

Univ.-Prof. Dr. Klaus Neumann
Univ.-Prof. Dr. Dietrich Ohse
Univ.-Prof. Dr. Wilhelm Rödder
Dr. Friedhelm Kulmann
Dr. Heinz Peter Reidmacher

Modul 31801 Problemlösen in graphischen Strukturen

Kurs 00852
Optimierung in Graphen
Kurs 00857
Optimierung mit intelligenten Strategien

LESEPROBE

Fakultät für
**Wirtschafts-
wissenschaft**

Der Inhalt dieses Dokumentes darf ohne vorherige schriftliche Erlaubnis durch die FernUniversität in Hagen nicht (ganz oder teilweise) reproduziert, benutzt oder veröffentlicht werden. Das Copyright gilt für alle Formen der Speicherung und Reproduktion, in denen die vorliegenden Informationen eingeflossen sind, einschließlich und zwar ohne Begrenzung Magnetspeicher, Computerausdrucke und visuelle Anzeigen. Alle in diesem Dokument genannten Gebrauchsnamen, Handelsnamen und Warenbezeichnungen sind zumeist eingetragene Warenzeichen und urheberrechtlich geschützt. Warenzeichen, Patente oder Copyrights gelten gleich ohne ausdrückliche Nennung. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Gliederung

Kurs 00852 Optimierung in Graphen

Kurseinheit 1 »Grundlagen der Graphentheorie«

1. Grundbegriffe

- 1.1. Praktische Probleme, die auf Graphen und Netzwerke führen
- 1.2. Grundlegende Definitionen
- 1.3. Kantenfolgen in Graphen und Pfeilfolgen in Digraphen
- 1.4. Erreichbarkeit und Zusammenhang in Graphen und Digraphen
- 1.5. Graphen, Digraphen und Matrizen
- 1.6. Für Anwendungen wichtige Klassen von Graphen und Digraphen
- 1.7. Bewertete Graphen und Digraphen, Netzwerke

2. Graphen und Computer

- 2.1. Rechenaufwand von Algorithmen
- 2.2. Einige wichtige Graphen-Algorithmen

3. Minimalgerüste und kürzeste Wege

- 3.1. Minimalgerüste
- 3.2. Kürzeste Wege von einem zu allen Knoten (Baumalgorithmen)
- 3.3. Bellmans Verfahren für zyklenfreie Netzwerke
- 3.4. Dijkstras Algorithmus für Netzwerke mit nichtnegativer Bewertung
- 3.5. Das Verfahren von Ford für Netzwerke mit beliebiger Bewertung
- 3.6. Kürzeste Wege zwischen allen Knoten:
Der Tripel-Algorithmus von Floyd und Warshall

4. Flüsse in Netzwerken

- 4.1. Flüsse und Schnitte in Netzwerken
- 4.2. Flüsse in Graphen und Zirkulationsflüsse
- 4.3. Der Algorithmus von Ford und Fulkerson
zur Bestimmung eines maximalen Flusses
- 4.4. Bestimmung eines zulässigen Anfangsflusses
- 4.5. Kostenminimale Flüsse und Zirkulationsflüsse
- 4.6. Optimalitätsbedingungen für Zirkulationsflüsse
- 4.7. Der Out-of-Kilter-Algorithmus
zur Bestimmung eines kostenminimalen Zirkulationsflusses

Lösungen zu den Übungsaufgaben

Kurseinheit 2 »Standortplanung und Transportoptimierung«

5. Optimale Standortplanung

- 5.1. Standardfragen der Standortplanung
- 5.2. Mediane und Zentren
- 5.3. Überdeckungsprobleme
- 5.4. Warehouse Location Probleme

6. Modellierung von Transportproblemen

- 6.1. Verallgemeinerungen des klassischen Transportproblems
- 6.2. Lösungsverfahren für Transportprobleme

7. Mengenorientierte Verfahren für das Transportproblem

- 7.1. Die Lösung des Transportproblems mit der Simplex-Methode
- 7.2. Die Stepping-Stone-Methode
- 7.3. Bestimmung einer zulässigen Ausgangslösung
- 7.4. Implementierung primaler Methoden
- 7.5. Ganzzahligkeit und vollständige Unimodularität

8. Mengen- und preisorientierte Verfahren für Transport- und Umladeprobleme

- 8.1. Das Umladeproblem
- 8.2. Der LP-Ansatz für das Umladeproblem
- 8.3. Das Out-of-Kilter-Verfahren
- 8.4. Sonderprobleme

9. Die Ungarische Methode:

Ein preisorientiertes Verfahren zur Lösung des Zuordnungsproblems

- 9.1. Das Zuordnungsproblem
- 9.2. Duale Zulässigkeit
- 9.3. Flussänderung
- 9.4. Potentialänderung
- 9.5. Netzwerkorientierte Darstellung der Ungarischen Methode
- 9.6. Die Ungarische Methode in Transporttableauforn

