W. GAUL

Optimale Wege bei zeitabhängigen Nebenbedingungen

Zusammenfassung:
Es wird über die Existenz und Berechenbarkeit von Wegen berichtet, falls für den zugrundeliegenden Graphen zeitabhängige Nebenbedingungen vorgegeben sind. Die Nebenbedingungen machen Aussagen über die Änderungen der Kosten- und Entfernungsstruktur und über Möglichkeiten der Benutzung des Graphen zu gewissen Zeitpunkten. Es werden Lösungsmöglichkeiten diskutiert, Rechenverfahren beschrieben und an einem Beispiel erläutert.

Einleitung

Netzwerkprobleme unter Berücksichtigung zeitabhängiger Nebenbedingungen finden in neuerer Zeit ein wachsendes Interesse. Der Übergang zur dynamischen Betrachtungsweise wurde schon früh von FORD/FULKERSON [7], [8] durch ihre Untersuchungen über dynamische Flüsse begonnen. Die Berechnung kürzester Wege mit zeitabhängigen Kantenlängen wurde zuerst von COOKE/HALSEY [4] mittels dynamischer Optimierung, dann von DREYFUS [6] und KLAFSZKY [14] durch Modifikation des DIJKSTRA [5]- Verfahrens behandelt. Erst kürzlich wurden in HALPERN/PRIESS [12] und GAUL [10] zusätzliche zeitabhängige Nebenbedingungen, die Benutzungs- und Aufenthaltsverbotsbedingungen in den Punkten und Kanten der zugrundeliegenden Graphen beschreiben, bei der Angabe solcher Weg-Verfahren berücksichtigt. In der vorliegenden Arbeit werden die in [10] beschriebenen Ergebnisse benutzt, um optimale Wege innerhalb vorgegebener Zeitperioden zu bestimmen, falls zusätzlich Kostenschranken und von den Abreisezeitpunkten abhängige Kosten vorgegeben sind. Dabei können ebenfalls zeitabhängige Benutzungs- und Aufenthaltsverbotsbedingungen berücksichtigt werden. Das vorgeschlagene Verfahren wird an einem Beispiel verdeutlicht.

Formulierung des Problems

Sei $G(P,K,l,k)$ ein schlichter, endlicher, gerichteter, zusammenhängender Graph (ein Netzwerk), wobei P die Punkte, $K \subset P \times P \setminus \{(i,i) \mid i \in P\}$ die Kanten, $l_{ij}(t)$ bzw. $k_{ij}(t)$ für $(i,j) \in K$, $t \in \mathcal{T} = \{0,1,\dots,T\}$ nicht negative, ganzzahlige, von der Benutzungszeit t für (i,j) abhängige Kantenlängen bzw. Kosten beschreibt. Ein Weg von a nach b, $a,b \in P$, der zur Zeit $\tau \in \mathcal{T}$ in $a \in P$ beginnt, wird durch

$$w(a,b,\tau) = (P_{w(a,b,\tau)}, A_{w(a,b,\tau)}, D_{w(a,b,\tau)})$$

beschrieben, wobei

$$P_{w(a,b,\tau)} = (a = p_o, p_1, \dots, p_m = b),$$
$$p_i \in P, \; i \in I_{w(a,b,\tau)} = \{0,1,\dots,m\}$$

die Folge der Punkte angibt, die dabei (in dieser Anordnung) durchlaufen werden. (Natürlich muß dann eine Kantenfolge

$$K_{w(a,b,\tau)} = ((p_o,p_1),\dots,(p_{m-1},p_m)),$$
$$(p_i,p_{i+1}) \in K, \; i \in I_{w(a,b,\tau)} - \{m\}$$ existieren.)

