörigen Intervalls in aufsteigender Reihenfolge sortiert. In Beispiel C.2 ordnet man die Produkte also aufsteigend nach der Temperatur m_i, in Beispiel C.3 sind es die Ankunftstage A_i, wohingegen es in Beispiel C.4 die Premierentage D_i sind.

Man besucht dann die Knoten in dieser Reihenfolge. Man setzt den ersten Knoten in eine erste Gruppe, und die Regel für die anderen Knoten ist wie folgt: Kann der betrachtete Knoten in die Gruppe eingefügt werden, die gerade konstruiert wird, fügt man ihn in diese Gruppe ein; anderenfalls schließt man die Gruppe und setzt den betrachteten Knoten in eine neue Gruppe.

Zur Illustration betrachten wir Beispiel C.4 mit den $n=5$ Theaterstücken aus Abbildung C.5 auf der nächsten Seite. Die Reihenfolge der Theaterstücke ist somit $2,3,1,4,5$.

i	1	2	3	4	5
D_i	26. Juni	20. Juni	22. Juni	17. Juli	19. Juli
F_i	22. Juli	24. Juni	15. Juli	23. Juli	20. Juli

Abbildung C.5: Beispiel mit fünf Theaterstücken.

Wir setzen das zweite Stück demnach in eine erste Gruppe. Das dritte Stück kann ebenfalls in diese Gruppe eingeteilt werden, weil es mit dem zweiten Stück einen Termin gemeinsam hat. Das erste Stück passt nicht in diese Gruppe, weil seine Premiere erst nach der letzten Vorstellung des zweiten Stücks stattfindet. Wir konstruieren also eine neue Gruppe für dieses Stück. Die Stücke vier und fünf werden dann in dieselbe Gruppe einsortiert, weil sie mit dem ersten Stück Vorstellungen am 19. Juli und 20. Juli gemeinsam haben. Der Theaterliebhaber kann somit am 22., 23. oder 24. Juni nach Montréal kommen, um die Stücke zwei und drei zu sehen, und am 19. oder am 20. Juli für die Stücke $1,4$ und 5.

Abbildung C.6 zeigt den diesem Problem entsprechenden Intervallgraphen mit einer Färbung der Knoten durch zwei Farben. Alle Knoten einer Farbe (das bedeutet ein und derselben Gruppe) sind darin untereinander durch eine Kante verbunden.

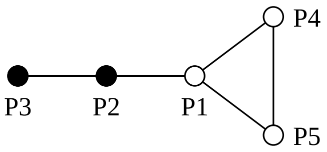


Abbildung C.6: Ein Intervallgraph mit einer Färbung der Knoten.

D Kantengraphen

Einem gegebenen Graphen G kann man einen neuen Graphen H zuordnen, den man auch als $L(G)$ bezeichnet. Er heißt *Kantengraph* oder *Line-Graph* von G. Einen solchen Graphen erhält man, indem man für jede Kante von G in $H = L(G)$ einen Knoten setzt und zwei Knoten in H genau dann verbindet, wenn die entsprechenden Kanten in G einen gemeinsamen Knoten haben.

Im nachfolgenden Beispiel enthält der Graph G genau 6 Kanten, was bedeutet, dass $L(G)$ genau 6 Knoten enthält. Die Knoten $[a,b]$ und $[a,c]$ sind in $L(G)$ durch eine Kante verbunden, weil die entsprechenden Kanten in G den Knoten a gemeinsam haben. Dagegen existiert keine einzige Kante zwischen den Knoten $[a,c]$ und $[b,e]$, denn die beiden Kanten haben in G keinen Knoten gemeinsam.

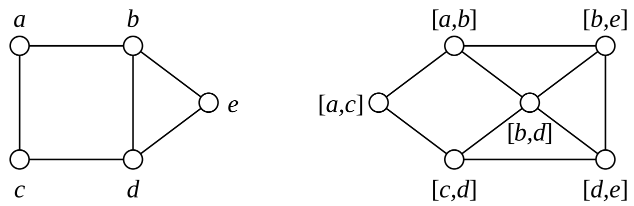


Abbildung D.1: Ein Graph und sein Kantengraph $L(G)$.

Es gibt Graphen, die keine Kantengraphen sind. Der folgende Graph H ist beispielsweise kein Kantengraph. Anderenfalls müsste ein Graph G existieren, für den $H = L(G)$ ist. Dieser Graph G müsste drei Kanten A,C und D haben, die keinen Knoten gemeinsam haben, und eine vierte Kante B, die mit den Kanten A, C und D einen Knoten gemeinsam hat. Das ist natürlich unmöglich, weil jede Kante nur zwei Enden hat.

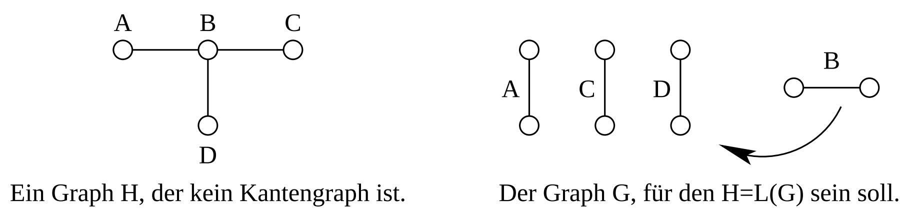


Abbildung D.2: Ein Graph, der kein Kantengraph ist.

Bedenken Sie: Hätte der obige Graph eine zusätzliche Kante, zum Beispiel eine Kante, die A mit D verbindet, so wäre H ein Kantengraph, denn die Kante B hätte dann einen Knoten (ein Ende) mit A und D gemeinsam und den anderen Knoten (das andere Ende) gemeinsam mit C.

Ein Kantengraph H. Der Graph G, für den H=L(G) ist.

Abbildung D.3: Ein Graph, der ein Kantengraph ist.

Es existieren genau neun Konfigurationen, die keinem Kantengraphen entsprechen. Sie sind in Abbildung D.4 dargestellt. Enthält ein Graph H mindestens eine dieser 9 Konfigurationen in der dargestellten Form (das heißt, ohne zusätzliche Kante), so ist er kein Kantengraph. Die vorletzte dieser Konfigurationen ist diejenige, die uns vorhin im Beispiel gegeben war.

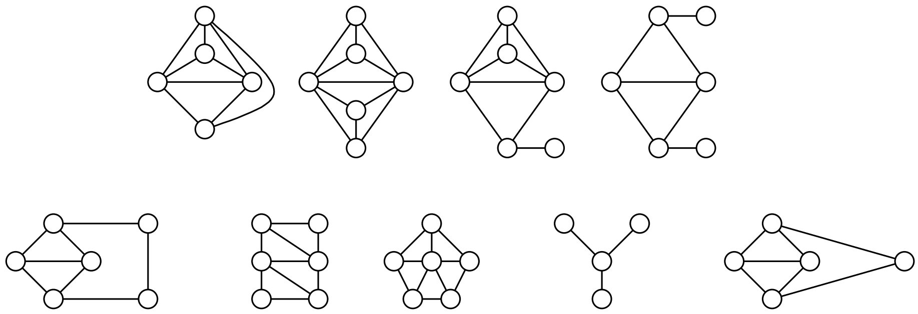


Abbildung D.4: Neun Konfigurationen, die keinem Kantengraphen entsprechen.

Außerdem gilt: Enthält ein Graph H keine dieser neun Konfigurationen, so existiert ein eindeutiger Graph G mit $H = L(G)$, insofern keine Zusammenhangskomponente von G ein Dreieck ist. Der Graph auf der linken Seite von Abbildung D.1 auf der vorherigen Seite vom Anfang dieses Kapitels ist also der eindeutige Graph G, dessen Kantengraph der Graph auf der rechten Seite dieser Abbildung ist.

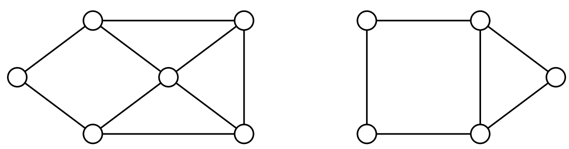


Abbildung D.5: Ein Kantengraph H und sein eindeutiger Graph G.

Die eben erwähnte Einschränkung hinsichtlich der Graphen G, deren Zusammenhangskomponenten Dreiecke sind, ergibt sich, weil es zwei Graphen G_1 und G_2 gibt, für die $L(G_1)$ und $L(G_2)$ Dreiecke sind (siehe Abbildung D.6).

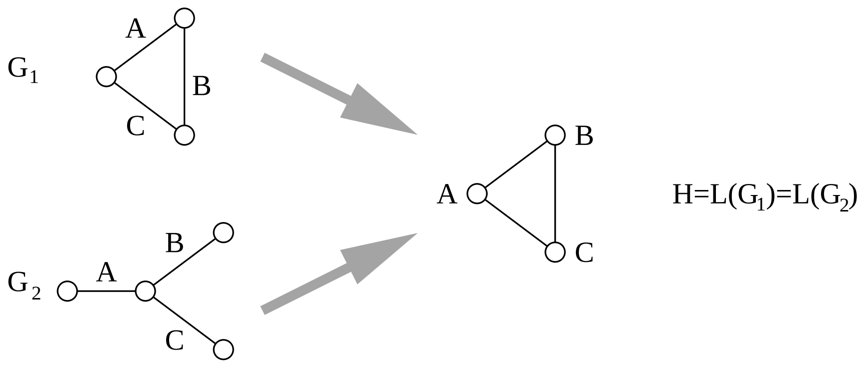


Abbildung D.6: Zwei Graphen G_1 und G_2, deren Kantengraph ein Dreieck ist.

Um den Betrüger in der Erbangelegenheit zu entlarven, konstruiert Bonvin einen Graphen, der das Geländespiel vollständig widergibt, einschließlich der Informationen, die in dem Teil des Dokuments enthalten sind, das angeblich von dem Lösegelderpresser gefunden wurde. Dazu konstruiert er einen Graphen, der für jeden Weg einen Knoten enthält, und er verbindet zwei Wege, wenn die beiden ein Ende gemeinsam haben. Der von ihm konstruierte Graph muss nämlich zu einem Kantengraphen gehören, in dem die Knoten die Kreuzungen an den Wegenden sind und die Kanten die Wege selbst. Bonvin zeigt Manori den folgenden Graphen, und Manori ist sich schnell darüber im Klaren, dass es sich dabei um die sechste verbotene Konfiguration handelt.

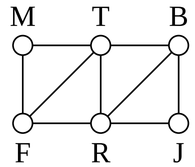


Abbildung D.7: Ein Graph, der keinem Kantengraphen entspricht.

In dem Teil des Dokuments, das der Lösegelderpresser angeblich gefunden hat, erweckt es den Anschein, dass der Maiglöckchenweg und der Begonienweg (also die Knoten M und B) keine Enden gemeinsam haben. Das ist die einzige Information, die falsch sein kann, weil sie keinen Zweifel über die Gültigkeit des übrigen Dokuments aufwirft. Nimmt man dagegen an, dass der Maiglöckchenweg und der Begonienweg ein Ende gemeinsam haben, muss man eine Kante zwischen M und B ergänzen. Und diesmal erhält man einen Kantengraphen, wie von Manori gezeigt.

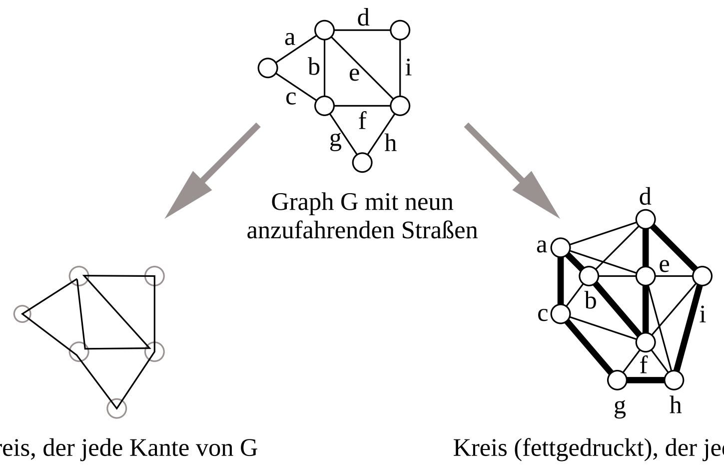
Der dem Geländespiel entsprechende Kantengraph H.

Der Graph G mit H=L(G).

Abbildung D.8: Ein Kantengraph und sein Ursprungsgraph.

Kantengraphen können verwendet werden, um eine andere Darstellung der Situation zu gewinnen. Betrachten wir etwa das Problem der Hausmüllentsorgung. Nehmen wir genauer an, dass ein Transportunternehmen mit der Hausmüllentsorgung in einem Viertel von Montréal beauftragt ist. Ihm wurde ein Plan des Viertels zur Verfügung gestellt, auf dem die anzufahrenden Straßen verzeichnet sind. Dieses Unternehmen möchte nun die Entsorgung mit nur einem einzigen Fahrzeug realisieren. Um außerdem die Anzahl der gefahrenen Kilometer zu minimieren, möchte das Unternehmen gerne wissen, ob eine Tour existiert, die jede Straße des Viertels genau ein Mal durchläuft. Das Fahrzeug soll also keinen einzigen überflüssigen Kilometer fahren. Mit den Worten der Graphentheorie: Das Unternehmen möchte wissen, ob in dem Graphen ein *Eulerkreis* existiert.

Definition D.1. Ein Kreis, der jede Kante eines Graphen genau ein Mal durchläuft, heißt *Eulerkreis*.



Graph G mit neun anzufahrenden Straßen

Kreis, der jede Kante von G genau ein Mal durchläuft (Eulerkreis)

Kreis (fettgedruckt), der jeden Knoten von L(G) genau ein Mal besucht (Hamiltonkreis)

Abbildung D.9: Korrespondenz zwischen Euler- und Hamiltonkreisen.