Lösungen zu den Übungsaufgaben

Leseprobe 00852, Kurseinheit 1

1. Grundbegriffe

In Kapitel 1 werden Sie die grundlegenden mit Graphen und Netzwerken zusammenhängenden Begriffe, die in dieser Kurseinheit benötigt werden, kennenlernen. Es ist für die Graphentheorie charakteristisch, dass schon für die Beschreibung einfacher Modelle und Probleme eine große Zahl von Begriffen notwendig ist. Da viele dieser Begriffe aber unmittelbar einleuchtend und durch ähnliche oder gleich lautende Bezeichnungen aus dem gewöhnlichen Sprachgebrauch leicht ableitbar sind, wird Sie das Verstehen und Einprägen dieser Begriffe nicht sehr stark belasten.

1.1 Praktische Probleme, die auf Graphen und Netzwerke führen

Um Ihnen einen Eindruck von der Verschiedenartigkeit der Probleme zu vermitteln, die mit Hilfe von Graphen beschrieben werden können, wollen wir in diesem einführenden Abschnitt einige Beispiele betrachten.

Beispiel 1.1

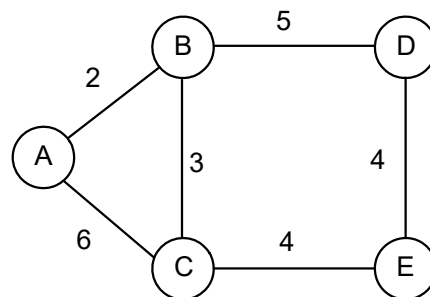


Abb. 1.1: Straßennetz

5 Orte A, B, C, D, E sind durch ein Straßennetz miteinander verbunden, das in Abb.1.1 veranschaulicht ist. Die Orte sind durch sogenannte *Knoten* und die direkten Straßenverbindungen zwischen benachbarten Orten durch Linien oder, wie man auch sagt, *Kanten* dargestellt. Die Entfernungen (in km) sind an den betreffenden Kanten notiert. Gesucht sind die (kürzesten) Entfernungen zwischen je 2 Orten sowie die entsprechenden Straßenverbindungen. Beispielsweise beträgt die Entfernung der Orte A und E 9 km, und die zugehörige Straßenverbindung führt über die Orte B und C.

Knoten

Kanten



Graph Das in Abb. 1.1 veranschaulichte System von Knoten und Kanten wird *Graph* genannt. Da die Kanten des Graphen mit Zahlen (in Beispiel 1.1 Entfernungen) „bewertet“ sind, spricht man auch von einem *bewerteten Graphen*.

[...]

3.2 Kürzeste Wege von einem zu allen Knoten (Baumalgorithmen)

[...]

Netzwerke Wir werden uns auf die Bestimmung kürzester Wege in *Netzwerken* beschränken. Kürzeste Ketten in einem bewerteten Graphen $G = [V, E; c]$ mit nichtnegativer Bewertungsfunktion c erhält man, indem man in dem „zugehörigen“ symmetrischen Digraphen \vec{G} (vgl. Abschnitt 1.2), wobei die Bewertung der Pfeile $\langle i, j \rangle$ und $\langle j, i \rangle$ von \vec{G} gleich der Bewertung der Kante $[i, j]$ von G ist, kürzeste Wege bestimmt.

[...]

Wir legen stets ein Netzwerk $N = \langle V, E; c \rangle$ mit reellwertiger Gewichtsfunktion c und, wie schon oben erwähnt, der Knotenmenge $V = \{1, \dots, n\}$ zugrunde. Wie in Abschnitt 1.7 setzen wir

$$c_{ij} = \begin{cases} 0, & \text{falls } i = j \\ c \langle i, j \rangle, & \text{falls } \langle i, j \rangle \in E \quad (i, j = 1, \dots, n) \\ \infty, & \text{sonst} \end{cases}$$

(vgl. (1.5)). Wir setzen voraus, dass N keine Zyklen negativer Länge enthält. Wir suchen kürzeste Wege von einem Startknoten a in N zu allen übrigen von a aus erreichbaren Knoten j von N sowie die entsprechenden Weglängen (Entfernungen).

Sei

$$d_j = \begin{cases} 0, & \text{falls } j = a \\ d \langle a, j \rangle, & \text{falls } j \in \dot{R}(a) \quad (j = 1, \dots, n) \\ \infty, & \text{falls } j \notin R(a) \end{cases} \quad (3.1)$$

wobei $d \langle a, j \rangle$ die Entfernung von a nach j ist (vgl. Kapitel 1.7). Für $j \in \dot{R}(a)$ sei $F_j = \langle a, \dots, k, j \rangle$ ein kürzester Weg von a nach j . $\langle k, j \rangle$ ist also der „letzte Pfeil“ auf diesem Weg. Der Teil $F_k = \langle a, \dots, k \rangle$ des Weges F_j muss ebenfalls ein kürzester Weg (und zwar von a nach k) sein. Gäbe es nämlich einen kürzeren Weg von a nach k , etwa F'_k , so würde F'_k zusammen mit dem Pfeil $\langle k, j \rangle$ einen Weg von a nach j liefern, der kürzer als der kürzeste Weg F_j wäre. Es gilt also $d_j = d_k + c_{kj}$.

