Gliederung

Leseprobe

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 $d\colon E \to [0,\infty)$.

Tour

a) *Wir bezeichnen mit $(v_1,\ldots, v_\ell)$ eine **Tour** über die Orte $v_1,\ldots, v_\ell$, falls $v_i \in V$ $(1 \le i \le \ell \le n+1)$ und $v_i \neq v_j$ für $i \neq j$, $1 \le i, j \le \ell-1$ gilt.*

offene Tour

b) *Eine Tour $(v_1,\ldots, v_\ell)$ nennt man*

geschlossene Tour

vollständige Tour

Teiltour

Rundreise

- ***offen**, wenn $v_1 \neq v_\ell$ gilt,*
- ***geschlossen**, wenn $v_1 = v_\ell$ gilt,*
- ***vollständig**, falls jeder Ort von V in $(v_1,\ldots, v_\ell)$ enthalten ist*
- ***Teiltour**, falls $(v_1,\ldots, v_\ell)$ nicht jeden Ort von V beinhaltet und*
- ***Rundreise**, falls sie geschlossen und vollständig ist.*

Länge einer Tour

c) *Die **Länge einer Tour** $(v_1,\ldots, v_\ell)$ ist definiert durch* $\sum_{i=1}^{\ell-1} d(v_i, v_{i+1})$.

Traveling-Salesman-Problem (TSP)

d) *Das Problem der Bestimmung einer Rundreise minimaler Länge über V bezeichnet man als **Traveling-Salesman-Problem** (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****.* **symmetrisches 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****.* **geometrisches 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 $(v_1,\ldots, v_\ell)$, $(1 \leq \ell \leq n+1)$	Zustand k
Tour (v), die nur den Ort v umfasst	Ausgangszustand k_0
Tour $(v_1,\ldots, v_{\ell+1})$ falls $\ell \leq n$	Nachfolgezustand von $(v_1,\ldots, v_\ell)$
Rundreise $(v_1,\ldots,v_{n+1})$	Zielzustand k^*
Länge zwischen den Orten v_i und v_j	$c(v_i, v_j)$

[…]

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

A*-Algorithmus

OPEN := {(**A**)}

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

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

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

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

sonst existiert keine Lösung.