Für die durch $w(a,b,\tau)$ realisierten Ankunfts- bzw. Abreisezeiten

$$A_{w(a,b,\tau)} = (A^\tau(p_o), A^\tau(p_1), \dots A^\tau(p_m)) \quad \text{bzw.}$$
$$D_{w(a,b,\tau)} = (D^\tau(p_o), \dots, D^\tau(p_{m-1})), \; D^\tau(p_i) \in \mathcal{T}$$

gilt zusätzlich

$$D^\tau(p_i) \geq A^\tau(p_i), \qquad i \in I_{w(a,b,\tau)} - \{m\} \tag{1}$$
$$A^\tau(p_o) = \tau \tag{2}$$
$$A^\tau(p_m) \leq T \tag{3}$$
$$A^\tau(p_i) = D^\tau(p_{i-1}) + l_{p_{i-1}p_i}(D^\tau(p_{i-1})), \quad i \in I_{w(a,b,\tau)} - \{0\} \tag{4}$$

Die Wartezeit in p_i beträgt $D^\tau(p_i) - A^\tau(p_i)$.

21. Intern.Wiss.Koll. TH Ilmenau 1976

Vortragsreihe
"Theorie der Graphen und Netzwerke und ihre Anwendungen"

Wenn im Punkt p Warteverbote bestehen während der Zeitintervalle $V_p = \bigcup [t^p_{1k}, t^p_{2k}]$, t^p_{1k}, t^p_{2k} ganzzahlig, $k=1,\dots,n(p)$, $\bigcap [t^p_{1k}, t^p_{2k}] = \emptyset$, hat man zusätzlich zu beachten

$$\begin{aligned} A^\tau(p) \in [t^p_{1k}, t^p_{2k}] &\Rightarrow D^\tau(p) = A^\tau(p) \\ t^p_{2k} < A^\tau(p) < t^p_{1(k+1)} &\Rightarrow D^\tau(p) \leq t^p_{1(k+1)} \end{aligned} \tag{5}$$

~~(Intervalle z.B. der Form~~ $[t^p_{1k}, t^p_{2k})$ kann man durch $[t^p_{1k}, t^p_{2k}-1]$ berücksichtigen.)

$$l(w(a,b,\tau)) = \sum_{K_{w(a,b,\tau)}} l_{ij}(D^\tau(i)) + D^\tau(i) - A^\tau(i)$$

ist die Länge

$$k(w(a,b,\tau)) = \sum_{K_{w(a,b,\tau)}} k_{ij}(D^\tau(i)) + k(A^\tau(i), D^\tau(i) - A^\tau(i))$$

sind die Kosten von $w(a,b,\tau)$. Dabei wird zur Vereinfachung in dieser Arbeit $k(.,.) \equiv 0$ gesetzt (siehe auch die Beschreibung von Algorithmus II).

$w(a,a,\tau)$ heißt l- (oder k-)positiver, negativer oder Null-Zyklus, wenn $l(w(a,a,\tau))$ (oder $k(w(a,a,\tau))$) positiv, negativ oder gleich Null ist (Zur Vereinfachung und Vermeidung von negativen Zyklen seien alle dem Graphen zugeordneten Werte nicht negativ.)

$w(a,b,\tau)$ heißt einfach bzw. elementar, wenn $K_{w(a,b,\tau)}$ bzw. $P_{w(a,b,\tau)}$ nur aus verschiedenen Elementen besteht.

Wegen $\tau + l(w(a,b,\tau)) = A^\tau(p_m)$ sind nur Wege, deren Ankunftszeiten die Schranke T, sogenannte T-Wege, außerdem nur solche, die vorgegebene Kosten C nicht überschreiten, sogenannte C-Wege von Interesse. Sei $W_T(a,b,\tau)$ bzw. $W_C(a,b,\tau)$ die Menge dieser T-Wege bzw. C-Wege Wichtig ist in diesem Zusammenhang die Menge $W_{CT}(a,b) = W_C(a,b,0) \cap W_T(a,b,0)$. Betrachtet werden folgende Problemstellungen:

I) Suche in $W_{CT}(a,b)$ einen Weg, der gleichzeitig zeit- und kostenoptimal ist.

II) Suche in $W_{CT}(a,b)$ einen zeitoptimalen Weg.

III) Suche in $W_{CT}(a,b)$ einen kostenoptimalen Weg.