[...]

Leseprobe 00852, Kurseinheit 2

[...]

5.2 Mediane und Zentren

5.2.1 Grundbegriffe und Definitionen

Modellgrundlage ist ein bewerteter ungerichteter Graph $G = [V, E, c; b]$, wobei die Bewertung c zunächst o.B.d.A. als Entfernung interpretiert wird. Gewichte b sind den Knoten zugeordnet, und es können etwa vorhandene Angebote oder Bedarfe quantifiziert werden. Die Länge einer kürzesten Kantenfolge mit den Endknoten i, j in einem bewerteten Graphen G heißt *Entfernung* (oder *Distanz*) der Knoten i und j , in Zeichen $d[i, j]$. Enthält G keine Kreise negativer Länge, so existiert für je zwei verschiedene Knoten $i, j \in V$, die miteinander verbunden sind, eine kürzeste Kantenfolge mit den Endknoten i, j und folglich die Entfernung $d[i, j]$. Sind zwei Knoten i, j von G nicht miteinander verbunden, so ist $d[i, j] = \infty$. Für die eingangs formulierten Lokationsprobleme verfolgt man naturgemäß eine knotenorientierte Betrachtung. So ist für ein Unternehmen etwa die Frage von Interesse, wie kann der Transportaufwand vom Standort aus zu allen Nachfragern bei unterschiedlichen Bedarfen bewertet werden. Formal bestimmt man hierzu für einen Knoten i die gewichtete Distanz $\sigma(i)$ zu allen Knoten $j \in V, j \neq i$. Für diese Berechnung ist es erforderlich, dass die kürzesten Entfernungen vom Knoten i zu allen anderen Knoten bekannt sind. Sollten diese Informationen nicht vorliegen, so ist vorab zunächst ein geeigneter Algorithmus zur Erstellung einer vollständigen Entfernungsmatrix anzuwenden (vgl. Abschnitte 3.2ff. der KE1).

**Entfernung /
Distanz**

$$\sigma(i) = \sum_{j \in V} d[i, j] \cdot b_j$$

[...]

| |
|--------------------------|
| Übungsaufgabe 5.2 |
|--------------------------|

Die fünf Dörfer Aalhus, Borscheidt, Churtingen, Dalenkamp und Estringen im Hochsauerland sind durch nur wenige Straßen miteinander verbunden; die Längen der direkten Verbindungen entnehmen Sie bitte folgender Matrix D . (Erinnerung: ∞ bedeutet, es gibt keine direkte Verbindung.)

$$D := \begin{pmatrix} 0 & 2 & \infty & \infty & \infty \\ 2 & 0 & 2 & \infty & 7 \\ \infty & \infty & 0 & \infty & 8 \\ 3 & \infty & 9 & 0 & \infty \\ \infty & 7 & \infty & 3 & 0 \end{pmatrix}$$

- a) Zeichnen Sie einen Graphen, durch den die obige Situation visualisiert wird und der alle Angaben enthält. Welche Verbindungen sind nur in einer Richtung befahrbar?
- b) Ein Dorf soll durch die Ansiedlung eines Einkaufszentrums aufgewertet werden. Analysen haben ergeben, dass von **Aalhus**, **Borscheidt** und **Churtingen** aus mit je 1000 Einkäufen pro Tag zu rechnen ist. **Dalenkamp** und **Estringen** dagegen sind größer; von hier aus werden 2000 Einkäufe täglich erwartet.
In welchem Dorf sollte das Einkaufszentrum gebaut werden, damit die Gesamtfahrstrecke für alle Tageseinkäufe minimal ist.
- c) Eine weitere Standortfrage beschäftigt die dortige Feuerwehr. In welchem Dorf sollte die neue Feuerwehrrstation gebaut werden, damit bei einem Einsatz der Weg zum am weitesten entfernt liegenden Dorf am kürzesten ist.



[...]

6. Modellierung von Transportproblemen

[...]

Beispiel 6.1

Ein homogenes Gut, das von den Lieferanten L_i in den Mengen a_i angeboten wird, soll zu den Kunden K_j mit den Nachfragen b_j transportiert werden. Die Kosten des Transportes einer Mengeneinheit vom Lieferanten L_i zum Kunden K_j seien c_{ij} .