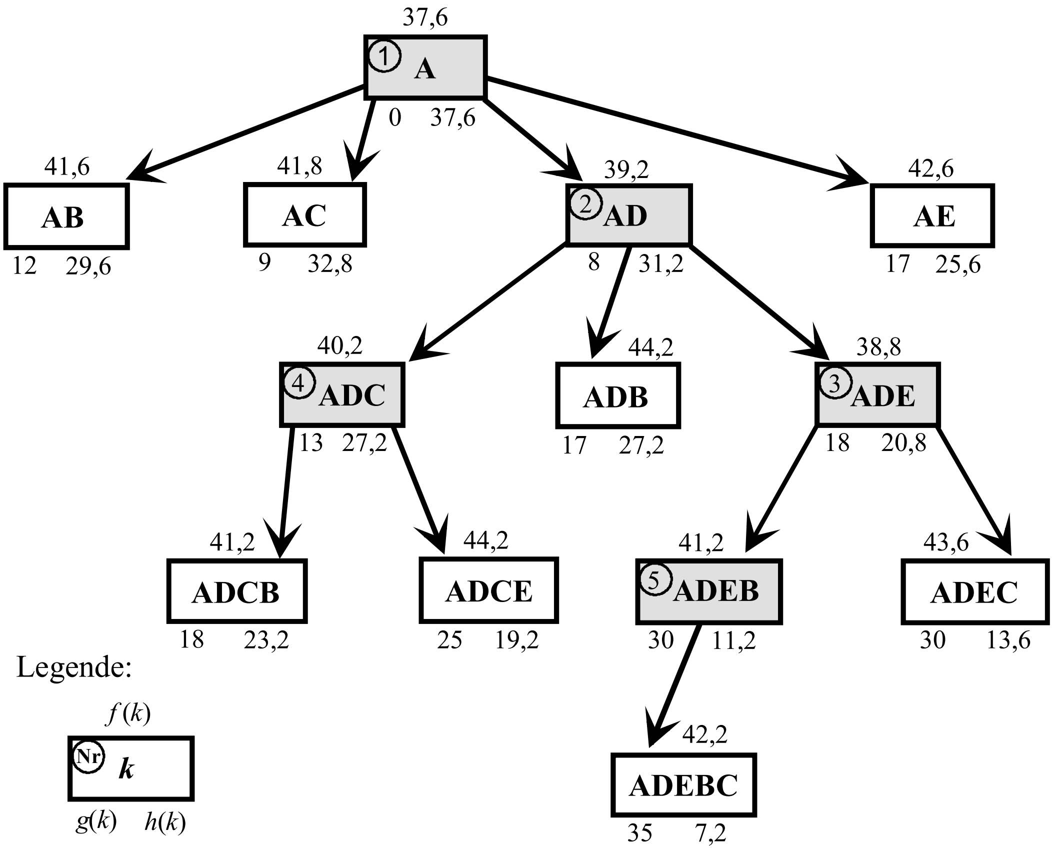


Abbildung 2.2: Der Suchgraph des A* zu Beispiel 2.1 nach 5 Iterationen

[…]

3.5 Praxisbeispiel Bandbreitennutzung

Das Verfahren der Bandbreitennutzung

In einem Telephon-Netz sollen Verbindungen zwischen jeweils zwei Teilnehmern hergestellt werden. Die Verbindungen können normalerweise nicht direkt, sondern nur indirekt über verschiedene Zwischenstationen, d.h. Knotenpunkte, realisiert werden. Zwei Stationen sind jeweils durch eine Leitung mit einer gegebenen Übertragungskapazität miteinander verbunden. Der Weg, über den ein Ruf zwischen zwei Punkten geschaltet wird, ist im allgemeinen nicht eindeutig bestimmt. Jede erfolgreich hergestellte Verbindung benötigt nun eine bestimmte Bandbreite zur Übertragung der Daten und erbringt für den Netzbetreiber einen gewissen Erlös. Die benötigte Gesamtkapazität ist im Allgemeinen nicht ausreichend, um alle Verbindungen herzustellen. Das Problem besteht also in der Auswahl der herzustellenden Verbindungen und geeignetem „Routing“ mit dem Ziel der Erlösmaximierung unter Berücksichtigung der jeweiligen Bandbreiten.

[…]

4.3 Beispiele zum Simulated Annealing

4.3.1 Das Färbungsproblem

Problemformulierung

Das Färbungsproblem besteht darin, die Knoten eines gegebenen Graphen mit möglichst wenig Farben so einzufärben, dass jeweils zwei benachbarte Knoten verschiedene Farben aufweisen. Eine Färbung, die letztere Eigenschaft hat, heißt zulässig. Gesucht ist also aus der Menge aller zulässigen die Färbung mit minimaler Farbenzahl.

Als Beispiel für die Praxisrelevanz dieser Fragestellung wird die Verkehrsregelung einer Kreuzung durch Ampelschaltungen betrachtet. Der Verkehr soll durch elf Ampeln gesteuert werden, sieben Ampeln sind für Fahrzeuge vorgesehen, vier für die Fußgängerüberwege. Die möglichen Verkehrswege der Fahrzeuge sind durch Richtungspfeile gekennzeichnet. Stehen alle Ampeln gleichzeitig auf Grün, ist das Verkehrschaos – wie in Abbildung 4.4 angedeutet – absehbar. Gesucht ist daher eine möglichst geringe Zahl von Phasen zulässiger Ampelschaltungen, so dass jeder Verkehrsteilnehmer die Kreuzung passieren kann.