Wenn man den Kantengraphen $L(G)$ von G betrachtet, sollte man sich somit fragen, ob eine Tour existiert, die jeden Knoten von $L(G)$ genau ein Mal besucht. Ein solcher Kreis heißt *Hamiltonkreis*.

Definition D.2. Ein Kreis, der jeden Knoten eines Graphen genau ein Mal besucht, heißt *Hamiltonkreis*.

Mit anderen Worten: Um herauszufinden, ob in einem Graphen G ein Eulerkreis existiert, kann man auch den Kantengraphen $L(G)$ konstruieren und prüfen, ob in $L(G)$ ein Hamiltonkreis existiert. Im nachfolgenden Beispiel nehmen wir an, dass das Unternehmen den Müll in neun Straßen $a, b, \ldots, i$ entsorgen muss. In Abbildung D.9 auf der vorherigen Seite haben wir unten links einen Eulerkreis in G gekennzeichnet, während der Graph rechts denselben Kreis darstellt, diesmal aber als Hamiltonkreis (fettgedruckt) in $L(G)$.

E Paarung und Färbung von Kanten

Betrachten wir n Hockeymannschaften, die bei einem Turnier gegeneinander antreten sollen. Nehmen wir an, dass jede Mannschaft genau ein Mal gegen jede andere Mannschaft antreten soll. Man kann sich zunächst die Frage stellen, wie groß die maximale Anzahl von Begegnungen ist, die an einem Tag unter der Bedingung stattfinden können, dass keine Mannschaft zwei Spiele an einem Tag austragen muss.

Die Antwort ist einfach: Ist n gerade, so können $n/2$ Begegnungen an einem Tag stattfinden; ist n ungerade, so reduziert sich diese Zahl auf $(n-1)/2$, was bedeutet, dass zumindest eine Mannschaft nicht jeden Tag spielt.

Mit den Worten der Graphentheorie: Man setzt für jede Mannschaft einen Knoten und verbindet jedes Knotenpaar durch eine Kante. Die Kanten entsprechen also den Begegnungen, die geplant werden sollen. Die oben gestellte Frage lässt sich nun in die Frage umformulieren, wie groß die maximale Anzahl der Kanten ist, die man im Graphen wählen kann, ohne dass zwei gewählte Kanten einen Knoten gemeinsam haben. Wonach man hier sucht, nennt sich eigentlich *Paarung* mit maximaler Kantenzahl.

Definition E.1. Eine Paarung in einem Graphen ist eine Menge von Kanten, von denen keine zwei Kanten einen Knoten gemeinsam haben.

Man kann sich auch fragen, was die minimale Anzahl von Tagen ist, die gebraucht werden, um alle Begegnungen wie gewünscht stattfinden zu lassen. Unter der Bedingung, dass jede Mannschaft jeder anderen Mannschaft genau ein Mal begegnet, geht es um die Planung von $n(n-1)/2$ Spielen.

- Für gerade n haben wir gesehen, dass pro Tag maximal $n/2$ Spiele stattfinden können, und daher dauert es mindestens $n-1$ Tage, bis alle Spiele stattgefunden haben.
- Für ungerade n haben wir gesehen, dass pro Tag maximal $(n-1)/2$ Spiele stattfinden können, man braucht daher für alle Spiele mindestens n Tage.

Mit den Worten der Graphentheorie: Wir wollen die Kanten eines Graphen färben, indem wir dabei so wenige Farben wie möglich derart verwenden, dass jede Farbe einer Paarung entspricht. Wir ordnen also jedem Tag eine Farbe zu.

Es ist wirklich einfach, alle $n(n-1)/2$ Spiele an $n-1$ Tagen zu planen, wenn n gerade ist. Dabei kann man wie folgt vorgehen. Wir nummerieren die Knoten für die n Mannschaften von 1 bis n. Wir zeichnen den oben erwähnten Graphen, indem wir den

Knoten n in der Mitte platzieren und die anderen Knoten kreisförmig um den Knoten n anordnen.

- Am ersten Tag organisiert man Begegnungen zwischen den Mannschaften 1 und n, 2 und $n-1$, 3 und $n-2$ bis zur Begegnung zwischen den Mannschaften $n/2$ und $n/2+1$.
- Am nachfolgenden Tag nimmt man die Paarung vom vorherigen Tag und modifiziert sie, indem man die Paarungskanten im Uhrzeigersinn dreht.

Da eine Zeichnung mehr sagt als lange Erklärungen, ist hier eine Illustration dieser Konstruktion für eine Menge von $n = 6$ Mannschaften.

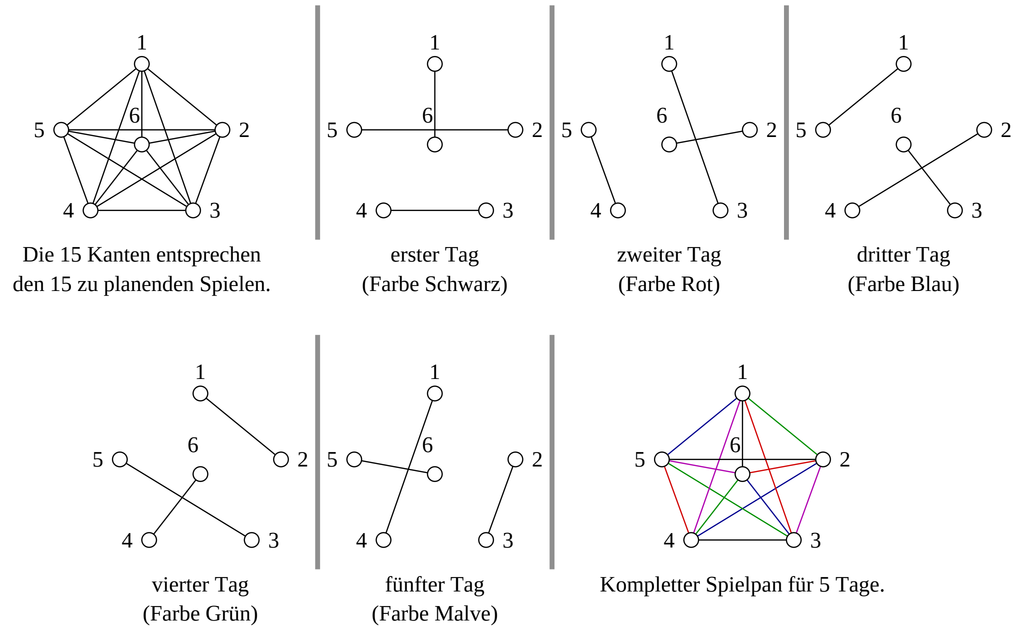


Abbildung E.1: Konstruktion einer Paarung für eine gerade Kantenzahl.

Der Fall mit ungeradem n lässt sich nun ebenfalls leicht planen. Es reicht nämlich aus, eine fiktive Mannschaft mit der Nummer $n+1$ hinzuzunehmen. Wenn eine Mannschaft gegen die Mannschaft $n+1$ spielt, bedeutet das einfach, dass diese Mannschaft einen Ruhetag hat. Man muss daher einen Spielplan mit $n+1$ Teams konstruieren, wobei $n+1$ eine gerade Zahl ist. Wir haben gesehen, dass ein solcher Spielplan existiert. Er nimmt nur $(n+1)-1 = n$ Tage in Anpsruch. Und wir haben gesehen, dass kein besserer Spielplan existiert. Um den Spielplan für die ursprünglichen n Mannschaften zu erhalten, muss man dann einfach alle Spiele mit der fiktiven Mannschaft $n+1$ streichen.

Spielen in dem Turnier zum Beispiel $n = 5$ Mannschaften, ergänzt man eine fiktive 6. Mannschaft und konstruiert den vorhin angegebenen Zeitplan. Anschließend streicht man alle Spiele, die Mannschaft 6 betreffen, um den nachfolgenden Zeitplan für 5 Mannschaften in 5 Tagen zu erhalten.

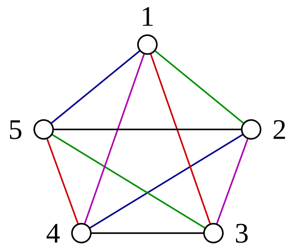


Abbildung E.2: Zeitplan für 5 Mannschaften.

Paarungen und Kantenfärbungen kommen auch in vielen anderen Zusammenhängen vor. Nehmen wir zum Beispiel an, dass man 5 Tätigkeiten T_1, T_2, T_3, T_4 und T_5 erledigen muss, von denen jede eine Zeitspanne des Tages in Anspruch nimmt. Die Tätigkeiten T_1 und T_2 müssen vom Angestellten E_1 erledigt werden, die Tätigkeiten T_3 und T_4 vom Angestellten E_2 und die Tätigkeit T_5 vom Angestellten E_3. Für die Tätigkeiten T_1 und T_3 wird die Maschine M_1 gebraucht, für die Tätigkeiten T_2 und T_4 wird die Maschine M_2 gebraucht und für die Tätigkeit T_5 die Maschine M_3. Berücksichtigt man, dass sich jeder Angestellte nur einer Tätigkeit auf einmal widmen kann und dass jede Maschine nur von einem Angestellten auf einmal verwendet werden kann, wollen wir wissen:

- Wie viele Tätigkeiten können an einem Tag maximal erledigt werden?
- Wie viele Tage dauert es mindestens, bis die 5 Tätigkeiten erledigt sind?

Man kann dieses Problem natürlich auf eine beliebige Anzahl von Tätigkeiten, Angestellten und Maschinen verallgemeinern.

Diese beiden Probleme können mithilfe eines Graphen modelliert werden. Die Knoten sind die Angestellten E_1, E_2 und E_3 sowie die Maschinen M_1, M_2 und M_3. Man verbindet dann einen Knoten E_i mit einer Maschine M_j, wenn eine Tätigkeit den Angestellten E_i und die Maschine M_j beansprucht. Die 5 Tätigkeiten aus unserem Beispiel werden daher mithilfe von 5 Kanten dargestellt. Um die erste Frage zu beantworten, reicht es folglich aus, in diesem Graphen eine Paarung mit einer maximalen Anzahl von Kanten zu bestimmen. In unserem Beispiel beinhaltet die größte Paarung drei Kanten. Man kann $[E_1, M_1]$, $[E_2, M_2]$ und $[E_3, M_3]$ wählen, was den Tätigkeiten T_1, T_4 und T_5 entspricht. Man hätte auch $[E_1, M_2]$, $[E_2, M_1]$ und $[E_3, M_3]$ wählen können, was den Tätigkeiten T_2, T_3 und T_5 entsprechen würde.

Das zweite Problem ist äquivalent zu einer Kantenfärbung desselben Graphen mithilfe einer minimalen Anzahl von Farben, sodass benachbarte Kanten verschiedene Farben haben. Jede Farbe entspricht damit einer Paarung (nicht zwangsläufig der maximalen) und stellt die Menge der Tätigkeiten dar, die an einem gegebenen Tag bewältigt werden. In unserem Beispiel dauert es zwei Tage, die 5 Tätigkeiten zu erledigen. Man kann zum Beispiel am 1. Tag die Tätigkeiten T_1, T_4 und T_5 erledigen (feine Linien) und die Tätigkeiten T_2 und T_3 am 2. Tag (dicke Linien).

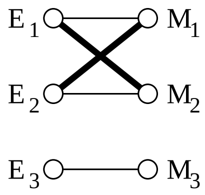


Abbildung E.3: Eine Kantenfärbung.

Definition E.2. Bei der Kantenfärbung eines Graphen G ordnet man allen Kanten eine Farbe zu, sodass Kanten, die an demselben Knoten liegen, verschiedene Farben haben. Im Allgemeinen will man eine Färbung finden, die zudem so wenig Farben wie möglich verwendet. Die kleinste Anzahl von Farben, die zur Färbung der Kanten eines Graphen benötigt werden, heißt *chromatischer Index* von G und wird mit $q(G)$ bezeichnet.

Sei $\Delta(G)$ der größte Grad eines Knotens in G. Unter der Bedingung, dass alle an einen Knoten liegenden Kanten verschiedene Farben haben, kann der chromatische Index $q(G)$ von G offensichtlich nicht kleiner als $\Delta(G)$ sein. Dagegen kann es sein, dass $q(G)$ echt größer als $\Delta(G)$ ist. Zum Beispiel können die Kanten des nachfolgenden Fünfecks nicht mit weniger als 3 Farben gefärbt werden, während $\Delta(G) = 2$ ist.

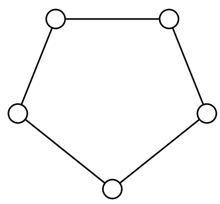
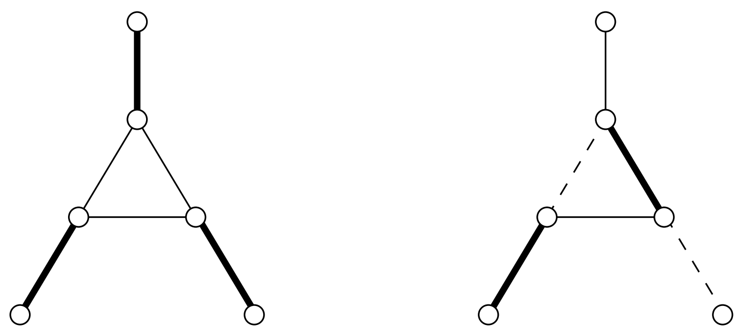


Abbildung E.4: Ein Fünfeck, das nicht mit weniger als 3 Farben gefärbt werden kann.

Satz E.1 (Satz von Vizing). Die folgenden Ungleichungen gelten für alle Graphen G: $\Delta(G) \leq q(G) \leq \Delta(G) + 1$.

Vizing hat selbst eine Prozedur angegeben, mit deren Hilfe man eine Kantenfärbung von G mit $\Delta(G) + 1$ Farben bestimmen kann. Sein Algorithmus kann sich also um eine Einheit gegenüber dem chromatischen Index vertun, was nicht wesentlich ist. Dagegen ist es sehr schwierig, festzustellen, ob $q(G)$ gleich $\Delta(G)$ oder $\Delta(G)+1$ sein muss, wenn G keine besondere Eigenschaft aufweist.