Tab. 6.1 Ausgangsdaten des Transportproblems

| | | Kunden | | | | Angebot |
|-------------|-------|--------|-------|-------|-------|---------|
| | | K_1 | K_2 | K_3 | K_4 | a_i |
| Lieferanten | L_1 | 12 | 8 | 11 | 7 | 100 |
| | L_2 | 9 | 13 | 10 | 6 | 150 |
| | L_3 | 11 | 7 | 9 | 14 | 90 |
| Nachfrage | b_j | 80 | 140 | 70 | 50 | |

Wie viele Mengeneinheiten x_{ij} sollen vom Lieferanten L_i zum Kunden K_j transportiert werden, damit die Nachfrage erfüllt ist und die gesamten Transportkosten minimiert werden?



Allgemein formuliert lautet das *Transportproblem*: Ein homogenes Gut, das an den Orten $A_i, i=1, \dots, m$, in den Mengen a_i angeboten wird, soll zu den Orten $B_j, j=1, \dots, n$, transportiert werden, an denen eine Nachfrage von b_j besteht. Die Kosten des Transportes einer Mengeneinheit (ME) von Ort A_i zum Ort B_j betragen c_{ij} . Wieviele ME sollen vom Ort A_i zum Ort B_j transportiert werden, so dass die Nachfrage erfüllt wird und die Gesamttransportkosten minimal sind ?

[...]

Das dem Beispiel 6.1 analoge Datenschema hat dann die Form der Tab. 6.2.

Tab. 6.2: Daten zum Transportproblem

| | | Nachfrageorte | | | | Angebot |
|--------------|-------|---------------|----------|-----|----------|---------|
| | | B_1 | B_2 | ... | B_n | a_i |
| Angebotsorte | A_1 | c_{11} | c_{12} | ... | c_{1n} | a_1 |
| | A_2 | c_{21} | c_{22} | ... | c_{2n} | a_2 |
| | . | . | . | ... | . | . |
| | . | . | . | ... | . | . |
| | A_m | c_{m1} | c_{m2} | ... | c_{mn} | a_m |
| Nachfrage | b_j | b_1 | b_2 | ... | b_n | |

Das Problem der Transportkostenminimierung wird in dieser Form allgemein als das *klassische Transportproblem* bezeichnet. Es handelt sich um den einfachsten Spezialfall eines allgemeinen Transportproblems, das in Kapitel 8 ausführlich behandelt wird.

**klassisches
Transportproblem**

[...]

9.1 Das Zuordnungsproblem

[...]

Als Sonderfall des klassischen Transportproblems ($a_i = 1, i = 1, \dots, n; b_j = 1, j=1, \dots, n$) lässt sich das Zuordnungsproblem selbstverständlich als Fluss- bzw. als *Zirkulationsflussproblem* darstellen und formulieren.

**Zirkulations-
flussproblem**

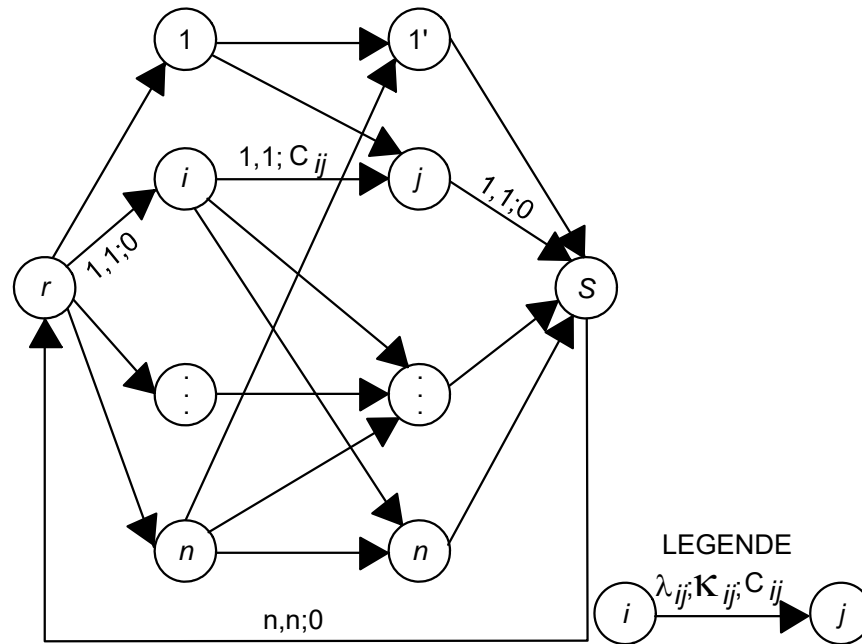


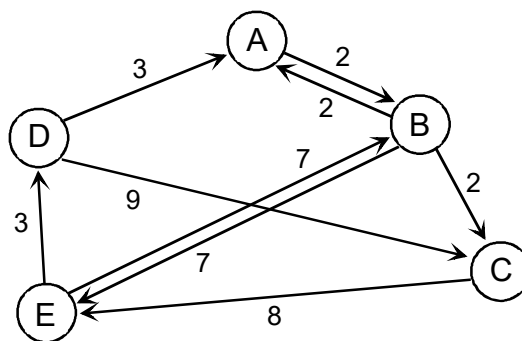
Abb. 9.1: Darstellung des Zuordnungsproblems als Zirkulationsproblem in $N = \langle V, E; \lambda, \kappa; c \rangle$

Lösungen der Übungsaufgaben

[...]