Rechenverfahren

Betrachtet man das Problem

$$\max_{\mathfrak{B}^\tau} B^\tau(b) \tag{6}$$

mit

$$\mathfrak{B}^\tau = \{B^\tau(i), i \in P \mid B^\tau(a) = \tau,\ B^\tau(j) \leq \min_{(i,j) \in K} \min_{t \in T^\tau_i} \{l_{ij}(t) + t\}, j \in P - \{a\}\}$$

wobei

$T^\tau_i = \{t \in \mathbb{Z} \mid \exists\, w(a,i,\tau)$ mit $\tau + l(w(a,i,\tau)) = d \leq t$ und $[d,t) \cap V_i = \emptyset\}$ gilt, so hat man immer

$$\mathfrak{B}^\tau \neq \emptyset \tag{7}$$

$$\min_{W_T(a,b,\tau)} A^\tau(b) \geq \max_{\mathfrak{B}^\tau} B^\tau(b) \tag{8}$$

Man erkennt sofort, daß $B^\tau(i) = \tau, \forall i \in P$, eine $\mathfrak{B}$-zulässige Lösung ist. Zum Beweis von (8) (für $\tau = 0$) siehe man [10]. Der dort beschriebene Algorithmus zur Berechnung einer Optimallösung von (6) hat folgende Gestalt:

Seien α_i bzw. β_i die Mengen der durch den Algorithmus berechneten Ankunfts- bzw. Abfahrtszeiten für $i \in P$, sei $P_o \subset P$ die Menge der Punkte i, für die zeitoptimale Wege $w_{opt}(a,i,\tau)$ und früheste Ankunftszeiten $\tau + l(w_{opt}(a,i,\tau))$ bekannt sind, $I_i(P_o) = 0$, falls $i \in P_o$, $=1$, sonst.

Algorithmus I:

Schritt 0:

$P^1_o = \{a\}$, $(B^{\tau 1}(i) = \tau, \alpha^1_i = \emptyset, i \in P)$, $(\beta^1_i = \emptyset, i \in P - \{a\})$,
$\beta^1_a = \{[\tau, t] \cap \mathbb{Z} \mid t$ maximal mit $[\tau, t) \cap V_a = \emptyset\}$, $\gamma = \beta^1_a$,
$m^1(i,t) = \emptyset, i \in P, t \in [\tau, T] \cap \mathbb{Z}$, $s = a$, $h = 1$

Schritt h:

$$\left.\begin{aligned} &\delta_i = \{d \mid d = l_{si}(t) + t, t \in \gamma\} \\ &m^h(i,d) = m^h(i,d) \cup \{\langle s,t \rangle\}, d \in \delta_i, t \in \gamma \\ &\alpha^h_i = \alpha^h_i \cup \delta_i - \beta^h_i \end{aligned}\right\} \quad i \in P - \{s\}$$

$$\varepsilon^h = \min_{i \in P} \min_d \{d \in \alpha^h_i\} - \max_{j \in P} B^{\tau h}(j) \longrightarrow \begin{matrix}\min\\ \text{in}\end{matrix}\ i(h) \in P$$

$$B^{\tau(h+1)}(j) = B^{\tau h}(j) + I_j(P^h_o) \cdot \varepsilon^h$$

$$\max_{j \in P} B^{\tau(h+1)}(j) > T \ ? \longrightarrow \text{STOP}$$

$$P^{h+1}_o = P^h_o \cup \{i(h)\}$$

$$i(h) = b \ ? \longrightarrow \text{STOP}$$

$$\Gamma = \{d \in \mathbb{Z} \mid d \in \bigcup_{\text{disjunkt}} [t, \theta], t \in \alpha^h_{i(h)}, \theta \text{ maximal mit } [t,\theta) \cap V_{i(h)} = \emptyset\}$$

$$m^{h+1}(i,d) = \begin{cases} \bigcup_{\kappa \leq d} m^h(i,\kappa) & \kappa, d \in [t,\theta] \text{ def. über } \Gamma \\ m^h(i,d) & i \neq i(h) \end{cases}$$

$$\gamma = \Gamma - \beta^h_{i(h)}, \quad \alpha^{h+1}_i = \alpha^h_i$$

$$\beta^{h+1}_i = \begin{cases} \beta^h_i \cup \gamma & i = i(h) \\ \beta^h_i & \text{sonst} \end{cases}$$

