Zeit- und Kostenoptimierung von Bottleneckwegen

W. Gaul und J. Hartung, Bonn

Zusammenfassung: Es wird die Einbeziehung mehrfacher Bewertungen bei Wegproblemen behandelt, wobei eine zeitabhängige Graphenstruktur berücksichtigt wird. Insbesondere sind neben zeitoptimalen Wegen mit zeitabhängigen Kantenbenutzungsdauern Wege von Interesse, deren maximale Kantenbenutzungsdauer Einschränkungen unterliegt.

Unter zusätzlichen zeitabhängigen Kostenüberlegungen werden Rechenverfahren angegeben, die (funktional-) effiziente Lösungen (im Sinne der Vektoroptimierung) unter Nebenbedingungen bestimmen können. Ein Beispiel ist zur Verdeutlichung beigelegt.

Einleitung

Eine Vielzahl von Problemformulierungen aus dem Operations Research Gebiet erlauben die Anwendung graphentheoretischer Überlegungen. Dabei vergrößert eine Berücksichtigung zeitabhängiger Änderungsmöglichkeiten bei der Modellwahl natürlich den Realitätsbezug. Als erste haben FORD/FULKERSON [4,1958] für Flußprobleme dynamische Aspekte durch Einbeziehung von Benutzungsdauern für die Verbindungen im Flußgraphen betrachtet. Wegprobleme mit vom Benutzungszeitpunkt abhängigen Benutzungsdauern haben zuerst COOKE/HALSEY [1,1966] mittels dynamischer Optimierung formuliert, weitere Ansätze DREYFUS [3,1969] mittels Modifikation des DIJKSTRA [2]-Verfahrens und KLAFSZKY [11,1972]. Die Berücksichtigung zusätzlicher zeitabhängiger Benutzungsverbote für Wegprobleme findet man bei GAUL [6,1975], HALPERN [8,1977] und HALPERN/PRIESS [9,1974].

In der vorliegenden Arbeit wird für zeitabhängige Wegformulierungen eine Berücksichtigung mehrerer Zielkriterien angestrebt. Dabei ist auf eine Ausnutzung des Konzepts der Ersatzprogramme (siehe z.B. HARTUNG [10,1974], KOSMOL [12,1973]) verzichtet worden, (obwohl Algorithmus II eine solche Formulierung zuließe), weil die für eine dann erforderliche Transformation der verschiedenen Bewertungskriterien (z. B. Kosten des Weges, Zeitdauer des Weges) auf ein gemeinsames Bewertungskriterium benötigte Information (Vorbewertungen) stark problemabhängig und häufig a priori nicht erhältlich ist. Unter Berücksichtigung einer sinnvollen Vorgabe von Schranken für die Einzelbewertungen wird deshalb die Angabe aller ((funktional-)effizienter) Lösungen in dem durch die Schranken definierten Kegel ermöglicht, wobei der Entscheidungsträger aus

dieser Lösungsmenge dann diejenige auswählen kann, die seiner (aufgrund der Kenntnis der Lösungsmenge a posteriori möglicherweise besser erhältlichen) Präferenzstruktur entspricht. Ein Beispiel ist zur Verdeutlichung beigefügt.

Problemformulierung

Sei $G(P, K, l, c)$ ein schlichter, zusammenhängender, gerichteter, endlicher Graph mit der Punktmenge P und der Kantenmenge K . Für $T = \{0, 1, \dots, T\}$ (T Beobachtungsgrenze) seien $l: K \times T \rightarrow R_+$ bzw. $c: K \times T \rightarrow R_+$ Bewertungen, deren (positive, ganzzahlige) Werte $l_{ij}(t)$ bzw. $c_{ij}(t)$ die Benutzungsdauer bzw. die Benutzungskosten für die Kante $(ij) \in K$ in Abhängigkeit vom Benutzungsbeginn $t \in T$ im Punkt $i \in P$ bestimmen (Ist eine Benutzung zum Zeitpunkt t nicht möglich, gelte $l_{ij}(t) = c_{ij}(t) = \infty$). Für $a, b \in P, \tau \in T$, kann ein Weg von a nach b , der Zeit $\tau \in T$ in a beginnt, eindeutig beschrieben werden durch