Übungsaufgabe 5.2

a)



Dem Graphen ist zu entnehmen, dass die Verbindungen von $\langle B, C \rangle$, $\langle C, E \rangle$, $\langle D, A \rangle$, $\langle D, C \rangle$ und $\langle E, D \rangle$ nur in einer Richtung befahrbar sind.

- b) Für den Einkauf sind jeweils die Hin- und Rückfahrt auf dem kürzesten Weg zu betrachten. Aus dem in a) erstellten Graphen sind deshalb entsprechend die kürzesten Entfernungen zwischen allen Knoten abzulesen

und in einer Distanzmatrix zusammenzustellen. Für einen komplexeren Graphen liefert der Tripelalgorithmus das gewünschte Ergebnis.

$$D^* := \begin{pmatrix} 0 & 2 & 4 & 12 & 9 \\ 2 & 0 & 2 & 10 & 7 \\ 14 & 15 & 0 & 11 & 8 \\ 3 & 5 & 7 & 0 & 12 \\ 6 & 7 & 9 & 3 & 0 \end{pmatrix}$$

Da die Anzahl der zu erwartenden Einkäufe je Dorf unterschiedlich ist, muss eine zusätzliche Gewichtung vorgenommen werden. Die Ergebnisse (in 1000 km) sind in der nachfolgenden Matrix zusammengefasst:

$$\begin{bmatrix} 0 & 2 & 4 & 12 & 9 \\ 2 & 0 & 2 & 10 & 7 \\ 14 & 15 & 0 & 11 & 8 \\ 3 & 5 & 7 & 0 & 12 \\ 6 & 7 & 9 & 3 & 0 \end{bmatrix} \begin{matrix} 48 \\ 38 \\ 67 \\ 39 \\ 28 \end{matrix}$$

34 41 38 39 48

Somit ergibt sich beispielsweise bei einer Wahl des Standorts „Aalhus“, dass von allen Einkaufenden aus den verschiedenen Dörfern insgesamt zunächst einmal 34.000 km zurückzulegen sind, um zum Kaufhaus zu gelangen, und dann 48.000 km, um wieder nach Hause zu kommen. In der Summe berechnet sich für A: 82.000 km, für B: 79.000 km, für C: 105.000 km, für D: 78.000 km und für E: 76.000 km. Das Kaufhaus sollte somit unter den gegebenen Annahmen in Estringen gebaut werden.

- d) Für den Bau des Feuerwehrhauses ergibt sich folgende Situation. Hier muss nur der zurückzulegende Weg zum Einsatzort betrachtet werden. Da die weiteste Entfernung minimal sein soll, ist allerdings auch hier der Standort Estringen zu wählen. In folgender Übersicht ist die weiteste Entfernung vom Standort aus rechts neben der Matrix notiert. Für Estringen ist der Wert mit 9 km minimal.

$$\begin{bmatrix} 0 & 2 & 4 & 12 & 9 \\ 2 & 0 & 2 & 10 & 7 \\ 14 & 15 & 0 & 11 & 8 \\ 3 & 5 & 7 & 0 & 12 \\ 6 & 7 & 9 & 3 & 0 \end{bmatrix} \begin{matrix} 12 \\ 10 \\ 15 \\ 12 \\ 9 \end{matrix}$$



Gliederung

Kurs 00857 »Intelligente Strategien in Theorie und Praxis«

1. Komplexität

- 1.1 Allgemeine Komplexitätsbetrachtungen
- 1.2 Die Komplexität spezieller Algorithmen

2. Exakte Methoden

- 2.1. Das Branch & Bound-Verfahren
- 2.2. Das A*-Verfahren als Bestensuchverfahren
- 2.3. Vergleich von A*- und Branch & Bound-Verfahren
- 2.4. Beispiel zu Suchstrategien

3. Klassische Heuristiken

- 3.1. Klassifizierung
- 3.2. Eröffnungsverfahren
- 3.3. Verbesserungsverfahren
- 3.4. Kritische Beurteilung klassischer Heuristiken
- 3.5. Praxisbeispiel Bandbreitennutzung

4. Simulated Annealing

- 4.1. Physikalischer Hintergrund und Idee des Simulated Annealing
- 4.2. Algorithmische Realisierung
- 4.3. Beispiele zum Simulated Annealing
- 4.4. Übungen zur Anwendung von Simulated Annealing