Will man $q(G)$ bestimmen, so kann man beispielsweise eine erste Paarung mit einer maximalen Anzahl von Kanten in G bestimmen, dieser Paarung eine Farbe zuweisen und sie aus dem Graphen entfernen. Durch Wiederholen dieses Vorgangs wird man alle Kanten des Graphen färben und man kann hoffen, auch die minimal mögliche Anzahl von Farben verwendet zu haben, weil man bei jedem Schritt die maximale Anzahl von Kanten sucht, die mit einer Farbe gefärbt werden können. Diese Vorgehensweise färbt G jedoch nicht immer mit $q(G)$ Farben. Im nachfolgenden Graphen enthält die größte Paarung beispielsweise 3 Kanten: Es handelt sich um die drei Kanten, die an Knoten vom Grad 1 liegen. Wenn man diese drei Kanten gleich einfärbt, bleibt noch das Dreieck übrig, zu dessen Färbung drei neue Farben gebraucht werden, weil die drei Kanten jeweils Knoten gemeinsam haben. Das eben beschriebene Verfahren liefert also eine Färbung mit insgesamt vier Farben, während der chromatische Index $q(G)$ gleich 3 ist, wie in der nachfolgenden Abbildung mit einfachen, fetten und gestrichelten Linien illustriert.



Graph, dessen größte Paarung sich aus 3 Kanten (fettgedruckt) zusammensetzt

Der chromatische Index q(G) ist 3.

Abbildung E.5: Färbung eines Graphen.

In bestimmten Fällen kann man hingegen einfach feststellen, ob $q(G) = \Delta(G)$ oder $q(G) = \Delta(G)+1$ ist. Das ist beispielsweise bei vollständigen Graphen der Fall, in denen keine einzige Kante fehlt.

Definition E.3. Ein Graph heißt *vollständig*, wenn zwischen allen Knotenpaaren eine Kante existiert.

Am Anfang dieses Kapitels haben wir gesehen, dass die Planung eines Turniers mit n Mannschaften äquivalent zu der Aufgabe ist, die Kanten eines vollständigen Graphen mit n Knoten zu färben. Wir haben gesehen, dass dafür $n-1$ Tage ausreichen, wenn n gerade ist, während man n Tage braucht, wenn n ungerade ist.

Satz E.2. Gegeben sei ein vollständiger Graph G mit n Knoten. Dann ist

- $q(G) = \Delta(G) = n - 1$, wenn n gerade ist,
- $q(G) = \Delta(G) + 1 = n$, wenn n ungerade ist.

Ein weiterer einfacher Fall liegt vor, wenn die Knoten des Graphen so in zwei Teilmengen N_1 und N_2 zerlegt werden können, dass jede Kante des Graphen genau an einem Knoten aus N_1 und an einem anderen Knoten aus N_2 liegt.

Definition E.4. Ein Graph G ist *bipartit*, wenn eine Zerlegung (N_1, N_2) seiner Knoten existiert, sodass jede Kante von G einen Endknoten in N_1 und den anderen in N_2 hat.

Im Beispiel über die Planung von Tätigkeiten, das wir vorhin betrachtet haben, geht es darum, die Kanten eines bipartiten Graphen zu färben. Allerdings ist die Zerlegung der Knoten einfach: Man kann die Angestellten in die Menge N_1 setzen und die Maschinen in die Menge N_2. Die Kanten, die für die Tätigkeiten stehen, verbinden immer einen Angestellten mit einer Maschine.

Satz E.3. Ist G ein bipartiter Graph, dann gilt $q(G) = \Delta(G)$.

In dem Graphen der fünf Tätigkeiten, den wir hier erneut abbilden, wird der höchste Grad von den Knoten E_1, E_2, M_1 und M_2 erreicht, und er ist gleich 2. Der chromatische Index ist also 2, was wir mithilfe von einfachen und fetten Linien dargestellt haben.

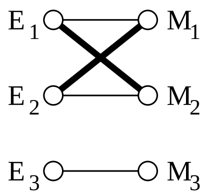


Abbildung E.6: Der chromatische Index.

Andere Beispiele In einem Cégep[1] von Montréal sollen n Lehrer Kurse in m Klassen geben. Jede Kante des Graphen auf der linken Seite der nachfolgenden Abbildung entspricht einem zu gebenden Kurs. Klar ist, dass ein Lehrer nicht zwei Kurse auf einmal geben kann und dass eine Klasse nicht gleichzeitig an zwei Kursen teilnehmen kann. Wie viele Unterrichtszeiträume müssen im Stundenplan vorgesehen werden, um alle möglichen Kurse zu erteilen?

[1] College d'enseignement général et professionel – Bildungseinrichtung, in der eine technische und voruniversitäre Ausbildung stattfindet.

Das ist ein Kantenfärbungsproblem für einen bipartiten Graphen. Die Anzahl der vorzusehenden Unterrichtszeiträume ist daher $q(G) = \Delta(G)$. In unserem Fall ist diese Zahl gleich 3, und ein 3-farbiger (in fetten, einfachen und gestrichelten Linien) Stundenplan ist auf der rechten Seite der nachfolgenden Abbildung dargestellt.

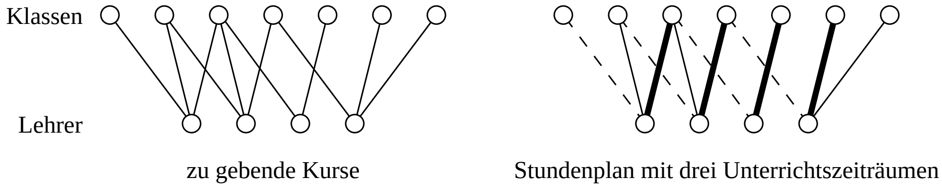


Abbildung E.7: Kantenfärbungsproblem für einen bipartiten Graphen.

Kommen wir auf einige Fälle unseres *Grafen der Graphen*, Inspektor Manori, zurück. Um Cindy ihr Lächeln wiederzugeben, führt er ihr einen Zaubertrick vor, in dem es darum geht, 9 Karten auf 9 Umschläge zu verteilen. Jeder Umschlag darf nur eine der von Manori vorgegebenen Karten enthalten. Die Situation ist mithilfe des nachfolgenden bipartiten Graphen dargestellt, in dem es darum geht, eine Paarung zu bestimmen, die alle Knoten umfasst.

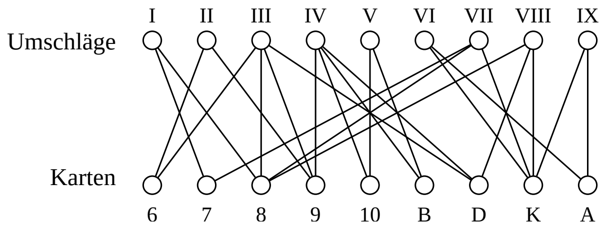


Abbildung E.8: Ein bipartiter Graph, um neun Karten auf neun Umschläge zu verteilen.

Indem er die Knoten so umordnete, dass Kantenüberschneidungen vermieden werden, gelang es Manori, die nachfolgende Darstellung desselben Graphen zu erhalten,

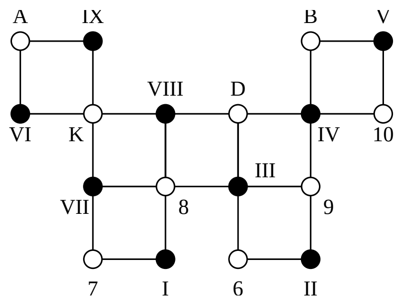


Abbildung E.9: Schwarz-Weiß-Darstellung des Graphen aus der vorherigen Abbildung.

wobei die Bipartition diesmal mithilfe der Farben Schwarz und Weiß an den Knoten gekennzeichnet ist. Schwarz steht für die Umschläge, Weiß für die Karten.

Manori demonstriert Cindy, dass in diesem Graphen jede Paarung mit 9 Kanten zwangsläufig eine Kante enthält, die den Knoten VIII mit dem Knoten für die Dame verbindet. Er zeigt ihr zwei derartige Paarungen.

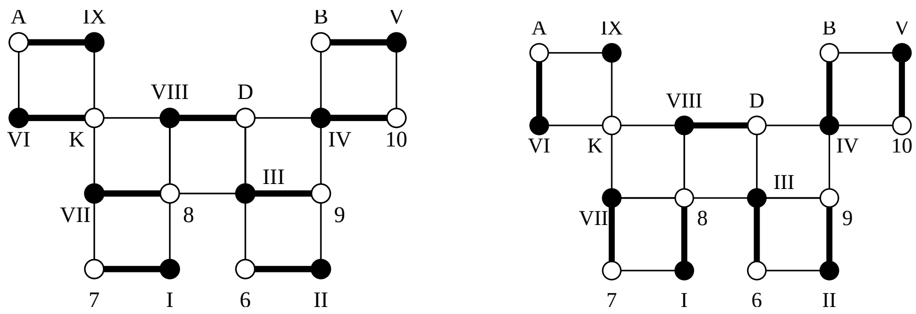


Abbildung E.10: Zwei verschiedene Paarungen zu neun Kanten.

Bevor er Bekanntschaft mit Cindy macht, führt Manori ein langes Gespräch mit Despontin, in dem sie über stabile Ehen sprechen. Eine Menge von n Ehen zwischen n Männern und n Frauen ist wiederum eine Paarung in einem bipartiten Graphen, in dem die Männer zu N_1 gehören und die Frauen zu N_2. Hier sind zwei Beispiele für Paarungen, die es für 3 Männer und 3 Frauen gibt.

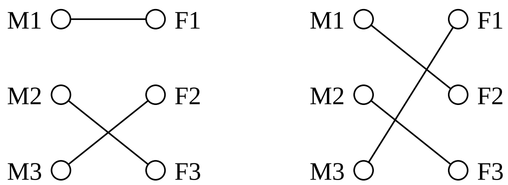


Abbildung E.11: Das Eheproblem für drei Männer und drei Frauen.

Wir beenden dieses Kapitel mit dem zweiten Sudoku-Gitter, das Manori für Lei ergänzen soll. Um sein Ziel zu erreichen, wählt Manori im Gitter sechs bestimmte Kästchen und stellt fest, dass diese nur die Zahlen 1, 2, 3, 4, 6 und 8 enthalten können. Er konstruiert dann einen bipartiten Graphen mit den Kästchen als erste Knotenmenge N_1 und den verfügbaren Zahlen als zweite Knotenmenge N_2. Er verbindet einen Knoten aus N_1 mit einem Knoten aus N_2, wenn die Zahl aus N_2 in dieses Kästchen aus N_1 geschrieben werden kann. Er erhält den in Abbildung E.12 auf der nächsten Seite dargestellten Graphen.

Um diese sechs Kästchen auszufüllen, muss er dann eine Paarung in diesem bipartiten Graphen bestimmen, weil jedes Kästchen nur eine der sechs Zahlen enthalten kann und

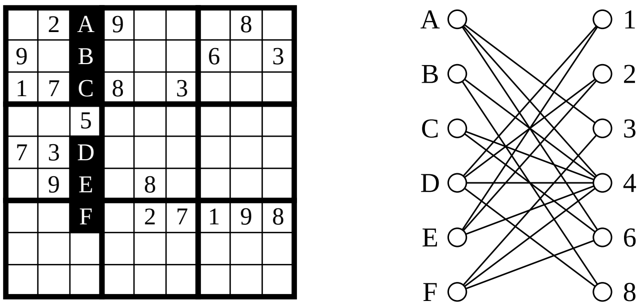


Abbildung E.12: Bipartiter Graph zur Lösung eines Sudoku-Gitters.

jede Zahl nur in eines der sechs Kästchen geschrieben werden kann (weil sie alle in derselben Spalte stehen).

Um einen besseren Blick auf diese Dinge zu haben, entschließt sich Manori, die Knoten ein wenig umzuordnen und sie so zu zeichnen, dass es keine Kantenüberschneidungen mehr gibt. Er erhält die neue nachfolgende Darstellung, in der die Kästchen aus N_1 schwarz gefärbt sind. Die in die Kästchen zu schreibenden Zahlen aus N_2 sind weiß.

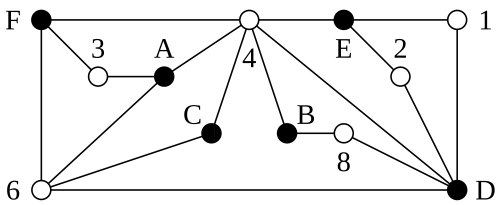


Abbildung E.13: Eine andere Darstellung des bipartiten Graphen aus der letzten Abbildung.

Man sieht gut, dass dieser Graph bipartit ist, weil es keine Kante gibt, die zwei weiße oder zwei schwarze Knoten verbindet. Abgesehen davon, ist dieser Graph auch topologisch planar, weil es keine Kantenüberschneidungen gibt.

Manori stellt dann fest, dass in diesem bipartiten Graphen alle Paarungen zu 6 Kanten zwangsläufig die Kante enthalten, die *B* mit 8 verbindet, was bedeutet, dass man in das Kästchen *B* nur die Zahl 8 schreiben kann.

F Eulerkreise und Hamiltonkreise

Beginnen wir mit der Wiederholung einiger Definitionen, die wir bereits im Kapitel D über Kantengraphen angegeben haben.

Definition F.1. Ein Kreis, der jede Kante eines Graphen genau ein Mal durchläuft, heißt *Eulerkreis*.

Definition F.2. Ein Kreis, der jeden Knoten eines Graphen genau ein Mal besucht, heißt *Hamiltonkreis*.

Manche Graphen enthalten weder Hamilton- noch Eulerkreise, wie dieser Graph.