$s = i(h)$, $h = h+1$, wiederhole Schritt h

In [10] wird gezeigt, daß das Verfahren eine Folge von $\mathfrak{B}^\tau$-zulässigen Lösungen mit nicht fallenden $B^{\tau h}(b)$- Werten berechnet, die nach endlich vielen Schritten entweder die Aussage $W_T(a,b,\tau) = \emptyset$ (falls $B^{\tau h}(b) > T$ gewählt werden kann) oder mit $B^\tau_{opt}(b) \leq T$ über einen Rückverfolgungsalgorithmus einen Weg $w_{opt}(a,b,\tau)$ mit minimaler Länge $l(w_{opt}(a,b,\tau))$ liefert, über den man, falls man sich zur Zeit τ in $a \in P$ befindet, $b \in P$ zur Zeit $\tau + l(w_{opt}(a,b,\tau))$ erreichen kann. Dabei bezeichnet $m(i,t)$ Markierungen, über die die Rückverfolgung zur Angabe eines optimales Weges möglich ist. Hat man $V_p = \emptyset$, $p \in P$, kann der optimale Weg in der Teilmenge der elementaren Wege gefunden werden. Der Algorithmus I benötigt dann maximal $|P|-1$ Schritte. Der Algorithmus wird im folgenden als Hilfsrechnung für die unter I), II), III) beschriebenen Probleme benutzt.

Bezeichne $w(a,b,\tau)^i$ für $i\in I_{w(a,b,\tau)}$ den Teilweg von $w(a,b,\tau)$, der von $a=p_o$ zum Zeitpunkt τ bis nach p_i (zum Zeitpunkt $A^{\tau}(p_i)$) führt. Ein Verbinden von Wegen $w(i,\rho,\nu)$, $w(\kappa,j,\mu)$ soll möglich sein, falls $\rho=\kappa$ und $\mu=A^{\nu}(\rho)$ gilt, man erhält $w(i,j,\nu)=w(i,\rho,\nu)\cup w(\rho,j,A^{\nu}(\rho))$ mit $l(w(i,j,\nu))=l(w(i,\rho,\nu))+l(w(\rho,j,A^{\nu}(\rho)))$ (entsprechendes gilt für die Kosten).

Bezeichnet $G(h)=(P(h),K(h))$ einen Graphen mit bestimmten Eigenschaften, so sei $G(h)[i,j,t]$ für $(i,j)\in K(h), t\in\mathcal{T}_o\subset\mathcal{T}$ derjenige Graph, der aus $G(h)$ durch Streichen der Verbindung $l_{ij}(t)$ entsteht, $h\in\mathbb{N}$. Man braucht zur besseren Unterscheidung noch spezielle Wege der Form $w(h)=(P_{w(h)},A_{w(h)},D_{w(h)})$ mit $P_{w(h)}=(p(h)=P_o,p_1,\dots,p_{m(h)}=b)$ und $A_{w(h)}=(A_h(p_i)\mid i\in I_{w(h)})$, $D_{w(h)}=(D_h(p_i)\mid i\in I_{w(h)}-\{m(h)\})$. Mit vorgegebenem C,T erhält man den

Algorithmus II:

Schritt 0:

$U=\emptyset$, $V=\emptyset$, $M=\emptyset$

Suche $w_o=w_{opt}(a,b,o)$ in $G(P,K,l,k)$ mittels Alg. I. Als Möglichkeiten ergeben sich

$B^o(b)>T$ wählbar $\longrightarrow W_T(a,b,o)=\emptyset \Rightarrow W_{CT}(a,b)=\emptyset$ (STOP)

$l(w_o)=T_o\leq T, k(w_o)>C \longrightarrow$ gehe zu 0.1

$l(w_o)=T_o\leq T, k(w_o)\leq C \longrightarrow w_o$ löst II) (evtl. STOP)

$U=\{w_o\}$, $V=U$

0.1: $w(1)=w_o$, $v(1)=w(1)^o$, $G(1)=G(P,K,l,k)$, $h=1$

Schritt h:

h.1: Suche $w_{hi}=w_{opt}(p_i,b,\rho), \rho=A_h(p_i)$, in $G(h)[p_i,p_{i+1},D_h(p_i)]$, $i\in I_{w(h)}-\{m(h)\}$ mittels Alg. I. Als Möglichkeiten ergeben sich