5. Tabu Search

- 5.1. Die Grundidee des Tabu-Search
- 5.2. Vermeidung von Zyklen
- 5.3. Berechnung des TSP-Beispiels
- 5.4. Ein Praxisbeispiel: Digitaler Hörfunk
- 5.5. Übungen zur Anwendung von Tabu Search

6. Genetische Algorithmen

- 6.1. Einführung in die Genetischen Algorithmen
- 6.2. Basiswissen zum Genetischen Algorithmus
- 6.3. Beispiele zum Genetischen Algorithmus
- 6.4. Genetische Algorithmen in der Praxis
- 6.5. Übungen zu Genetischen Algorithmen

Lösungen zu den Übungsaufgaben

Leseprobe 00857

2. Exakte Methoden

[...]

2.2 Das A*-Verfahren als Bestensuchverfahren

Sehr stark verwandt mit dem Branch & Bound-Verfahren ist das *A*-Verfahren*. Den Grundstein hierzu legten bereits 1980 HART et al. Im Folgenden wird es skizziert und im Anschluss auf das TSP angewendet. Dazu ist das TSP genauer zu präzisieren.

Wie allgemein üblich, wollen wir eine exakte Definition des TSPs in der Terminologie der Graphentheorie (siehe Kurs 00852 „Optimierung in Graphen“) geben. Das Problem kann durch einen bewerteten Digraphen repräsentiert werden, bei dem die zu besuchenden Orte den Knoten, die Strecken zwischen je zwei Orten den Kanten und die Länge dieser Strecken der Kantenbewertung entsprechen:

| |
|-----------------------|
| Definition 2.1 |
|-----------------------|

Sei $D=(V, E; d)$ ein bewerteter (vollständiger) Digraph mit der Knotenmenge V ($|V| = n$), der Kantenmenge $E = V \times V$ und der Bewertung $d: E \rightarrow [0, \infty)$.

- | | | |
|----|---|---|
| a) | Wir bezeichnen mit (v_1, \dots, v_ℓ) eine Tour über die Orte v_1, \dots, v_ℓ , falls $v_i \in V$ ($1 \leq i \leq \ell \leq n+1$) und $v_i \neq v_j$ für $i \neq j$, $1 \leq i, j \leq \ell-1$ gilt. | Tour |
| b) | Eine Tour (v_1, \dots, v_ℓ) nennt man <ul style="list-style-type: none"> – offen, wenn $v_1 \neq v_\ell$ gilt, – geschlossen, wenn $v_1 = v_\ell$ gilt, – vollständig, falls jeder Ort von V in (v_1, \dots, v_ℓ) enthalten ist – Teiltour, falls (v_1, \dots, v_ℓ) nicht jeden Ort von V beinhaltet und – Rundreise, falls sie geschlossen und vollständig ist. | offene Tour geschlossene Tour vollständige Tour Teiltour Rundreise |
| c) | Die Länge einer Tour (v_1, \dots, v_ℓ) ist definiert durch $\sum_{i=1}^{\ell-1} d(v_i, v_{i+1})$. | Länge einer Tour |
| d) | Das Problem der Bestimmung einer Rundreise minimaler Länge über V bezeichnet man als Traveling-Salesman-Problem (TSP) . | Traveling-Salesman-Problem (TSP) |

symmetrisches TSP e) Gilt $d(v_i, v_j) = d(v_j, v_i)$ für alle $1 \leq i, j \leq n$, so spricht man von einem **symmetrischen TSP**.

geometrisches TSP f) Gilt $d(v_i, v_j) + d(v_j, v_k) \geq d(v_i, v_k)$ für alle $1 \leq i, j, k \leq n$, so spricht man von einem **geometrischen TSP**.



[...]

Beispiel 2.1

Betrachten Sie das symmetrische Traveling-Salesman-Problem mit den 5 Orten **A**, **B**, **C**, **D** und **E** und folgender Entfernungsmatrix :

| | A | B | C | D | E |
|----------|----------|----------|----------|----------|----------|
| A | 0 | 12 | 9 | 8 | 17 |
| B | 12 | 0 | 5 | 9 | 12 |
| C | 9 | 5 | 0 | 5 | 12 |
| D | 8 | 9 | 5 | 0 | 10 |
| E | 17 | 12 | 12 | 10 | 0 |

Zu einem symmetrischen TSP mit n Orten existieren $(n-1)!/2$ verschiedene Rundreisen; für das konkrete Beispiel listen wir alle 12 auf:

ABCDEA, ABCEDA, ABDCEA, ABDECA,
ABECDA, ABEDCA, ACBDEA, ACBEDA,
ACDBEA, ACEBDA, ADBCEA, ADCBEA.