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

wobei

$$P_{w(a, b, \tau)} = (a = p_0, p_1, \dots, p_m = b), p_i \in P, i \in I_{w(a, b, \tau)} = \{0, 1, \dots, m\}$$

die Folge der dabei (in dieser Anordnung) durchlaufenen Punkte,

$$A_{w(a, b, \tau)} = (A^T(p_0), \dots, A^T(p_m))$$

$$D_{w(a, b, \tau)} = (D^T(p_0), \dots, D^T(p_{m-1})), D^T(p_i) \in T,$$

die dabei realisierten Ankunfts- bzw. Startzeiten für die Punktfolge $P_{w(a, b, \tau)}$ angibt, für die natürlich

$$(1) D^T(p_i) \geq A^T(p_i) \quad i \in I_{w(a, b, \tau)} \setminus \{m\}$$

$$(2) A^T(p_0) = \tau$$

$$(3) A^T(p_m) \leq T$$

$$(4) A^T(p_i) = D^T(p_{i-1}) + l_{p_{i-1}p_i}(D^T(p_{i-1})) \quad i \in I_{w(a, b, \tau)} - \{0\}$$

gelten muß. Die Differenz $D^T(p_i) - A^T(p_i)$ beschreibt die Wartezeit in p_i . Falls im Punkt $p \in P$ Warteverbote bestehen während der Zeitintervalle

$$V_p = \bigcup_{k=1}^{n(p)} [t_{1k}^p, t_{2k}^p], t_{1k}^p \leq t_{2k}^p < t_{1(k+1)}^p \quad \text{falls } n(p) > 1$$

so hat man, falls für einen Weg $w(a, b, \tau)$ die Beziehung $p_i = p$ erfüllt ist, zusätzlich zu beachten

$$(5) D^T(p) = A^T(p) \quad \text{falls} \quad A^T(p) \in V_p$$

$$(6) D^T(p) \leq t_{1(k+1)}^p \quad t_{2k}^p < A^T(p) < t_{1(k+1)}^p$$

Es interessieren nun Wege, für die die Bewertungen

$$(7) l^1(w(a,b,\tau)) = \max_{i \in I_{w(a,b,\tau)}} \{ \sum_{\{m\} p_i p_{i+1}} 1 (D^\tau(p_i)) \}$$

$$(8) l^2(w(a,b,\tau)) = \sum_{i \in I_{w(a,b,\tau)}} \sum_{\{m\} p_i p_{i+1}} 1 (D^\tau(p_i)) + D^\tau(p_i) - A^\tau(p_i)$$

$$(9) c(w(a,b,\tau)) = \sum_{i \in I_{w(a,b,\tau)}} \sum_{\{m\} p_i p_{i+1}} c_{p_i p_{i+1}} (D^\tau(p_i)) + \tilde{c}_i(A^\tau(p_i), D^\tau(p_i) - A^\tau(p_i))$$

minimal werden, bzw. gewissen Schranken L^1, L^2, τ, C genügen, sogenannte L^1, L^2 - bzw. C-Wege. Bei der hier benutzten Problemformulierung gilt sinnvollerweise $L^1 \leq L^2 \leq \tau$, aus (4) und (8) folgt außerdem für L^2 -Wege

$$(10) l^2(w(a,b,\tau)) = A^\tau(b) - \tau \leq L^2 - \tau$$

Mit $\tilde{c}_i(\dots)$ kann man Wartezeitkosten in p_i berücksichtigen.

L^1 -minimale Wege seien als Bottleneckwege bezeichnet (weil "große" Kantenbenutzungsdauern Transporthindernisse darstellen können, siehe auch [5]), sie sind von Interesse, wenn die Benutzungsdauer von Verbindungen zwischen zwei Punkten einschränkenden Bedingungen genügen muß. Daneben ist natürlich eine Optimierung der Bewertungen l^2 und c wünschenswert. Dabei wird man sich manchmal mit

- I) l^1 -optimalen L^2 -C-Wegen
- II) l^2 -optimalen L^1 -C-Wegen
- III) c -optimalen L^1 - L^2 -Wegen

begnügen (Man beachte: Die Lösungen von I, II, III brauchen nicht effizient zu sein.).