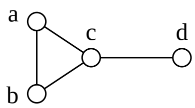


Abbildung F.1: Ein Graph, der weder einen Hamiltonkreis noch einen Eulerkreis besitzt.

Es sei erwähnt, dass man Hamilton- und Euler*wege* in analoger Weise definiert, indem man den Begriff *Kreis* durch *Weg* ersetzt. Der oben abgebildete Graph hat einen Hamiltonweg, nämlich $d - c - a - b$. Er hat auch einen Eulerweg, nämlich $d - c - a - b - c$.

Anwendungen

Ein Handelsreisender fährt jeden Morgen von seinem Wohnsitz los. Dieser Startpunkt ist in Abbildung F.2 durch einen schwarzen Knoten gekennzeichnet. Der Handelsreisende soll sich zu einer Menge von Kunden begeben, die durch weiße Knoten dargestellt sind, und dann zu seinem Wohnsitz zurückkehren. Welchen Weg muss er nehmen,

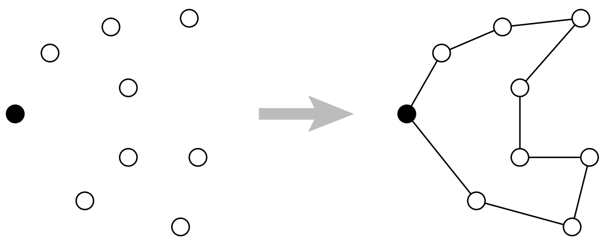


Abbildung F.2: Die Suche nach einem Hamiltonkreis mit minimaler Länge.

damit die zurückgelegte Wegstrecke minimal wird. (Man nimmt an, dass die Entfernungen zwischen allen Kundenpaaren sowie dem Kunden und dem Wohnsitz bekannt sind.) Man sucht hier einen Hamiltonkreis mit minimaler Länge.

Ein Müllauto startet von seinem Depot, in der nachfolgenden Abbildung schwarz gezeichnet, und es soll jede Straße des abgebildeten Netzplans durchfahren, um den Müll zu entsorgen. Welchen Weg soll es nehmen? Hier sucht man einen Eulerkreis.

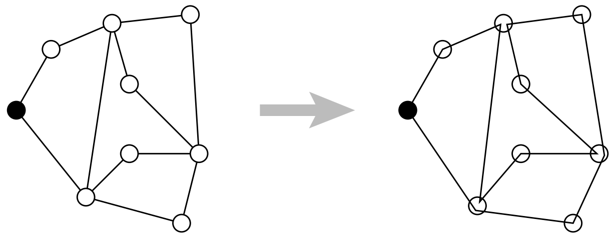


Abbildung F.3: Die Suche nach einem Eulerkreis.

Die Kreise (nicht zwangsläufig Hamiltonkreise oder Eulerkreise) verfügen über eine wichtige Eigenschaft, auf die Manori Sébastien aufmerksam macht, als er versucht, die entwischte Maus wiederzufinden. Diese nachfolgend angegebene Eigenschaft lässt sich leicht auf Wege übertragen.

Satz F.1. Zeichnet man in einem Graphen einen Kreis, so haben alle Knoten zwangsläufig einen geraden Grad, weil man jeden Knoten, auf den man zuläuft, auch wieder verlässt. Zeichnet man in einem Graphen einen Weg, so haben alle Zwischenknoten (also alle Knoten, die nicht Start- oder Endpunkt des Weges sind) einen geraden Grad.

Genau dieser Eigenschaft bedient sich Manori, um die Maus wiederzufinden. Der Weg der Maus von ihrem Käfig bis zu ihrem Versteck ist nämlich ein Kantenzug. Manori zeichnet den Graphen, der alle Kanten enthält, welche die Maus entlanggelaufen ist. Dazu ist er aufgrund der Informationen in der Lage, welche die Bewegungsmelder geliefert haben. Sein Gedankengang ist wie folgt:

- Wenn der Grad aller Knoten gerade ist, so ist die Maus zu ihrem Ausgangsort zurückgekehrt und es wird schwer sein, sie wiederzufinden.
- Wenn der Graph nur zwei Knoten enthält, deren Grad ungerade ist, so ist einer dieser Knoten ihr Fluchtort und der andere der Ort, an dem sie sich versteckt.
- Wenn der Graph mehr als zwei Knoten enthält, deren Grad ungerade ist, so sind die gesammelten Informationen inkohärent.

Glücklicherweise befindet man sich in der zweiten Situation und einer der Knoten mit ungeradem Grad (Knoten A) gehört zu einem Labor, während der andere (Knoten H)

einem Bereich zwischen den Laboren entspricht, der nur eine Abluftöffnung hat. Die Maus befindet sich zwangsläufig in dieser.

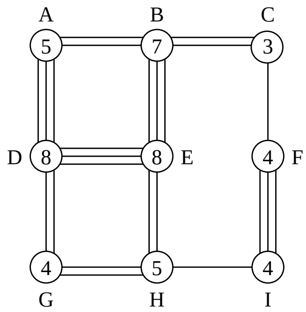


Abbildung F.4: Die von der Maus durchlaufenen Kanten.

Im 18. Jahrhundert hat der große Mathematiker Euler folgendes Problem gelöst. Die Stadt Königsberg (später Kaliningrad) liegt an dem Fluss Pregel, der die Insel Kneiphof zu beiden Seiten umfließt und sieben Brücken besitzt, wie in Abbildung F.5 dargestellt. Kann ein Fußgänger in einem Spaziergang alle Brücken genau ein Mal überqueren?

Abbildung F.5: Der Fluss Pregel mit der Insel Kneiphof und seinen sieben Brücken.

Zur Lösung dieses Problems hat Euler einen Graphen G konstruiert, in dem die Knoten die verschiedenen zusammenhängenden Bereiche sind und in dem jede Kante eine Verbindung zwischen zwei Bereichen durch eine Brücke darstellt. Damit ein Fußgänger einen Spaziergang machen kann, bei dem er jede Brücke genau ein Mal überquert, muss G einen Eulerkreis enthalten, was nicht der Fall ist, weil G vier Knoten enthält, deren Grad ungerade ist.

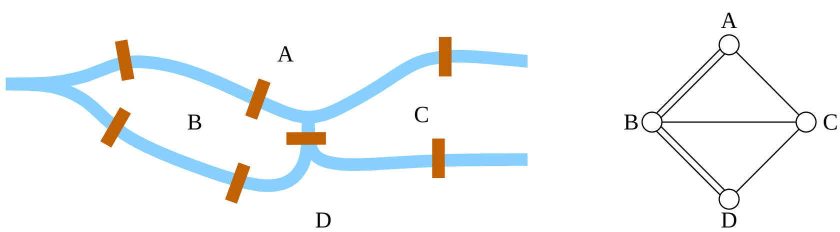


Abbildung F.6: Der zum Königsberger Brückenproblem gehörende Graph.

Wenn alle Knotengrade in einem Graphen gerade sind, ist es ziemlich leicht, einen Eulerkreis zu zeichnen. Dazu kann man wie folgt vorgehen.

Konstruktion eines Eulerkreises in einem Graphen, in dem alle Knoten einen geraden Grad haben.

1. Bestimme einen beliebigen Kreis C.
2. Wenn C alle Kanten durchläuft, dann STOPP. Gehe anderenfalls zu Punkt 3.
3. Wähle einen Knoten v in C, der ein Endpunkt einer Kante außerhalb von C ist. Konstruiere einen zweiten Kreis C' mit v, sodass C und C' keine Kante gemein haben.
4. Verschmelze C und C' zu einem Kreis C''. Diese Verschmelzung ergibt sich, indem man vom Knoten v ausgeht, die Menge der Kanten des Kreises C durchläuft, bis man wieder beim Knoten v angekommen ist, dann die Menge der Kanten des Kreises C' durchläuft, bis man schließlich wieder bei v ankommt.
5. Setze C gleich C'' und gehe zu Punkt 2.

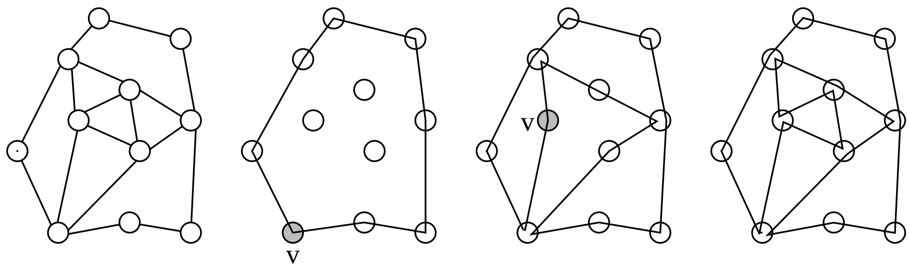


Abbildung F.7: Konstruktion eines Eulerkreises in einem Graphen, in dem alle Knoten einen geraden Grad haben.

Definition F.3. Angenommen, jeder Kante sei eine Entfernung zugeordnet. Das Problem, einen Hamiltonkreis mit minimaler Gesamtlänge zu bestimmen, nennt man *Problem des Handlungsreisenden*.

Das Problem des Handlungsreisenden ist sehr schwer zu lösen. Dennoch existieren Softwarepakete, mit denen man solche optimalen Lösungen bestimmen kann, solange die Anzahl der Knoten ein paar Hundert nicht übersteigt (darunter CONCORDE, das man sich auf der Seite `http://www.tsp.gatech.edu/concorde.html` herunterladen kann). Bis man eine optimale Lösung erhält, können Stunden und sogar Tage vergehen.

Wenn man es mit großen Graphen zu tun hat oder man die Lösungen sehr schnell erhalten möchte, verwendet man sogenannte *Heuristiken*, die in akzeptabler Zeit Lösungen akzeptabler Güte liefern.

Heuristik des nächsten Nachbarn (Nearest-Neighbor-Heuristik)

1. Wähle einen Ausgangsknoten x.
2. Solange nicht alle Knoten bereits besucht wurden, führe folgenden Schritt aus: Gehe zum nächstgelegenen noch nicht besuchten Knoten.

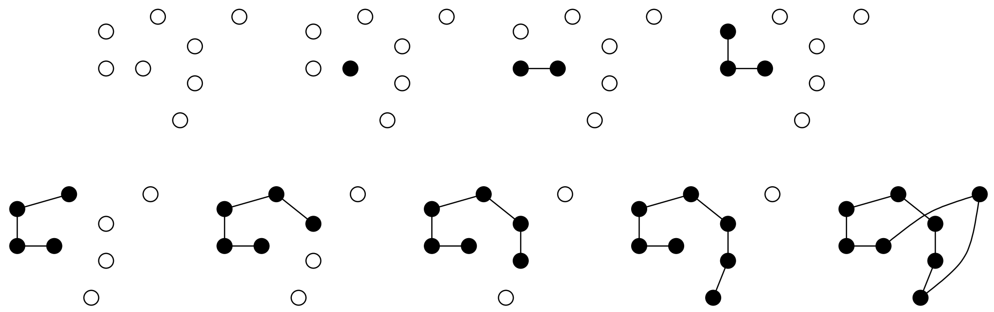


Abbildung F.8: Eine Lösung des Problems des Handlungsreisenden nach der Nearest-Neighbor-Heuristik.

Beispiel F.1. Man hat diese Heuristik mit dem verglichen, was beispielsweise die Software CONCORDE liefert. Der Fehler liegt im Mittel bei ungefähr 15%.

Alle Heuristiken, einschließlich der Nearest-Neighbor-Heuristik, können leicht verbessert werden, indem man am Ende ein Verfahren zur *Nachoptimierung* (Post-Optimization) anschließt. Dieses besteht darin zu prüfen, ob im Kreis ein Kantenpaar existiert, das man durch ein anderes Kantenpaar ersetzen kann (es gibt pro Kantenpaar nur eine mögliche Vertauschung), sodass sich die insgesamt zurückgelegte Wegstrecke verringert. Solange solche Kantenpaare existieren, werden die Vertauschungen ausgeführt. Diese Vorgehensweise illustriert die nachfolgende Abbildung.

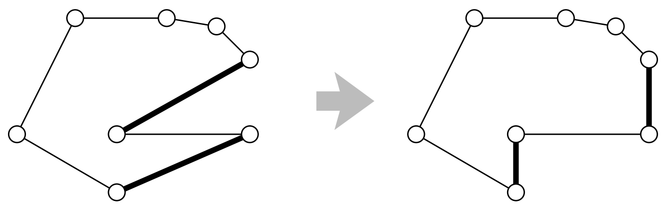


Abbildung F.9: Illustration zum Verfahren der Nachoptimierung.

Dieses Verfahren der Nachoptimierung erlaubt eine beachtliche Reduzierung der Abweichung vom Optimum. Wenn man etwa die Lösungen der Software CONCORDE mit der Kombination aus Nearest-Neighbor-Heuristik und anschließender Nachoptimierung vergleicht, so kann man feststellen, dass der Fehler im Mittel nur noch bei ungefähr 2 bis 3% liegt.

Es gibt Verfahren, mit denen man diese Abweichung auf einige Zehntel Prozent verringern kann, aber diese Verfahren sind viel komplexer einzusetzen.

Kommen wir nun aber auf Eulerkreise zurück. Wie bereits erwähnt, ist es leicht, einen Eulerkreis zu bestimmen, wenn alle Knoten eines Graphen G einen geraden Grad haben, während der Graph keinen Eulerkreis enthält, wenn mindestens ein Knoten einen ungeraden Grad hat. Definitionsgemäß durchläuft man in einem Eulerkreis jede Kante *genau* ein Mal. Wir interessieren uns hier für Probleme, bei denen ein Kreis bestimmt werden soll, der jede Kante des Graphen *mindestens* ein Mal besucht.

Definition F.4. Jeder Kante eines zusammenhängenden Graphen sei eine Entfernung zugeordnet. Das *chinesische Briefträgerproblem* besteht darin, einen kürzesten Kreis zu bestimmen, der jede Kante des Graphen mindestens ein Mal durchläuft.