Die im Verfahren A* aufzubauenden Touren des TSP können nun durch Zustände repräsentiert werden. Ausgangszustände bilden Teiltouren mit nur einem Ort; ein Nachfolgezustand wird erzeugt, indem die Tour um einen Ort verlängert wird, und vollständige Touren werden durch Zielzustände repräsentiert. Die Länge einer Rundreise entspricht dann den Gesamtkosten zur Erzeugung des Ziel- aus einem Ausgangszustand. Ein TSP über n Orte repräsentieren wir für eine Behandlung mit dem A*-Verfahren wie folgt:

| | |
|--|---|
| Tour (v_1, \dots, v_ℓ) , $(1 \leq \ell \leq n+1)$ | Zustand k |
| Tour (v) , die nur den Ort v umfasst | Ausgangszustand k_0 |
| Tour $(v_1, \dots, v_{\ell+1})$ falls $\ell \leq n$ | Nachfolgezustand von (v_1, \dots, v_ℓ) |
| Rundreise (v_1, \dots, v_{n+1}) | Zielzustand k^* |
| Länge zwischen den Orten v_i und v_j | $c(v_i, v_j)$ |

[...]

A*-Algorithmus zur Lösung des Traveling-Salesman-Problems

OPEN := {(A)}

A*-Algorithmus

$g((A)) := 0$

(IT_{A*}) Wähle Knoten $k \in OPEN$ mit $f(k)$ minimal

Falls k Zielknoten ist, dann STOP;

k ist Lösung.

Erzeuge die Menge aller Nachfolgeknoten $N(k)$ des Knotens k

OPEN := OPEN \cup $N(k) \setminus \{k\}$

Für alle $k' \in N(k)$:

$g(k') := g(k) + c(k, k')$

Berechne NN(k') mittels Nearest-Neighbour

$h(k') := 0,8 \cdot NN(k')$

$f(k') := g(k') + h(k')$

Falls OPEN $\neq \emptyset$, dann gehe zu (IT_{A*})

sonst existiert keine Lösung.

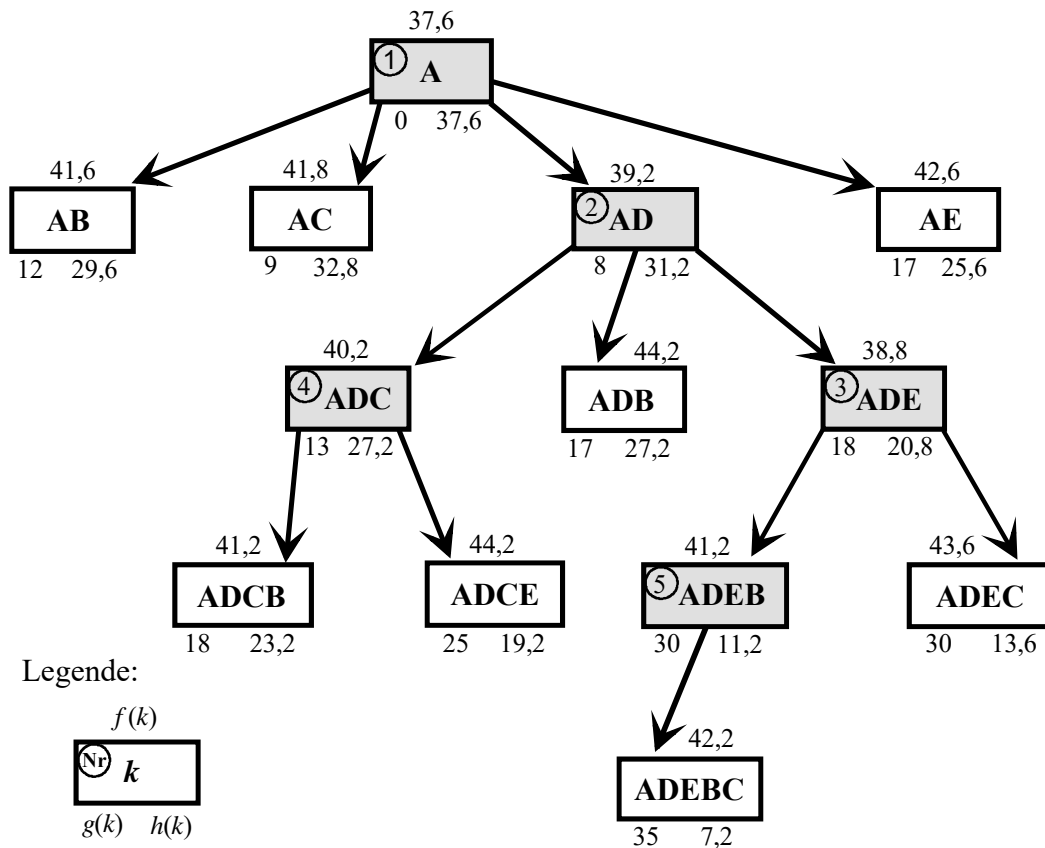


Abbildung 2.2: Der Suchgraph des A* zu Beispiel 2.1 nach 5 Iterationen

[...]