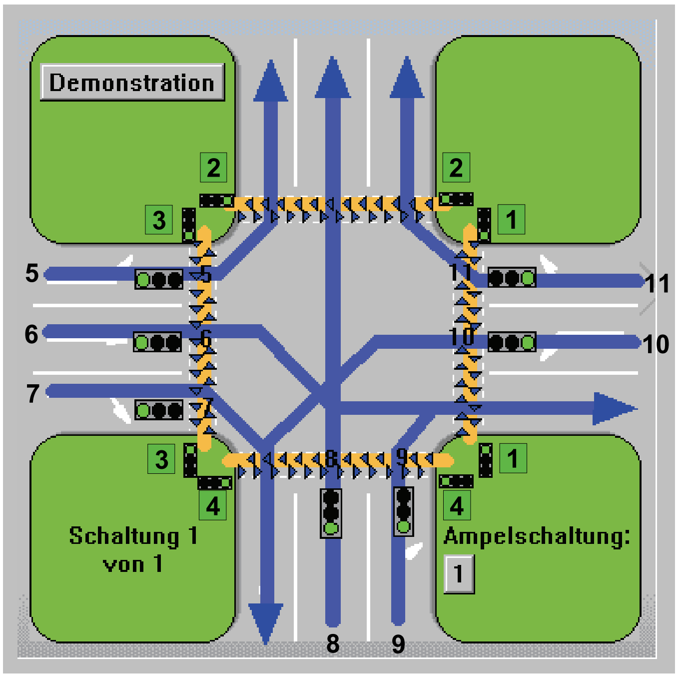


Abbildung 4.4: Eine Ampelschaltung mit einer (ständigen) Grünphase

Formal lässt sich das Ampelproblem in einen Graphen übertragen, in dem die Ampeln durch Knoten repräsentiert sind. Dürfen zwei Ampeln nicht gleichzeitig auf Grün stehen, sind die zugehörigen Knoten durch Kanten im Graphen verbunden. Findet man eine zulässige Färbung, entspricht jede Farbe einer konfliktfreien Ampelschaltung; die Ampeln gleicher Farbe stehen gleichzeitig auf Grün. Ziel ist die Minimierung der Farbenzahl in einer Färbung, da ja somit im Realproblem (s.o.) die Zahl der konfliktfreien Ampelschaltungen = Phasen minimiert wird. Spätestens nach einem vollständigen Phasendurchlauf hat jeder Verkehrsteilnehmer die Kreuzung passiert.

[…]

Anwendung des Algorithmus zum Färbungsproblem

Das folgende Beispiel beschränkt sich auf einen Graphen mit 11 Knoten. Zunächst sei auf einfache Weise eine Ausgangslösung bestimmt. Dazu betrachtet man die Knoten in der Reihenfolge ihrer Nummerierung und färbt sie nacheinander so ein, dass ein Knoten eine Farbe erhält, die noch für keinen adjazenten Knoten verwendet wurde.

Man beginnt also etwa mit der Farbe weiß und färbt damit die Knoten **1**, **2**, **3** und **4**. Bei Knoten **5** muss man bereits zur nächsten Farbe „quer gestreift“ übergehen, da sonst zwei durch eine Kante verbundene Knoten die gleiche Farbe aufweisen würden. Die Einfärbung des gesamten Graphen erfolgt somit mit den vier Farben

„weiß“ {**1**, **2**, **3**, **4**}, „quer gestreift“ {**5**, **6**, **7**, **11**}, „längs gestreift“ {**8**, **9**}und „kariert“ {**10**}; damit ist aber noch nicht die optimale Lösung gefunden (vgl. Abbildung 4.7).

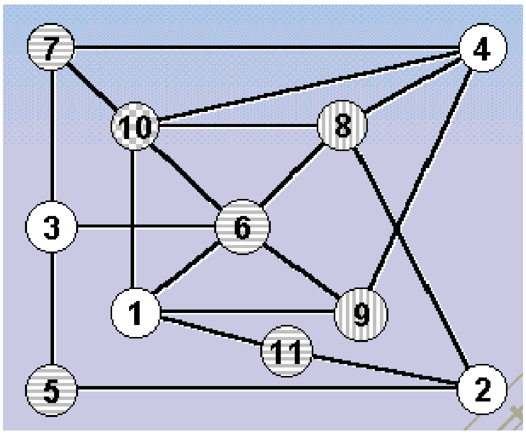


Abbildung 4.7: Einfärbung des gesamten Graphen

[…]

Nun initialisiert man die Starttemperatur etwa mit dem Wert $\sqrt{n}$, d.h. Wurzel aus der Anzahl der Knoten des Graphen; im Beispiel ist dies der Wert $\sqrt{11} \approx 3{,}3$. Die durch Umfärben erzeugte Lösung stellt noch keine zulässige Färbung des Graphen dar. Es werden jetzt solange benachbarte Lösungen durch Umfärben jeweils eines Knotens erzeugt, bis der Zielfunktionswert Null ist oder bis keine benachbarte Lösung gefunden werden kann, deren Zielfunktionswert unter der aktuellen Kühltemperatur liegt.

So entsteht z.B. ein Nachbar durch Umfärben von Knoten **8** auf „quer gestreift“. Der Zielfunktionswert bleibt bei 1 und damit unter der Kühltemperatur (3,2). Die Temperatur wird nun reduziert und eine neue zufällige Lösung z.B. durch weißes Färben von Knoten **6** erzeugt. Der Zielfunktionswert hat sich jetzt verschlechtert (auf 2), liegt jedoch noch unter der aktuellen Temperatur (3,1). Also reduziert man die Temperatur noch weiter. Anschließend führe der Zufallsprozess zu einer Färbung des Knotens **3** mit „längs gestreift“ (Zielfunktionswert von 1, aktuelle Temperatur: 3,0). In zwei weiteren Schritten erhält man letztendlich eine zulässige Färbung des Graphen mit drei Farben, wie in Abbildung 4.9 zu sehen ist.