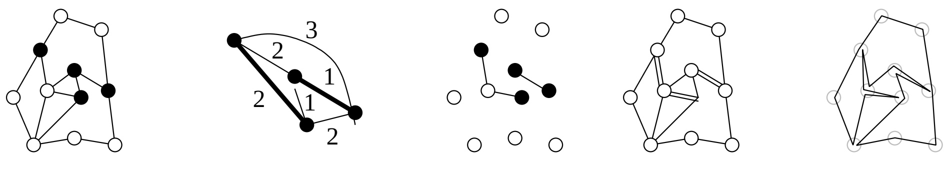
Im Gegensatz zum Problem des Handelsreisenden ist das des chinesischen Briefträgers einfach zu lösen. Wenn kein Knoten mit ungeradem Grad existiert, wissen wir bereits, wie man einen Eulerkreis bestimmt, und dieser Kreis ist zwangsläufig der kürzeste, weil er jede Kante genau ein Mal durchläuft. Anderenfalls kann man wie folgt vorgehen.

Algorithmus zum Finden einer optimalen Lösung des chinesischen Briefträgerproblems in einem Graphen G, der Knoten mit ungeradem Grad enthält.

1. Erzeuge einen vollständigen Graphen H aus den Knoten von G mit ungeradem Grad und verbinde jedes Knotenpaar in H durch eine Kante, deren Kantengewicht die Länge des kürzesten Weges zwischen den betrachteten Knoten in G ist.

2. Bestimme eine kostenminimale Paarung in H.

3. Füge zu jeder Kante der unter 2. bestimmten optimalen Paarung den entsprechenden kürzesten Weg in G ein.



Graph G mit 4 Knoten mit ungeradem Grad (schwarz) | Graph H unter der Annahme, dass alle Entfernungen in G gleich 1 sind; die kostenminimale Paarung ist fett gezeichnet | in G einzufügende Kanten | Graph G mit eingefügten Kanten; alle Knoten haben einen geraden Grad | optimaler Kreis im Briefträgerproblem

Abbildung F.10: Illustration des Lösungsalgorithmus zum Briefträgerproblem.

4. Alle Knoten des resultierenden Graphen haben einen geraden Grad. Man kann daher leicht einen Eulerkreis konstruieren.

Im letzten Beispiel haben wir angenommen, dass alle Entfernungen in G gleich 1 sind. Vergegenwärtigen wir uns dabei: Damit der 2. Schritt des Algorithmus einen Sinn hat und man in der Folge alle Knoten in G mit ungeradem Grad in Knoten mit geradem Grad verwandeln kann, muss es in G eine gerade Anzahl von Knoten mit ungeradem Grad geben. Das ist immer der Fall, wie wir gleich zeigen werden.

Satz F.2. Alle Graphen enthalten eine gerade Anzahl von Knoten mit ungeradem Grad.

Beweis. Man kann eine Zerlegung $V = I \cup P$ der Menge V der Knoten eines Graphen definieren, indem man I als Menge der Knoten mit ungeradem Grad und P als Menge der Knoten mit geradem Grad definiert. Es gilt also

$$\sum_{v \in V} \mathrm{Grad}(v) = \sum_{v \in P} \mathrm{Grad}(v) + \sum_{v \in I} \mathrm{Grad}(v)\,.$$

Wir haben bereits gezeigt, dass diese Summe gerade ist, weil sie gleich dem Doppelten der Anzahl der Kanten des Graphen ist. Da der erste Summand dieser Summe gerade ist (es handelt sich um eine Summe gerader Zahlen), muss auch der zweite Summand dieser Summe gerade sein. Damit eine Summe von ungeraden Zahlen gerade ist, muss man über eine gerade Anzahl von Zahlen summieren, woraus man schließt, dass die Menge I eine gerade Anzahl von Knoten enthält. □

Ein Problem, das dem des chinesischen Briefträgers ähnelt, besteht darin, einen kostenminimalen Kreis zu bestimmen, der eine feste Teilmenge von Kanten mindestens ein Mal durchläuft. Praktisch ist auch das Straßennetz einer Stadt aus zahlreichen Kanten aufgebaut und die Müllabfuhr muss nur bestimmte Kanten des Graphen bedienen. Der mit Müll beladene Laster kann die anderen Kanten des Graphen unterdessen benutzen, um sich von einem Ort zum anderen zu bewegen.

Definition F.5. Jeder Kante eines zusammenhängenden Graphen G sei eine Entfernung zugeordnet und D sei eine Teilmenge zu durchlaufender Kanten in G. Das *Problem des ländlichen Briefträgers* besteht darin, einen kürzesten Kreis in G zu bestimmen, der jede Kante von D mindestens ein Mal durchläuft.

Das Problem des ländlichen Briefträgers ist sehr schwer zu lösen, genau wie das des Handelsreisenden. Tatsächlich sind reale Probleme wesentlich komplexer als die bereits erwähnten, weil dabei zahlreiche Nebenbedingungen berücksichtigt werden müssen, wie etwa die Kapazität der Fahrzeuge, die Pausen der Fahrer usw.

Wenn beispielsweise die bei den Kunden abzuliefernde oder bei ihnen abzuholende Menge die Kapazität eines Fahrzeugs übersteigt, müssen mehrere Lastwagen eingesetzt werden. Das Problem verkompliziert sich demzufolge, weil man nicht nur den Weg jedes Fahrzeugs bestimmen, sondern auch die Kunden unter den Fahrzeugen aufteilen muss, wie in Abbildung F.11 auf der nächsten Seite illustriert.

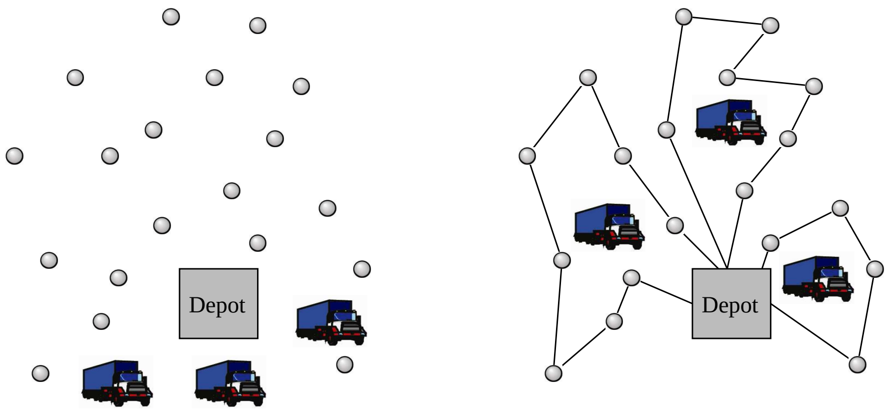


Abbildung F.11: Belieferungsproblem mit realen Nebenbedingungen.

G Färbung von Knoten

Im Jahr 1852 stellte sich der junge Engländer Francis Guthrie die Frage, ob es immer möglich sei, eine geografische Karte mithilfe von vier Farben unter der Bedingung zu färben, dass zwei benachbarte Länder nicht dieselbe Farbe haben. Erst 1976 konnten die beiden amerikanischen Forscher K. Appel und W. Haken an der University of Illinois in Urbana-Champaign bejahend auf Guthries Frage antworten. Mehr als ein Jahrhundert ist also zwischen der Formulierung und der Lösung dieses scheinbar sehr einfachen Problems vergangen. All die Jahre über sind die Forscher natürlich nicht untätig geblieben und es gab zahlreiche Versuche, die Vermutung auf Basis neuer mathematischer Entwicklungen zu beweisen, folglich auch auf Basis einer graphentheoretischen Formulierung des Problems. Die zu färbende Karte wurde durch einen Graphen ersetzt, wobei jedes Land durch einen Knoten dargestellt wird und zwei benachbarte Länder durch eine Kante verbunden werden.

Definition G.1. Eine *Färbung der Knoten* eines Graphen H ist eine Farbzuweisung an Knoten, sodass die Endpunkte jeder Kante von G verschieden gefärbt sind. Im Allgemeinen versucht man eine Färbung zu bestimmen, die so wenige Farben wie möglich verwendet. Die minimale Anzahl von Farben, die man zur Färbung der Knoten eines Graphen G braucht, heißt *chromatische Zahl* oder *Knotenfärbungszahl* von G und wird mit $\chi(G)$ bezeichnet.

Man kann auch die Kanten färben, indem man vermeidet, dass Kanten, die an demselben Knoten liegen, dieselbe Farbe haben. Das haben wir bereits im Kapitel E über Paarung definiert.

Definition G.2. Eine *Kantenfärbung* eines Graphen G ist eine Farbzuweisung an Kanten, sodass Kanten mit einem gemeinsamen Endpunkt verschieden gefärbt sind. Im Allgemeinen versucht man eine Färbung zu bestimmen, die so wenige Farben wie möglich verwendet. Die minimale Anzahl von Farben, die man zur Färbung der Kanten eines Graphen G braucht, heißt *chromatischer Index* von G und wird mit $q(G)$ bezeichnet.

Greifen wir das Beispiel für die Färbung von Kanten aus einem der vorherigen Kapitel wieder auf. Es sollen 5 Tätigkeiten T_1, T_2, T_3, T_4 und T_5 erledigt werden, von denen jede eine Zeitspanne des Tages in Anspruch nimmt. Die Tätigkeiten T_1 und T_2 müssen vom Angestellten E_1 erledigt werden, die Tätigkeiten T_3 und T_4 vom Angestellten E_2 und die Tätigkeit T_5 vom Angestellten E_3. Für die Tätigkeiten T_1 und T_3 wird die Maschine M_1 gebraucht, für die Tätigkeiten T_2 und T_4 wird die Maschine M_2 gebraucht und für die Tätigkeit T_5 die Maschine M_3. Unter Berücksichtigung der Tatsache, dass sich jeder Angestellte nur einer Tätigkeit auf einmal widmen kann und dass jede Maschine nur von einem Angestellten auf einmal verwendet werden kann, wollen wir wissen, wie viele Tage es mindestens dauert, bis die 5 Tätigkeiten erledigt sind.

Dieses Problem ist äquivalent zur Färbung der Kanten des nachfolgenden Graphen mit einer minimalen Anzahl von Farben, sodass sich berührende Kanten verschiedene Farben haben. Jede Farbe entspricht einem Tag; man braucht also 2 Tage. Am 1. Tag kann man zum Beispiel die Tätigkeiten T_1, T_4 und T_5 ausführen und die Tätigkeiten T_2 und T_3 am 2. Tag.

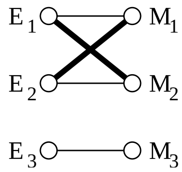


Abbildung G.1: Eine Kantenfärbung, um 5 Tätigkeiten zu erledigen.

Betrachten wir nun dasselbe Problem mit einer zusätzlichen Bedingung. Die Tätigkeiten T_2, T_3 und T_5 beanspruchen denselben ferngesteuerten Roboter, und dieser Roboter kann nicht die Ausführung zweier Tätigkeiten gleichzeitig unterstützen. Die Anzahl der Tage, die mindestens gebraucht werden, um die 5 Tätigkeiten zu erledigen, kann bestimmt werden, indem man die Knoten des nachfolgenden Graphen mithilfe einer minimalen Anzahl von Farben färbt, sodass keine Kante zwei gleichfarbige Endpunkte hat. Wieder entspricht jede Farbe einem Tag; man braucht also 3 Tage. Beispielsweise kann man am 1. Tag die Tätigkeiten T_1, T_4 und T_5 ausführen, die Tätigkeit T_2 am 2. Tag und die Tätigkeit T_3 am 3. Tag.

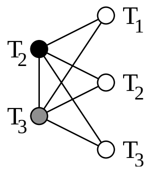


Abbildung G.2: Eine Knotenfärbung, um 5 Tätigkeiten mit Nebenbedingung zu erledigen.

Zahlreiche konkrete Probleme lassen sich tatsächlich auf eine Zerlegung einer Menge von Objekten in Teilmengen zurückführen, die nur paarweise kompatible Elemente enthalten. Beispielsweise ist die Festlegung eines Schulstundenplans eine Zerlegung der Zeit in Zeitabschnitte in denen nur einige der gegebenen Unterrichtsstunden gleichzeitig stattfinden können. Man kann diese Art von Problemen als Färbungsproblem der Knoten eines Graphen modellieren: Die Knoten sind die in Teilmengen zu zerlegenden Objekte und die Kanten stellen die Inkompatibilitäten unter den Objekten dar. Im Beispiel des

Schulstundenplans stehen die Knoten für die in den Stundenplan einzubauenden Unterrichtsstunden und man verbindet zwei Unterrichtsstunden durch eine Kante, wenn diese Unterrichtstunden nicht gleichzeitig gegeben werden können (weil sie denselben Lehrer betreffen, sich an dieselben Schüler richten oder im gleichen Zimmer unterrichtet werden müssen).

Mit dem Problem der Färbung von Kanten haben wir uns bereits in einem der vorherigen Kapitel beschäftigt. Hier interessieren wir uns für die Bestimmung der chromatischen Zahl. Wenn wir den Kantengraphen $L(G)$ eines Graphen G betrachten, so stellen wir unschwer fest, dass die Färbung der Kanten von G äquivalent ist zur Färbung der Knoten von $L(G)$. Der chromatische Index von G ist daher gleich der chromatischen Zahl seines Kantengraphen, was man wie folgt schreibt:

$$q(G) = \chi(L(G)).$$

Definition G.3. Eine *Clique* in einem Graphen ist eine Menge von Knoten, die untereinander alle paarweise durch jeweils eine Kante verbunden sind. Die Größe der größten Clique in einem Graphen G wird mit $\omega(G)$ bezeichnet.

Da die Knoten einer Clique alle verschiedene Farben haben müssen, leitet sich daraus ab, dass die chromatische Zahl $\chi(G)$ von G nicht kleiner als $\omega(G)$ sein kann. Sie kann dagegen echt größer sein, wie es beispielsweise beim Fünfeck der Fall ist. Man braucht nämlich $\chi(G) = 3$ Farben, um seine Knoten zu färben, während $\omega(G) = 2$ ist.