$B^{\rho}(b)>T$ wählbar $\longrightarrow$ gehe zu h.2

$B^{\rho}_{opt}(b)\leq T$, $w(a,b,o)=v(h)\cup w(h)^i\cup w_{hi}$

$M=M\cup\{w(a,b,c)\}$

h.2: $i+1\in I_{w(h)}-\{m(h)\} \longrightarrow$ setze $i=i+1$ gehe zu h.1

andernfalls prüfe

$M=\emptyset \longrightarrow$ STOP $\begin{cases} w\in U \text{ löst II)} \\ w\in V \text{ löst III)} \\ w\in V \text{ löst I), falls } l(w)=T_o \end{cases}$

$M\neq\emptyset \longrightarrow$ Bestimme $\hat{w}$ mit $l(\hat{w})=\min\limits_{w\in M}\{l(w)\}$

$\hat{w}=v(\hat{h})\cup w(\hat{h})^{\hat{i}}\cup w_{\hat{h}\hat{i}}, \hat{h}\leq h, \hat{i}\in I_{w(\hat{h})}-\{m(\hat{h})\}$

$k(\hat{w})>C \longrightarrow$ gehe zu h.3

$k(\hat{w})\leq C, U=\emptyset \longrightarrow U=\{\hat{w}\}$, $\hat{w}$ löst II) (evtl. STOP)

$V=U$ gehe zu h.3

$k(\hat{w})\leq C, U\neq\emptyset, k(\hat{w})\geq k(w), w\in V \longrightarrow$ gehe zu h.3

$k(w)\leq C, U\neq\emptyset, k(\hat{w})<k(w), w\in V \longrightarrow$ setze $V=\{\hat{w}\}$

h.3: $M=M-\{\hat{w}\}$

$G(h+1)=G(\hat{h})[p_{\hat{i}},p_{\hat{i}+1},D_{\hat{h}}(p_{\hat{i}})]$

$w(h+1)=w_{\hat{h}\hat{i}}, v(h+1)=v(\hat{h})\cup w(\hat{h})^{\hat{i}}$

$h=h+1$, wiederhole Schritt h

Algorithmus II basiert auf Überlegungen, wie sie auch in [1],[3],[6],[9],[13],[15] bis [18] benutzt werden. Dabei wird hier davon ausgegangen, daß das Branch and Bound- Verfahren entlang eines augenblicklich zeitoptimalen Weges weiter verzweigt. Bei Berücksichtigung von Aufenthaltskosten $k(A^{\tau}(i),D^{\tau}(i)-A^{\tau}(i))$ in einem Punkt $i\in P$ bzgl. eines Weges $w(a,b,\tau)$ müssen zusätzlich Verzweigungen unter Einbeziehung der Wartezeiten überprüft werden. Allgemeiner könnte man ein Überdeckungsproblem, wie es in [2] (und in den Literaturangaben zu [2]) beschrieben wird, anwenden. Vor allem, wenn mehrere augenblicklich zeitoptimale Wege existieren (mit den Markierungen $m(i,t)$ aus Algorithmus I kann man leicht die Menge aller augenblicklich zeitoptimalen Wege bestimmen) kann das von Vorteil sein. Welche rechentechnischen Auswirkungen sich hierbei ergeben wird noch untersucht.

Beispiel:

Für ein Beispiel einfachster Form bestehe der Graph aus den Punkten $P=\{a,p,q,b\}$. Seine Struktur sei gegeben durch

	(a,p)	(a,q)	(p,a)	(p,q)	(p,b)	(q,p)	(q,b)
0	3,2				4,2	3,3	3,5
1	1,4	3,2	2,3		4,1	4,3	4,5
2	1,3	5,1	1,4			2,4	3,5
3			3,1	4,6		3,1	3,5
4		2,3				2,3	
5	3,3			4,6		4,2	
6		3,2		2,8		1,4	
7		3,2		3,7	1,5		2,7
8	2,4						1,8
9					1,5		
10							

wobei der erste Wert die Länge, der zweite die Kosten für die entsprechende Kante bei Benutzung zum Zeitpunkt t beschreibt. Es gelte $T=10$. Als Verbotsintervalle seien festgelegt $V_a=[1,2]$, $V_p=[3,6]\cup[8,9]$, $V_q=[5,9]$, $V_b=\emptyset$.