Im folgenden wird es von Vorteil sein, mehrere Wege zu einem Weg zu verbinden.

Aus $w(i, \rho, \nu)$, $w(x, j, \mu)$ erhalte man

$$(11) w(i, j, \nu) = w(i, \rho, \nu) \cup w(x, j, \mu) \quad \text{falls } \rho=x \text{ und } A^\nu(\rho)=\mu$$

Weiter bezeichne $w(a,b,\tau)^i$ [bzw. $w(a,b,\tau)_i$] für $i \in I_{w(a,b,\tau)}$ den Teilweg von $w(a,b,\tau)$, der von $a=p_0$ (zum Zeitpunkt τ) bis nach p_i (zum Zeitpunkt $A^\tau(p_i)$) [bzw. von p_i (zum Zeitpunkt $A^\tau(p_i)$) bis nach b (zum Zeitpunkt $A^\tau(b)$)] führt.

Rechenverfahren

Bezeichne $W(a,b,\tau)$ die Menge der Wege $w(a,b,\tau)$ in $G(P,K,l,c)$, hinzugefügte nach unten gestellte Schranken -Indices die Teilmengen vom entsprechenden Beschränkungstyp, speziell sei $W_{L^2}(a,b,\tau)$ die Menge der L^2 -Wege in $G(P,K,l,c)$.

Zuerst soll die Frage nach der Existenz solcher Wege untersucht werden.

Wegen der durch die Berücksichtigung dynamischer Aspekte bedingten aufwendigeren Anschrift für Wege ist es sinnvoll, das folgende Problem zu betrachten

$$(12) \max_{L^\tau} B^\tau(b)$$

mit

$$\mathcal{A}^\tau := \{B^\tau(i), i \in P \mid B^\tau(a) = \tau, B^\tau(j) \leq \min_{(i,j) \in K} \min_{t \in T_i} \{l_{ij}(t) + t\}, j \in P - \{a\}\}$$

wobei

$T_i^\tau := \{t \in \mathbb{R}^+ \mid \exists w(a, i, \tau) \text{ mit } \tau + l^2(w(a, i, \tau)) = d \leq t \text{ und } [d, t) \cap V_i = \emptyset\}$ die Menge der Zeitpunkte angibt, in denen eine Ankunft (bzw. ein Aufenthalt) im Punkt i nach einem Start von Punkt a zum Zeitpunkt τ möglich ist. Es gilt

Lemma 1:

- (i) $\mathcal{A}^\tau \neq \emptyset$
- (ii) $\min_{W_{L^2}(a, b, \tau)} A^\tau(b) \geq \max_{\mathcal{A}^\tau} B^\tau(b)$
- (iii) $B_{\max}^\tau(b) > L^2 \implies W_{L^2}(a, b, \tau) = \emptyset$
- (iv) $B_{\max}^\tau(b) \leq L^2 \implies \exists w_0(a, b, \tau) \in W_{L^2}(a, b, \tau)$ mit $A_0^\tau(b) = B_{\max}^\tau(b)$

Beweis:

- (i) Wegen $l_{ij}(t) \in \mathbb{R}_+$ kann man $B^\tau(i) = \tau$ für alle $i \in P$ wählen.
- (ii) $W_{L^2}(a, b, \tau) = \emptyset$ ergibt $\min_{W_{L^2}(a, b, \tau)} A^\tau(b) = \infty \geq \max_{\mathcal{A}^\tau} B^\tau(b)$,
andernfalls gilt für $w(a, b, \tau) \in W_{L^2}(a, b, \tau)$ und $B^\tau \in \mathcal{A}^\tau$ (wegen (4) und (12))
 $A^\tau(b) = A^\tau(p_m) = l_{p_{m-1}p_m}(D^\tau(p_{m-1})) + D^\tau(p_{m-1}) \geq$
 $\geq \min_{t \in T_{p_{m-1}}^\tau} \{l_{p_{m-1}p_m}(t) + t\} \geq \min_{(i, p_m) \in K} \min_{t \in T_i^\tau} \{l_{ip_m}(t) + t\} \geq B^\tau(p_m) = B^\tau(b)$
- (iii) $B_{\max}^\tau(b) > L^2$ und $W_{L^2}(a, b, \tau) \neq \emptyset$ ergibt wegen (ii) die Beziehung $A^\tau(b) > L^2$ im Widerspruch zu (10).
- (iv) Sei $P_p = \{i \in P \mid (i, p) \in K\}$ für $p \in P$. Wegen $B_{\max}^\tau \in \mathcal{A}^\tau$ hat man
 $B_{\max}^\tau(b) \leq \min_{(i, b) \in K} \min_{t \in T_i^\tau} \{l_{ib}(t) + t\}$. Es muß gelten: Es gibt $i_0 \in P_b$ und
 $t_0 \in T_{i_0}^\tau$ mit $B_{\max}^\tau(b) = l_{i_0 b}(t_0) + t_0 (\leq L^2)$. Anderfalls wäre
 $B_{\max}^\tau(b) > L^2$ wählbar. Dann gibt es aber $w(a, i_0, \tau)$ mit
 $l^2(w(a, i_0, \tau)) + \tau = d \leq t_0$ und deshalb
 $w_0(a, b, \tau) = w(a, i_0, \tau) \cup ((i_0, b), (d, B_{\max}^\tau(b)), (t_0))$

Man braucht also einen Algorithmus zur Lösung des Problems (12). Wegen Lemma 1 wird dadurch nicht nur die Frage nach der Existenz eines L^2 -Weges beantwort-

tet, sondern darüberhinaus, falls $W_{L^2}(a,b,\tau) \neq \emptyset$ gilt, ein zeitoptimaler L^2 -Weg gefunden. Ein solches Verfahren ist (für den Spezialfall $\tau=0$, der eine Verallgemeinerung auf die hier betrachtete Situation zulässt) in [6] und [7] beschrieben. Brauchen keine Warteverbote berücksichtigt zu werden ($V_i = \emptyset$ für $i \in P$), so können nach $|P|-1$ Schritten als Lösungen sogar zyklfreie Lösungen gefunden werden. (Siehe aber auch [1], [3], [8], [9], [11]). Die Kenntnis eines solchen Verfahrens, kurz Basisverfahren genannt, wird für die folgenden Überlegungen vorausgesetzt. Erfüllt ein so gefundener Weg die Schranken L^1 und C , hat man schon eine Lösung von Problem II.

Sei nun $G_{L^1}(P,K,l,c)$ der Graph, der aus $G(P,K,l,c)$ durch Streichen der Verbindungen $l_{ij}(t) > L^1$ ($(i,j) \in K, t \in T$) entsteht. Die Überprüfung von $W_{L^1}(a,b,\tau) \neq \emptyset$ (man setzt $L^2=T$) ermöglicht der

Algorithmus I:

Schritt 0: Setze $L(0)=0$, $N=\emptyset$.

Suche $w_0 = w_{\text{opt}}(a,b,\tau)$ im $G_{L^1-L(0)}(P,K,l,c)$ mittels Basis-Alg.

$$B_{\max}^T(b) > T \rightarrow W_{L^1}(a,b,\tau) = \emptyset \rightarrow \text{STOP}$$

$$N = \{w_0, l^1(w_0)\}$$

$$L(1) = L^1 - (l^1(w_0) - 1), \quad h=1$$

Schritt h:

Suche $w_h = w_{\text{opt}}(a,b,\tau)$ in $G_{L^1-L(h)}(P,K,l,c)$ mittels Basis-Alg.

$$B_{\max}^T(b) > T \rightarrow \text{STOP} \quad (N \text{ enthält Lösung})$$

$$N = \{w_h, l^1(w_h)\}$$

$$L(h+1) = L^1 - (l^1(w_h) - 1), \quad h=h+1, \quad \text{gehe zu Schritt h.}$$

Endet Alg. I nicht schon im 0-ten Schritt, gilt $W_{L^1}(a,b,\tau) \neq \emptyset$ und man kann einen l^1 -optimalen (Bottleneck-) Weg finden.