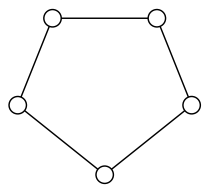


Abbildung G.3: Chromatisches Zahl und größte Clique eines Füngfecks.

Als weiteres Beispiel können wir den Graphen betrachten, den Manori definiert hat, um den Schuldigen für den Raubüberfall in Madame Rossiers Lebensmittelladen zu bestimmen (siehe Abbildung G.4 auf der nächsten Seite). Manori hat gezeigt, dass obwohl dieser Graph keine Clique aus 4 Knoten enthält, seine chromatische Zahl größer als 3 ist. Man kann ihn nämlich nicht mit 3 Farben färben, solange man den Knoten I nicht entfernt.

Zur Erinnerung sei erwähnt, dass Vizing gezeigt hat, dass der chromatische Index immer höchstens gleich dem maximalen Grad $+1$ ist, also $q(G) \leq \Delta(G)+1$. Wir können Folgendes festhalten:

- Eine Menge von Kanten des Graphen G, die alle einen Endpunkt gemeinsam haben, entspricht einer Clique in seinem Kantengraphen $L(G)$; was besagt, dass $\omega(L(G)) \geq \Delta(G)$ ist.

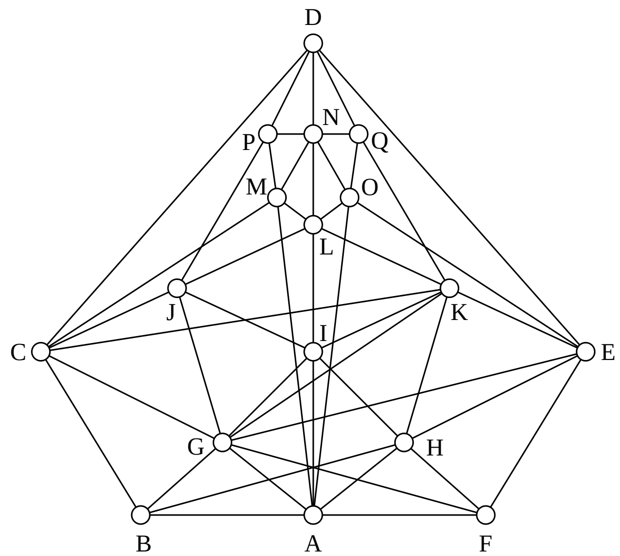


Abbildung G.4: Der Graph für alle Gefangenen aus dem Fall mit dem Kapuzenmann.

- Ist $L(G)$ kein Dreieck, so entspricht eine Clique in $L(G)$ einer Menge von Kanten, die alle an demselben Knoten in G liegen, was besagt, dass $\Delta(G) \geq \omega(L(G))$ ist.

Zusammenfassend stellen wir fest: Ist $L(G)$ kein Dreieck, so gilt $\Delta(G) = \omega(L(G))$. Wenn $L(G)$ ein Dreieck ist, so kann es sein – wie bereits in einem anderen Kapitel festgestellt und nachfolgend erneut dargestellt, dass auch G ein Dreieck ist, in welchem Fall $2 = \Delta(G) < \omega(L(G)) = 3$ ist.

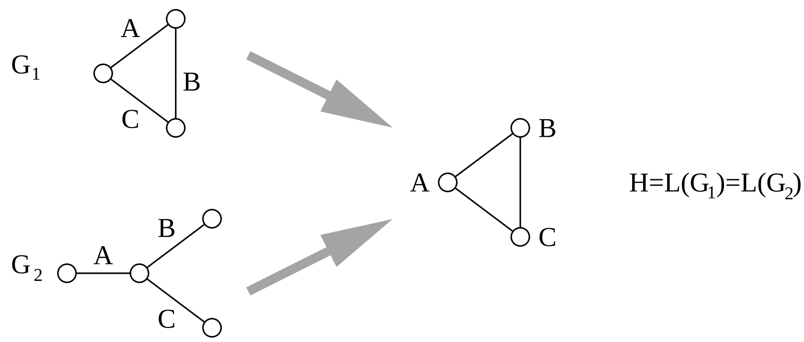


Abbildung G.5: Zwei Graphen G_1 und G_2, deren Kantengraph ein Dreieck ist.

Da $q(G) = \chi(L(G))$ ist, kann das von Vizing bewiesene Resultat wie folgt umgeschrieben werden:

$$\chi(L(G)) \leq \omega(L(G)) + 1\,.$$

Aufgrund der oben genannten Überlegungen gilt dieses Resultat, wenn $L(G)$ kein Dreieck ist. Es gilt ebenfalls, wenn $L(G)$ ein Dreieck ist, weil in diesem Fall $3 = \chi(L(G)) < \omega(L(G)) + 1 = 4$ ist.

Mit anderen Worten: Wir haben gerade gesehen, dass die chromatische Zahl eines Kantengraphen nie wesentlich größer als die Größe seiner größten Clique ist. Für beliebige Graphen gilt dieses Resultat allerdings nicht mehr. Es wurde nämlich bewiesen, dass der Abstand zwischen $\chi(G)$ und $\omega(G)$ auch groß werden kann, wie man sich vorstellen kann. Abbildung G.6 zeigt ein Beispiel, in dem $\chi(G) = \omega(G) + 2$ ist. Dieser Graph wurde wie folgt konstruiert. Man betrachtet zunächst zwei Fünfecke, das eine besteht aus den Knoten a, b, c, d und e und das andere aus den Knoten A, B, C, D und E. Anschließend fügt man zwischen alle Knoten des ersten Fünfecks und alle Knoten des zweiten Fünfecks eine Kante ein.

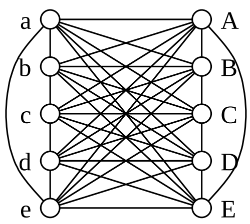


Abbildung G.6: Ein Graph, für den $\chi(G) = \omega(G) + 2$ ist.

Da man zum Färben eines Fünfecks 3 Farben braucht, werden 6 Farben gebraucht, um diesen Graphen zu färben. Außerdem enthält die größte Clique in einem Fünfeck 2 Knoten, was bedeutet, dass die Größe der größten Clique im gesamten Graphen 4 ist. Insgesamt hat man $\chi(G) = 6 = 4 + 2 = \omega(G) + 2$.

Das Problem, die chromatische Zahl eines Graphen G zu bestimmen, ist schwer zu lösen. Mit den meisten heute bekannten exakten Algorithmen kann man die chromatische Zahl für Graphen mit mehr als 100 Knoten nicht mehr bestimmen. Dagegen können diese Probleme für bestimmte Arten von Graphen wesentlich einfacher sein. Für einen bipartiten Graphen, der mindestens eine Kante enthält, ist $\chi(G) = 2$, weil man dem ersten Teil der Zerlegung von V eine Farbe zuweisen kann und dem zweiten Teil die andere Farbe. Wir haben auch gesehen, dass es leicht ist, einen Intervallgraphen zu färben. Bei einem großen beliebigen Graphen hat man keine andere Wahl, als auf Heuristiken zurückzugreifen, die Lösungen akzeptabler Güte in akzeptabler Zeit liefern. Wir stellen hier einige dieser Heuristiken vor.

Die *Konstruktionsverfahren* zur Färbung der Knoten eines Graphen, auch als *sequentielle Färbungsalgorithmen* bezeichnet, durchlaufen die Knoten nacheinander, wobei sie jedem Knoten die kleinstmögliche Farbe zuweisen (die Farben sind nummeriert). Die Güte der auf diese Weise erhaltenen Lösung hängt stark von der Reihenfolge ab, in der man die Knoten betrachtet. Diese Reihenfolge kann von vornherein gewählt sein oder aber dynamisch konstruiert werden. Es ist wichtig, sich zu vergegenwärtigen, dass es für jeden Graphen eine Reihenfolge von Knoten gibt, bei der das Konstruktionsverfahren eine Färbung mit einer minimalen Anzahl von Farben liefert. Um sich davon zu überzeugen, braucht man nur eine Färbung des Graphen G mit $\chi(G)$ Farben zu betrachten und die Knoten von G so zu ordnen, dass man zuerst die Knoten der Farbe 1 nimmt, dann die der Farbe 2 usw.

Satz G.1. Ein Konstruktionsverfahren verwendet nie mehr als $\Delta(G)+1$ Farben.

Beweis. Wenn man einen Knoten färbt, können maximal bereits $\Delta(G)$ Farben unter seinen Nachbarn aufgetreten sein. □

Es ist interessant festzustellen, dass sich ein sequentieller Färbungsalgorithmus schon bei der Färbung des folgenden Graphen täuschen kann, der aus nur 4 Knoten besteht.

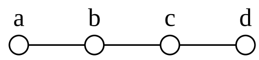


Abbildung G.7: Der Graph P_4.

Wenn man diesen Graphen in der Reihenfolge a, d, b und c färbt, weist man den Knoten a und d die Farbe 1 zu, b die Farbe 2 und c die Farbe 3. Dieser Graph wird als P_4 bezeichnet (*englisch* Path on 4 vertices), denn es handelt sich um eine Kette von 4 Knoten.

Es wurde bewiesen (von Chvátal): Enthält ein Graph kein Quadrupel von Knoten a, b, c und d mit der Struktur von P_4 (das heißt, dass a mit b, nicht aber mit c und d verbunden ist, b außerdem mit c nicht aber mit d verbunden ist und c außerdem mit d verbunden ist), so liefert der sequentielle Algorithmus eine optimale Färbung mit $\chi(G)$ Farben, und das unabhängig von der verwendeten Reihenfolge.

Die verbreitetsten Konstruktionsverfahren zur Färbung werden wir im Folgenden beschreiben. In erster Linie gehören zu den bekanntesten Konstruktionsverfahren, die auf einer statischen Reihenfolge basieren, die folgenden:

RANDOM Die Knoten sind zufällig geordnet.

LF (Largest First) Die Knoten sind in nicht wachsender Reihenfolge nach dem Grad geordnet (siehe Abbildung G.8 auf der nächsten Seite).

SL (Smallest Last) Die Reihenfolge $v_1, v_2, \ldots, v_{|V|}$ der Knoten ist so, dass v_i in dem Graphen, der nur die Knoten $v_1, v_2, \ldots, v_i$ enthält, den kleinsten Grad hat (siehe Abbildung G.9 auf der nächsten Seite).

Ist eine Teilfärbung des Graphen G gegeben, so definieren wir den Sättigungsgrad $\mathrm{DSAT}(v)$ eines Knotens v in G als die Anzahl der verschiedenen Farben, die bereits an

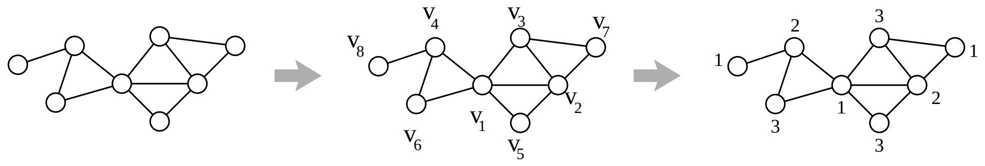


Abbildung G.8: Färbung mithilfe des Algorithmus LF.

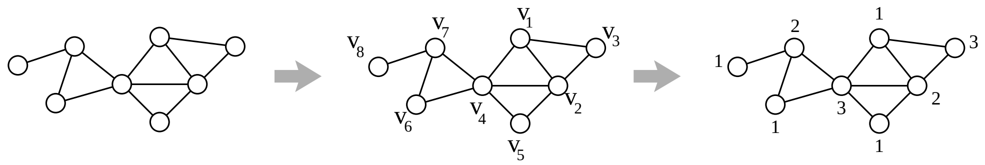


Abbildung G.9: Färbung mithilfe des Algorithmus SL.

Nachbarknoten von v vergeben wurden. Ist eine Teilmenge W von Knoten in G gegeben, so gibt $\text{GRAD}_w(v)$ die Anzahl der Knoten w in G an, die Nachbarn von v sind. Im Folgenden entspricht die Teilmenge A von Knoten der Menge von Knoten, die noch nicht gefärbt sind. Unter den Konstruktionsverfahren, die auf dynamischen Reihenfolgen basieren, wird das Folgende am häufigsten verwendet:

DSATUR Die Reihenfolge der Knoten wird konstruiert, indem man in jedem Schritt den Knoten v aus A mit maximalem $\text{DSAT}(v)$ wählt. Haben zwei Knoten v und w denselben Sättigungsgrad, so kommt v vor w, sofern $\text{GRAD}_A(v) \geq \text{GRAD}_A(w)$ ist.

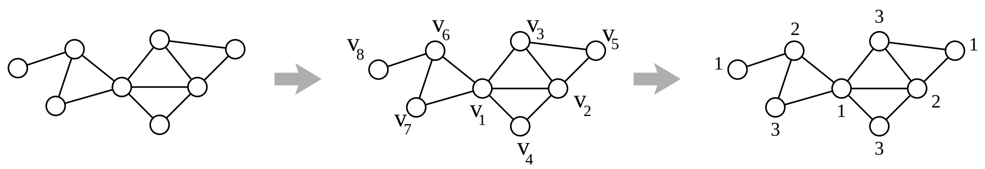


Abbildung G.10: Färbung mithilfe des Algorithmus DSATUR.

Tabelle G.1 auf der nächsten Seite gibt einige Aufschlüsse über die Performanz der vier hier vorgestellten Algorithmen auf zufällig erzeugten Graphen. Wir bezeichnen mit $G_{n,d}$ einen Zufallsgraphen aus n Knoten, in dem jede Kante unabhängig von anderen Kanten mit einer Wahrscheinlichkeit d existiert. Die Wahrscheinlichkeit d heißt *Dichte* des Graphen. Alle Ergebnisse aus Tabelle G.1 sind über mindestens zehn Zufallsgraphen gemittelt. Für jeden Algorithmus geben wir die mittlere Anzahl der Farben an, die zum Färben des betrachteten Graphen $G_{n,d}$ verwendet wurden.