Schritt 0 ergibt

$w_o=((a,p,a,q,p,b),(0,2,3,6,7,8),(1,2,4,6,7))$

Als weitere Wege gibt es nur noch

$w_I=((a,p,q,b),(0,3,7,9),(0,3,7))$

$w_{II}=((a,p,q,b),(0,2,7,9),(1,3,7))$

$w_{III}=((a,q,p,q,b),(0,4,6,8,9),(1,4,6,8))$

$w_{IV}=((a,q,p,b),(0,4,9,10),(1,5,9))$

Die Weglängen lassen sich sofort aus den entsprechenden Tupeln ablesen. Für die Kosten gilt $k(w_o)=20$, $k(w_I)=15$, $k(w_{II})=17$, $k(w_{II})=21$, $k(w_{IV})=9$. (Für $T=8$, $C\geq 20$ wäre w_o eindeutige Lösung für die Probleme I), II), III). Für $T<8$ oder $C<9$ existiert für keines der Probleme eine Lösung.)

[1] BELLMAN,R. and R.KALABA: On k-th Best Policies. J.SIAM 8 (1960) 582-588.
[2] CHRISTOFIDES,N. and S.KORMAN: A Computational Survey of Methods for the Set Covering Problem. Manag. Sc. 21 (1975) 591-599.
[3] CLARKE,S.,KRIKORIAN,A. and J.RAUSEN: Computing the n Best Looples Paths in a Network. J.SIAM 11(1963) 1o96-11o2.
[4] COOKE,K.L. and E.HALSEY: The Shortest Route through a Network with Time-Dependent Internodal Transit Times. J.Math. Anal. and Appl. 14(1966) 493-498.
[5] DIJKSTRA,E.W.:A Note on Two Problems in Connexion with Graphs. Num.Math. 1(1959) 269-271.
[6] DREYFUS,S.E.: An Appraisal of Some Shortest-Path Algorithms. J.ORSA 17(1969) 395-412.
[7] FORD,L.R. and D.R.FULKERSON: Constructing Maximal Dynamic Flows from Static Flows. J.ORSA 6(1958) 419-433.
[8] FORD,L.R. and D.R.FULKERSON: Flows in Networks. Princeton University Press, Princeton N.Y. (1962).
[9] FOX,B.L.: Calculating k th Shortest Paths. INFOR 11(1973) 66-70.
[10] GAUL,W.: Zur Lösung von Wegproblemen mit zeitabhängigen Beschränkungen. Erscheint in ZAMM.
[11] GAUL,W.: On the Maximum Capacity-Route Problem with Time-Dependent Constraints. Submitted.
[12] HALPERN,J. and I.PRIESS: Shortest Path with Time Constraints on Movement and Parking. Networks 4(1974) 241-253.
[13] HOFFMAN,W. and R.PAVLEY: A Method for the Solution of the n Best Path Problem. J.ACM 6(1959) 506-514.
[14] KLAFSZKY,E.: Determination of Shortest Path in a Network with Time-Dependent Edge-Lengths. Math.Operationsforsch. u. Statist. 3(1972) 255-257.
[15] LAWLER,E.L.: A Procedure for Computing the k Best Solutions to Discrete Optimization Problems and its Application to the Shortest Path Problem. Manag.Sc. 18(1972) 401-405.
[16] POLLACK,M.: Solutions of the k th Best Route through a Network - A Review. J.Math.Anal. and Appl. 3(1961) 547-559.
[17] WEIGAND,M.M.: Ein leistungsfähiger Algorithmus zur Bestimmung von k-kürzesten Wegen in einem Graphen. Computing 12(1974) 273-283.
[18] YEN,J.Y.: Finding the k Shortest Loopless Paths in a Network. Manag.Sc. 17(1971) 712-716.

Verfasser:

Dr. Wolfgang Gaul
wiss. Assistent
Institut Angewandte Mathematik
Universität Bonn, Bonn/BRD