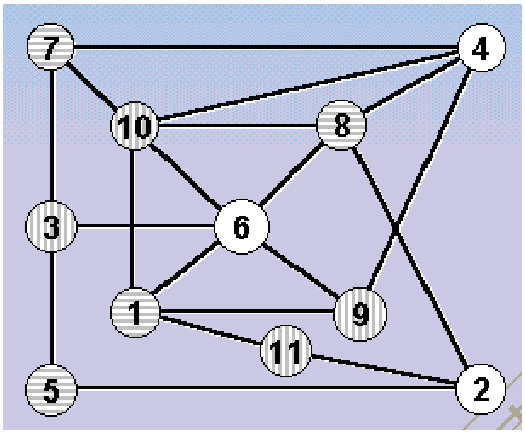


Abbildung 4.9: Lösung des Färbungsproblems

Überträgt man die Lösung auf das eingangs beschriebene Problem der Ampelschaltung, so ergeben sich insgesamt die in Abbildung 4.10 zu sehenden (gefahrlosen) Kreuzungsüberquerungen.

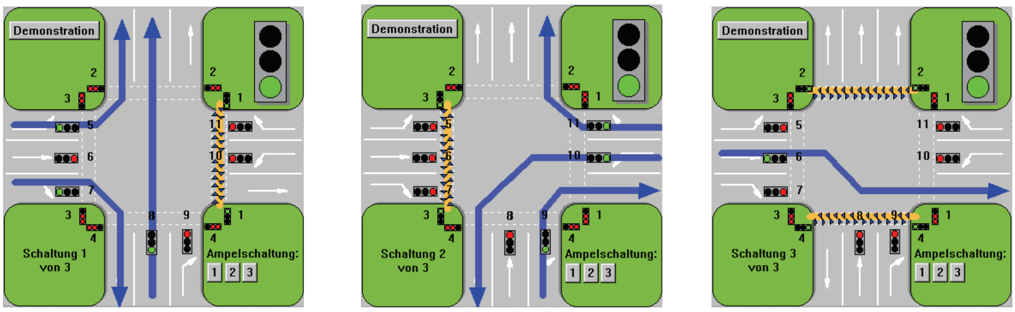


Abbildung 4.10: Phasen konfliktfreier Ampelschaltungen

[…]

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 Eigenschaftenmenge $E = \{E_1,..., E_5\}$, 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).

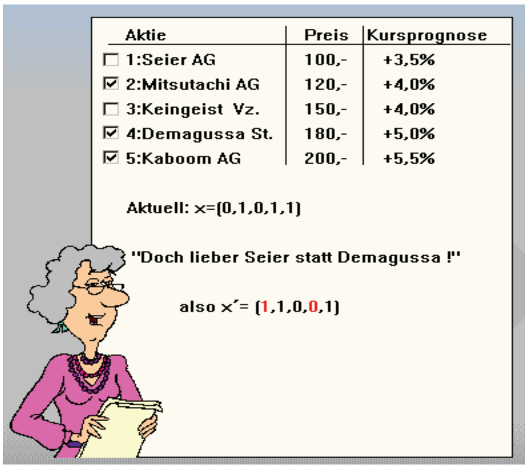


Abbildung 5.4: Änderung der Aktienauswahl

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

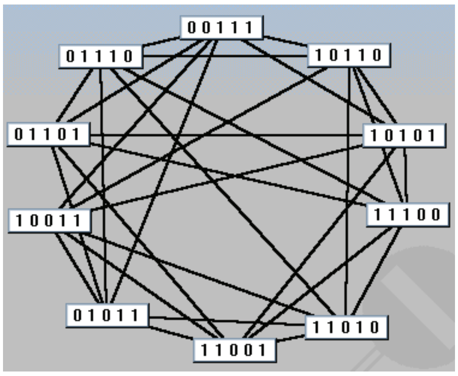
[…]

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 (vgl. nachfolgende Abbildung).



Für die in der Aufgabenstellung angegebenen Knoten gelten somit folgende Nachbarschaftsbeziehungen.

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

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

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

✓