Für Zufallsgraphen von bis zu 1000 Knoten haben alle Algorithmen Rechenzeiten in der Größenordnung von einer Sekunde. Bedenken Sie, dass es wesentlich effizientere

Tabelle G.1: Performanz der Algorithmen zur Färbung von Knoten.

n	**d**	**RANDOM**	**LF**	**SL**	**DSATUR**
100	0.4	17.60	16.10	16.65	14.70
	0.5	21.35	20.15	20.50	18.45
	0.6	26.20	24.50	24.85	22.60
300	0.4	39.00	37.10	38.10	34.30
	0.5	48.20	46.00	46.80	43.30
	0.6	61.00	58.20	58.90	53.70
500	0.4	57.20	56.00	56.80	51.00
	0.5	72.40	69.20	71.80	65.00
	0.6	91.40	87.60	89.60	82.80
1000	0.4	99.50	96.50	97.00	92.00
	0.5	126.50	122.00	124.50	116.00
	0.6	160.50	155.00	157.00	146.50

Algorithmen gibt, die aber schwerer zu implementieren sind. Für den Zufallsgraphen mit 1000 Knoten und einer Dichte von 0.5 liegen die Resultate aus Tabelle G.1 beispielsweise zwischen 116 und 127 Farben, während es der gegenwärtig beste bekannte Algorithmus schafft, Färbungen derselben Graphen mit nur 82 Farben zu bestimmen (wenn auch mit einigen Stunden Rechenzeit).

Wir beenden dieses Kapitel, indem wir noch einmal auf Manori zu sprechen kommen, der die Färbung von Knoten eines Graphen verwendet hat, um die Lösung zum ersten Sudoku-Gitter zu finden, das ihm von der jungen Lei vorgelegt wird. Tatsächlich lässt sich jedes Sudoku-Problem auf die Aufgabe zurückführen, einen Graphen aus 81 Knoten mit 9 Farben zu färben, wobei einigen Knoten bereits eine Farbe zugeordnet wurde. Dieser Graph enthält einen Knoten pro Kästchen, und zwei Knoten sind durch

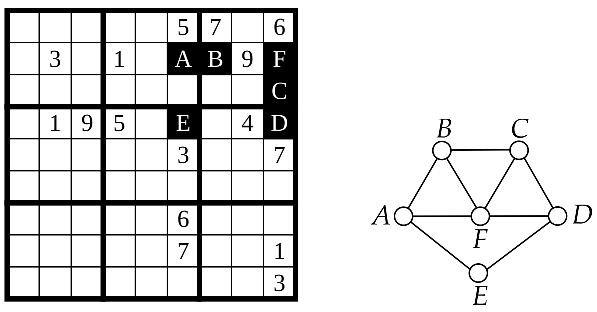


Abbildung G.11: Der Graph zu speziellen Kästchen eines Sudoku-Gitters.

eine Kante miteinander verbunden, wenn sich die zugehörigen Kästchen in derselben Zeile, derselben Spalte oder demselben 3×3-Quadrat befinden.

Im von Lei vorgeschlagenen Gitter betrachtet Manori nur 6 spezielle Kästchen, wie in Abbildung G.11 auf der vorherigen Seite dargestellt.

Manori stellt fest, dass diese 6 Kästchen nur die Zahlen 2, 4 und 8 enthalten können. Man muss diesen Graphen daher mit 3 Farben färben, von denen eine Farbe für die 2, eine andere für die 4 und die letzte für die 8 steht. Man stellt sofort fest, dass die einzige mögliche Lösung den Kästchen A und C eine erste Farbe zuordnet, eine zweite Farbe den Kästchen B und D und schließlich den Kästchen E und F eine dritte Farbe. Es reicht dann festzustellen, dass die Kästchen D und E die Zahl 4 nicht enthalten können, um zu schlussfolgern, dass die Farbe zur Zahl 4 nur den Kästchen A und C zugeordnet werden kann.

H Kürzeste und längste Wege

Sei $I = \{1, 2, \dots, n\}$ eine Menge zu disponierender Aufgaben. Die Ausführungsdauer jeder Aufgabe i ist bekannt und gleich p_i. Per Definition kann eine Aufgabe, die gerade ausgeführt wird, nicht unterbrochen werden. Die Ausführung der Aufgaben ist an die Bedingungen geknüpft, dass bestimmte Aufgaben nicht begonnen werden können, bevor andere beendet sind oder einen gewissen Fortschritt erreicht haben. Andere Bedingungen verlangen, dass bestimmte Aufgaben nicht zu lange nach dem Beginn anderer Aufgaben begonnen werden dürfen. Das ist beispielsweise im Zuge chemischer Prozesse der Fall, wo eine zu lange Wartezeit zwischen zwei Aufgaben die gewünschte chemische Reaktion abbrechen kann. Bezeichnen wir mit t_i den Zeitpunkt, an dem die Aufgabe i beginnt. Alle Bedingungen können in folgende Form gebracht werden:

$$t_j - t_i \geq a_{ij} \, ,$$

wobei a_{ij} eine gegebene reelle Zahl ist, die einer Dauer entspricht. Hier sind einige Beispiele:

- Die Aufgabe j kann nicht beginnen, bevor die Aufgabe i beendet ist: $t_j - t_i \geq p_i$.
- Die Aufgabe j beginnt spätestens, nachdem eine Zeitspanne T seit dem Beginn der Aufgabe i verstrichen ist: $t_i + T \geq t_j$, was äquivalent zu $t_i - t_j \geq -T$ ist.
- Die Aufgaben i und j sollen gleichzeitig beginnen: $t_i - t_j \geq 0$ und $t_j - t_i \geq 0$.
- Die Aufgabe j soll unmittelbar auf die Aufgabe i folgen, ohne dass es zu einer Unterbrechung kommt: $t_i - t_j \geq -p_i$ und $t_j - t_i \geq p_i$.

Das Problem, für das wir uns hier interessieren, besteht darin, einen Ablaufplan mit minimaler Dauer zu finden. Es handelt sich mit anderen Worten um eine Ablaufplanung der Aufgaben, sodass der Zeitpunkt, an dem die Ausführung der letzten Aufgabe abgeschlossen ist, so früh wie möglich liegt.

Graphentheoretische Formulierung

Man kann diesem Problem einen Graphen zuordnen. Dazu betrachtet man zusätzlich zwei fiktive Aufgaben 0 und $n+1$.

- Jeder Aufgabe $i \in \{0, 1, 2, \dots, n, n+1\}$ entspricht ein Knoten des Graphen.

- Die Knoten i und j, die durch die Relation $t_j - t_i \geq a_{ij}$ verknüpft sind, sind durch eine gerichtete Kante von i nach j der Länge a_{ij} verbunden.

- Jeder Knoten $i \in \{1, 2, \ldots, n\}$ ist durch eine gerichtete Kante der Länge p_i mit dem Knoten $n+1$ verbunden.

- Der Knoten 0 ist mit jedem Knoten $i \in \{1, 2, \ldots, n\}$ durch eine gerichtete Kante der Länge 0 verbunden.

Eine Ablaufplanung mit minimaler Dauer zu finden, läuft dann darauf hinaus, den längsten Weg zwischen den Knoten 0 und $n+1$ zu bestimmen. Ein solcher Weg heißt *kritischer Weg*, weil die Aufgaben auf diesem Weg festgelegte Startzeitpunkte haben, wenn man die Menge der Aufgaben frühestmöglich abschließen möchte. Die auf einem kritischen Weg liegenden Aufgaben heißen *kritische Aufgaben*. Wenn man die kritischen Wege bestimmt hat, kann man seine Aufmerksamkeit auf die Ausführung der kritischen Aufgaben richten: Jede geplante Verspätung bei der Ausführung einer kritischen Aufgabe erhöht unweigerlich die Dauer des Projekts.

Als Manori versucht, die Weckzeit der Familie Courtel hinauszuschieben, konstruiert er einen solchen Graphen. Aus der Liste der von Sébastien angegebenen Aufgaben kann man den in Abbildung H.2 auf der nächsten Seite dargestellten Graphen konstruieren.

Aufgaben	Beschreibungen	Dauer	Vorgänger
V1	Sich Waschen	10	A1, M1
V2	Schränke im Zimmer der Eltern ausräumen	15	V1
V3	Sachen der Eltern packen	20	V2
V4	Gepäck zur Rezeption bringen	10	V3, M2
V5	Schlüssel abgeben	5	A3
V6	Leihwagen entgegennehmen	0	V5
M1	Sich waschen	20	A1
M2	Sachen der Töchter packen	20	M1, T1
M3	Sich schminken und das Zimmer der Eltern abgehen	15	V3, M2
T1	Schränke im Zimmer der Töchter ausräumen	15	A1
T2	Sich waschen	20	T1, V1, M1
T3	Zimmer der Töchter abgehen	5	T2, M2
A1	Aufstehen	0	
A2	Frühstücken	45	V4, M3, T3
A3	Abschließendes Zähneputzen	10	A2

Abbildung H.1: Aufgaben der einzelnen Familienmitglieder nach dem Aufstehen.

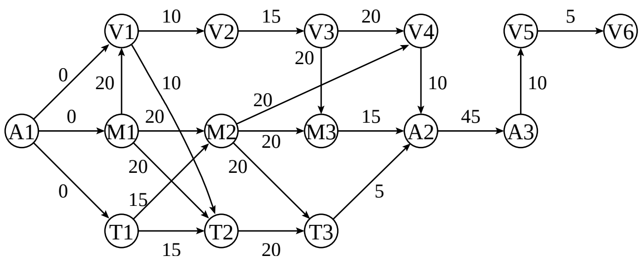


Abbildung H.2: Der Graph zur vorherigen Tabelle.

Hier entspricht die Aufgabe 0 der Aufgabe $A1$, und die Aufgabe $n+1$ ist die Aufgabe $V6$. Es gibt zum Beispiel die Bedingung, dass die Aufgabe $M1$ vor der Aufgabe $V1$ erledigt sein muss, was $t_{V1} \geq t_{M1} + 20$ bedeutet. Diese Ungleichung kann man in die Standardform $t_{V1} - t_{M1} \geq 20$ bringen, was die gerichtete Kante von $M1$ nach $V1$ mit der Länge 20 erklärt.

In diesem Graphen ist $A1$ nur mit $V1$, $M1$ und $T1$ direkt verknüpft. Wie von Sébastien angesprochen, müsste man theoretisch auch eine gerichtete Kante von $A1$ zu den anderen Knoten einfügen. Doch diese gerichteten Kanten sind eigentlich unnötig, da jeder Weg, der von $A1$ zu diesen anderen Knoten führt, $V1$, $M1$ oder $T1$ durchlaufen muss.

Der längste Weg von $A1$ nach $V6$ hat in diesem Graphen die Länge 140, was bedeutet, dass Familie Courtel 2 Stunden und 20 Minuten braucht, um sich für die Abreise am Vormittag vorzubereiten.

Während es in dem Problem, das wir eben untersucht haben, um die Suche nach dem längsten Weg in einem Graphen geht, bestehen andere Probleme darin, den kürzesten Weg zu finden. Will man sich beispielsweise über ein Straßennetz von einem Ort zum anderen begeben, kann man den kürzesten oder den schnellsten Weg suchen. Genau das machen alle GPS-Systeme. Will man die zurückgelegte Wegstrecke minimieren, sind die Werte an den gerichteten Kanten die von einem Ort zum anderen zurückzulegenden Entfernungen. Will man dagegen erreichen, dass die Route auch so schnell wie möglich ist, geben die an den gerichteten Kanten stehenden Werte die Dauer der Teilrouten an.

Auch wenn man nach der besten Strategie sucht, um von einem Zustand A zu einem Zustand B zu kommen, kann man die Zwischenschritte betrachten, die man durchlaufen muss. Man kann auch die aufzubringenden Kosten messen, um von einem Schritt zum nächsten zu kommen. Die beste Strategie besteht dann darin, den kürzesten Weg von A nach B zu nehmen. Genau das tut Manori, als er das Problem der Flucht der 4 Gefangenen löst. Dazu hat er den Graphen aus Abbildung H.3 auf der nächsten Seite gezeichnet.

Das Problem besteht darin, sich vom obersten Knoten des Graphen, der für die Situation steht, in der sich alle Gefangenen in ihrer Zelle befinden, zum untersten Knoten zu bewegen, der für die Situation steht, in der die Gefangenen frei sind. Der kürzeste Weg liefert die Ausbruchstrategie mit der kürzesten Dauer.

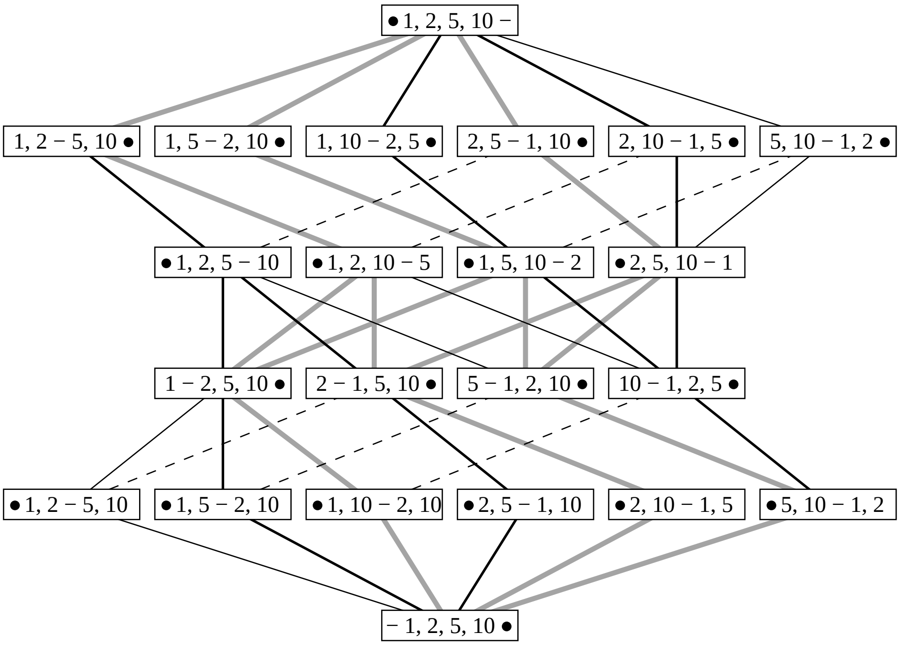


Abbildung H.3: Der gerichtete Graph zum Gefangenenproblem aus Kapitel 8.