5.5 Übungen zur Anwendung von Tabu Search

[...]

Übungsaufgabe 5.6

Oma Elli verfügt über ein Guthaben von 500,- EURO und möchte sich nun 3 verschiedene Aktien als Alterssicherung zulegen. Ihr Anlageberater sucht 5 vielversprechende Aktien als Angebot aus; sie sind zusammen mit dem derzeitigen Kurs und einer Kursprognose in Tabelle 5.2 zusammengestellt.

Tabelle 5.2: Aktiengesellschaften mit jeweiligem Aktienkurs und Prognose

| Aktie | Kurs | Kursprognose |
|-------------------|------------|--------------|
| 1) SEIER AG | 100,- EURO | +3,5% |
| 2) MITSUTACHI AG | 120,- EURO | +4,0% |
| 3) KLEINGEIST VZ. | 150,- EURO | +4,0% |
| 4) DEMAGUSSA ST. | 180,- EURO | +5,0% |
| 5) KABOOM AG | 200,- EURO | +5,5% |

Lösungen des Rucksackproblems lassen sich – wie bereits in Beispiel 5.2 beschrieben – durch Vektoren $x = (x_1, x_2, x_3, x_4, x_5)$ darstellen. Zu einer benachbarten Lösung gelangt man über eine Austauschoperation, d.h. ein Gegenstand wird entfernt und dafür ein anderer hinzugefügt. Die Zulässigkeit ist entsprechend zu beachten.

- a) Geben Sie zu den drei folgenden Lösungen die jeweilige Nachbarschaft an.
- i) $(0,1,1,0,1)$ ii) $(1,1,0,0,1)$ iii) $(1,0,1,1,0)$
- b) Nennen Sie zur Eigenschaftsmenge $E = \{E_1, \dots, E_5\}$, wobei E_i Wert der i -ten Komponente, sowohl das from- als auch das to-Attribut zu folgender Entscheidung von Oma Elli:
- Sie hält zur Zeit die zweite, vierte und fünfte Aktie. Sie entschließt sich nun, die DEMAGUSSA Aktie gegen ein Papier der SEIER AG auszutauschen (vgl. Abbildung 5.4).

| Aktie | Preis | Kursprognose |
|---|-------|--------------|
| <input type="checkbox"/> 1:Seier AG | 100,- | +3,5% |
| <input checked="" type="checkbox"/> 2:Mitsutachi AG | 120,- | +4,0% |
| <input type="checkbox"/> 3:Keingeist Vz. | 150,- | +4,0% |
| <input checked="" type="checkbox"/> 4:Demagussa St. | 180,- | +5,0% |
| <input checked="" type="checkbox"/> 5:Kaboom AG | 200,- | +5,5% |

Aktuell: $x=[0,1,0,1,1]$

"Doch lieber Seier statt Demagussa !"

also $x'=[1,1,0,0,1]$




Abbildung 5.4: Änderung der Aktienausswahl

Notieren Sie sowohl from- als auch to-Attributenvektor zu obiger Entscheidung.



[...]

Lösungen der Übungsaufgaben

[...]

Übungsaufgabe 5.6

- a) Aufgrund der definierten Nachbarschaftsstruktur besitzt jeder Knoten des Nachbarschaftsgraphen genau sechs adjazente Knoten. Für jeden der drei Gegenstände, die entfernt werden können, gibt es jeweils zwei Möglichkeiten einen anderen Gegenstand hinzu zu nehmen.
- Für die in der Aufgabenstellung angegebenen Knoten gelten unter Berücksichtigung der Zulässigkeit somit folgende Nachbarschaftsbeziehungen.

i.) $(0,1,1,0,1)$

Nachbarn: $(0,1,1,1,0)$ $(1,1,1,0,0)$
 $(1,1,0,0,1)$ $(1,0,1,0,1)$ $(0,1,0,1,1)$

ii.) $(1,1,0,0,1)$

Nachbarn: $(0,1,0,1,1)$ $(1,0,1,0,1)$ $(1,0,0,1,1)$
 $(1,1,1,0,0)$ $(0,1,1,0,1)$ $(1,1,0,1,0)$

iii.) $(1,0,1,1,0)$

Nachbarn: $(1,0,1,0,1)$ $(1,0,0,1,1)$ $(1,1,1,0,0)$
 $(0,1,1,1,0)$ $(1,1,0,1,0)$

- b) Der from-Attributenvektor lautet für diese Operation $(0, -, -,1, -)$; analog ist als to-Attributenvektor in diesem Beispiel $(1,-,-,0,-)$ festzuhalten.