Wir werden nun schildern, wie man einen kürzesten Weg in einem Graphen bestimmt. Wenn wir einen längsten Weg bestimmen müssen, braucht man übrigens nur das Vorzeichen der Kantengewichte zu vertauschen (also die Länge d_{ij} durch $-d_{ij}$ zu ersetzen) und nach dem kürzesten Weg zu suchen. Zum Beispiel ist der längste Weg von A nach B in dem linken Graphen aus Abbildung H.4 der untere Weg mit einer Länge von -1, und der kürzeste im rechten Graphen ist ebenso der untere Weg mit einer Länge von 1.

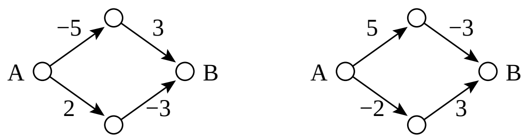


Abbildung H.4: Zwei Graphen, in denen die Vorzeichen der Kantengewichte vertauscht wurden.

Manche mögen überrascht sein, dass man negative Kantengewichte betrachtet, da die Entfernungen in einem Straßennetz und auch alle Wegzeiten immer positiv sind. Wir haben aber gesehen, dass die Suche nach dem kürzesten oder dem längsten Weg in anderen Zusammenhängen vorkommen kann. Wenn man etwa fordert, dass die Aufgabe A 20 Minuten nach der Aufgabe B beginnt, lautet die Bedingung $t_A \leq t_B + 20$, was man auch als $t_B - t_A \geq -20$ schreiben kann. In dem zugehörigen Graphen treffen wir dann auf eine gerichtete Kante von A nach B mit dem negativen Gewicht -20.

Wir fangen mit der Vorstellung eines Algorithmus an, mit dem nach dem kürzesten Weg zwischen einem Knoten A und allen anderen Knoten des Graphen gesucht wird, falls der betrachtete Graph keinen gerichteten Kreis (Zyklus) enthält. Dieser Algorithmus markiert einen Knoten, sobald die Länge des kürzesten Weges zu ihm bekannt ist. Bezeichnen wir mit $PCC(v)$ die Länge des kürzesten Weges von A zum Knoten v, und sei d_{xy} die Länge der gerichteten Kante von x nach y.

Algorithmus zur Suche nach einem kürzesten Weg zwischen einem Knoten A und allen anderen Knoten des Graphen für den Fall, dass der betrachtete Graph keinen Zyklus enthält.

1. Setze $PCC(A) = 0$ und markiere A.
2. Wenn mindestens ein unmarkierter Knoten existiert, gehe zu 3., anderenfalls STOPP.
3. Wähle einen unmarkierten Knoten v, dessen Vorgänger alle markiert sind. Bestimme den minimalen Wert von $PCC(w) + d_{wv}$, indem du alle Vorgänger w von v betrachtest, und setze $PCC(v)$ gleich diesem Minimum; markiere den Knoten v und gehe wieder zu 2.

In dem kleinen nachfolgenden Beispielgraph kann man, nachdem man $PCC(A) = 0$ gesetzt und A markiert hat, v gleich C oder D wählen. Wählen wir $v = C$. Man setzt folglich $PCC(C) = PCC(A) + 5 = 5$ und markiert C. Der einzige Knoten, dessen Vorgänger alle markiert sind, ist dann der Knoten D. Man setzt daher $PCC(D) = PCC(A) - 2 = -2$ und markiert D. Schließlich ist der letzte zu betrachtende Knoten der Knoten B. Man setzt $PCC(B) = \min\{PCC(C) - 3, PCC(D) + 3\} = \min\{2, 1\} = 1$. Der Algorithmus stoppt, nachdem er B markiert hat.

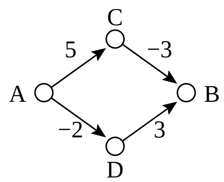


Abbildung H.5: Azyklischer Beispielgraph zur Suche nach dem kürzesten gerichteten Weg zwischen einem Knoten A und allen anderen Knoten des Graphen.

Wenn der betrachtete Graph Zyklen enthält, kann der oben angegebene Algorithmus nicht angewendet werden, weil nicht mehr zwangsläufig ein Knoten existiert, dessen Vorgänger alle markiert sind. Bevor der erste Knoten in einem Zyklus markiert wird, hat nämlich jeder Knoten des Zyklus einen nicht markierten Vorgänger.

Der zweite Algorithmus, den wir vorstellen, ist nur für den Fall gültig, dass alle Gewichte (Entfernungen) positiv sind. Dieser Algorithmus markiert ebenfalls die Knoten nacheinander. Er betrachtet für den kürzesten Weg von A nach v temporäre Werte

$pcc(v)$. Diese temporären Werte werden aktualisiert, um schließlich einmal in den echten Wert $PCC(v)$ zu münden, wenn der Knoten v markiert ist. Der Algorithmus kann wie folgt beschrieben werden.

Algorithmus zur Suche nach einem kürzesten Weg zwischen einem Knoten A und allen anderen Knoten des Graphen für den Fall, dass alle Kantengewichte positiv sind.

1. Setze $pcc(A) = 0$ und $pcc(v) = \infty$ (unendlich) für alle $v \neq A$.
2. Wenn mindestens ein unmarkierter Knoten existiert, gehe zu 3., anderenfalls STOPP.
3. Wähle einen unmarkierten Knoten v mit dem minimalen Wert $pcc(v)$ und setze $PCC(v) = pcc(v)$. Markiere v.
4. Setze für alle unmarkierten Nachfolger w von v $pcc(w) = \min\{pcc(w), PCC(v) + d_{vw}\}$ (wobei v der zuletzt im 3. Schritt markierte Knoten ist) und gehe wieder zu 2.

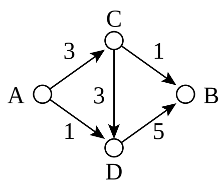


Abbildung H.6: Beispielgraph mit positiven Kantengewichten zur Suche nach dem kürzesten gerichteten Weg zwischen einem Knoten A und allen anderen Knoten des Graphen.

Im Beispielgraphen aus Abbildung H.6 wählt man, nachdem man $pcc(A) = 0$ und $pcc(B) = pcc(C) = pcc(D) = \infty$ gesetzt hat, $v = A$ und setzt $PCC(A) = 0$, wobei man A markiert. Dann nimmt man folgende Aktualisierungen vor:

- $pcc(C) = \min\{\infty, PCC(A) + 3\} = 3$
- $pcc(D) = \min\{\infty, PCC(A) + 1\} = 1$

Man wählt folglich $v = D$ als nächsten Knoten und setzt $PCC(D) = 1$, wobei man D markiert. Die einzige neue Aktualisierung ist $pcc(B) = \min\{\infty, PCC(D) + 5\} = 6$. Anschließend wählt man $v = C$ und setzt $PCC(C) = 3$, wobei man C markiert, und man aktualisiert $pcc(B) = \min\{6, PCC(C) + 1\} = 4$. Schließlich wählt man $v = B$, setzt $PCC(B) = 4$, wobei man B markiert, und der Algorithmus hält an.

Vergegenwärtigen wir uns, dass dieser Algorithmus nicht zwangsläufig einen kürzesten Weg von A zu anderen Knoten liefert, wenn einige Kantengewichte des Graphen negativ sind. Modifiziert man beispielsweise das Gewicht der gerichteten Kante von C nach D, sodass es nicht mehr 3 sondern -3 ist, wird der vorgeschlagene Algorithmus wie oben beschrieben starten und folglich $PCC(D) = 1$ setzen, obwohl der kürzeste Weg von A nach D die Länge 0 hat (indem man über C läuft).

Schließlich geben wir an, wie man den kürzesten gerichteten Weg in einem Graphen findet, der Zyklen und gerichtete Kanten mit negativen Kantengewichten enthält. Wir nehmen an, dass kein Zyklus mit negativer Länge existiert, da das Problem anderenfalls keine Lösung mit endlichem Wert hätte. Man müsste sich nämlich nur von A in den Zyklus begeben und ihn unbegrenzt lange durchlaufen, bevor man ihn in Richtung eines anderen Knotens verlässt, um einen Weg der Länge $-\infty$ zu erhalten. Der nachfolgende Algorithmus betrachtet nur Werte $PCC(v)$ für jeden Knoten v; diese Werte werden während des gesamten Algorithmus aktualisiert, und wir wissen erst ganz am Ende des Algorithmus, ob sie Längen der kürzesten Wege von A zu den anderen Knoten entsprechen.

Algorithmus zur Suche nach einem kürzesten Weg zwischen einem Knoten A und allen anderen Knoten des Graphen im allgemeinen Fall (jedoch unter der Annahme, dass kein Zyklus mit negativer Länge darin existiert).

1. Setze $PCC(A) = 0$ und $PCC(v) = \infty$ (unendlich) für alle $v \neq A$.

2. Bestimme für jeden Knoten v das Minimum der Werte $PCC(w) + d_{wv}$, indem du alle Vorgänger w von v betrachtest, und setze $f(v)$ gleich diesem Minimum.

3. Nimm für jeden Knoten v die Aktualisierung $PCC(v) = \min\{PCC(v), f(v)\}$ vor.

4. Wenn im 3. Schritt mindestens für einen Knoten v der Wert $PCC(v)$ modifiziert wurde, gehe wieder zu 2., anderenfalls STOPP.

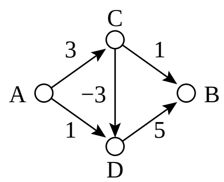


Abbildung H.7: Beispielgraph mit beliebigen Kantengewichten zur Suche nach dem kürzesten gerichteten Weg zwischen einem Knoten A und allen anderen Knoten des Graphen.

Für den Graphen aus Abbildung H.7 setzt man $PCC(a) = 0$, $PCC(B) = PCC(C) = PCC(D) = \infty$ und berechnet:

- $f(B) = PCC(D) + 5 = \infty + 5 = \infty$
- $f(C) = \min\{PCC(A) + 3, PCC(B) + 1\} = \min\{0 + 3, \infty + 1\} = 3$
- $f(D) = \min\{PCC(A) + 1, PCC(C) - 3\} = \min\{0 + 1, \infty - 3\} = 1$

Man muss daher $PCC(C)$ aktualisieren, was den Wert 3 annimmt, sowie $PCC(D)$, was den Wert 1 annimmt. Da im 3. Schritt zwei Werte modifiziert wurden, berechnet man die Werte $f(v)$ neu:

- $f(B) = PCC(D) + 5 = 1 + 5 = 6$
- $f(C) = \min\{PCC(A) + 3, PCC(B) + 1\} = \min\{0 + 3, \infty + 1\} = 3$
- $f(D) = \min\{PCC(A) + 1, PCC(C) - 3\} = \min\{0 + 1, 3 - 3\} = 0$

Man muss daher $PCC(B)$ aktualisieren, was den Wert 6 annimmt, und $PCC(D)$, was den Wert 0 annimmt. Da im 3. Schritt zwei Werte modifiziert wurden, berechnet man die Werte $f(v)$ neu:

- $f(B) = PCC(D) + 5 = 0 + 5 = 5$
- $f(C) = \min\{PCC(A) + 3, PCC(B) + 1\} = \min\{0 + 3, 6 + 1\} = 3$
- $f(D) = \min\{PCC(A) + 1, PCC(C) - 3\} = \min\{0 + 1, 3 - 3\} = 0$

Man muss daher $PCC(B)$ aktualisieren, was den Wert 5 annimmt. Da im 3. Schritt ein Wert modifiziert wurde, berechnet man die Werte $f(v)$ neu:

- $f(B) = PCC(D) + 5 = 0 + 5 = 5$
- $f(C) = \min\{PCC(A) + 3, PCC(B) + 1\} = \min\{0 + 3, 5 + 1\} = 3$
- $f(D) = \min\{PCC(A) + 1, PCC(C) - 3\} = \min\{0 + 1, 3 - 3\} = 0$

Da keiner der Werte $f(v)$ kleiner als $PCC(v)$ ist, hält der Algorithmus an.

In den Algorithmen, die wir beschrieben haben, ist es leicht, den kürzesten Weg von A zu allen anderen Knoten des Graphen zu verfolgen. Dazu muss man nur speichern, von wo die Kante ausgeht, die auf v gerichtet ist, wenn man $PCC(v)$ (oder $pcc(v)$ im vorherigen Algorithmus) modifiziert. In unserem Beispiel ging die letzte Aktualisierung von $PCC(B)$ von D aus, die von $PCC(C)$ ging von A aus und die von $PCC(D)$ von C. Die gerichteten Kanten für den kürzesten Weg sind daher die aus Abbildung H.8.

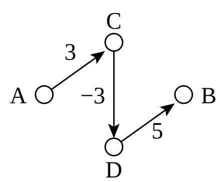


Abbildung H.8: Der kürzeste Weg von A nach B.

Sachverzeichnis

http://www.springer.com/978-3-8348-1814-0

Der Graf der Graphen
Kriminalistische Verwicklungen mit mathematischer Pointe
Hertz, A.
2012, X, 192 S. 96 Abb., Softcover
ISBN: 978-3-8348-1814-0