Erfüllt ein so gefundener Weg die Schranken L^2, C , hat man schon eine Lösung von Problem I.

Zur Einbeziehung von Kostengesichtspunkten benutzen wir zur besseren Unterscheidung für Wege $w(h)$ (h Iterationsindex) die Bezeichnungswiese

$$w(h) = (P_{w(h)}, A_{w(h)}, D_{w(h)}) \quad \text{mit } P_{w(h)} = (p_0, p_1, \dots, p_{m(h)}), \quad I_{w(h)} = \{0, 1, \dots, m(h)\}$$

$$A_{w(h)} = (A_h(p_i), i \in I_{w(h)}), \quad D_{w(h)} = (D_h(p_i), i \in I_{w(h)} \setminus \{m(h)\}).$$

Sei mit $G[i,j,t]$ derjenige Graph bezeichnet, der aus einem Graphen $G(P,K,l,c)$

durch Streichen der Verbindung $l_{ij}(t)$ ($(i,j) \in K, t \in \bar{T}$) entsteht. Die Angabe von kostenoptimalen Lösungen (man setze $L^2=T$) ermöglicht der

Algorithmus II:

Schritt 0: Setze $U=\emptyset, M=\emptyset$.

Suche $\hat{w}_0 = w_{\text{opt}}(a,b,\tau)$ im $G(P,K,L,c)$ mittels Basis-Alg.

$B_{\text{max}}^T(b) > T \rightarrow W_C(a,b,\tau) = \emptyset \rightarrow \text{STOP}$

$c(\hat{w}_0) > C \rightarrow$ gehe zu 0.1

$U = \{\hat{w}_0\}$

0.1: $w(1) = \hat{w}_0, v(1) = w(1)^0, G(1) = G(P,K,L,c), h=1$

Schritt h: Setze $i=0$.

h.1.:

Suche $w_{hi} = w_{\text{opt}}(p_i, b, \rho), \rho = A_h(p_i)$, im $G(h) [p_i, p_{i+1}, D_h(p_i)]$, $p_i, p_{i+1} \in P_{w(h)}$,

mittels Basis-Alg.

$B_{\text{max}}^D(b) > T \rightarrow$ gehe zu h.2

$w(a,b,\tau) = v(h) \cup w(h)^i \cup w_{hi}$

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

h.2: $i+1 \in I_{w(h)} \setminus \{m(h)\} \rightarrow i=i+1$, gehe zu h.1

Prüfe $M, M=\emptyset \rightarrow \text{STOP}$ (U enthält Lösungsantwort)

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

mittels $l^2(\hat{w}_h) = \min_{w \in M} l^2(w)$

$c(\hat{w}_h) > C \rightarrow$ gehe zu h.3

$U = \emptyset \rightarrow U = \{\hat{w}_h\}$, gehe zu h.3

$U \neq \emptyset, c(\hat{w}_h) < c(w), w \in U \rightarrow U = \{\hat{w}_h\}$

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

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

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

$h=h+1$, gehe zu Schritt h

Für Algorithmus II in der hier angegebenen Form ist $\tilde{c}_1^T(A^T(p_i), D^T(p_i) - A^T(p_i)) \equiv 0$ (siehe (9)) vorausgesetzt worden, um eine aufwendige Zyklenüberprüfung bei Vorliegen von Wartezeiten zu vermeiden (Es bleiben dann dadurch höchstens Lö-

sungen unberücksichtigt, für die man schon "nicht schlechtere" kennt.) Endet Algorithmus II nicht schon im 0-ten Schritt, kann man, falls $U \neq \emptyset$ gilt, einen c-optimalen Weg finden. Erfüllt ein so gefundener Weg die Schranken L^1, L^2 (Dies läßt sich immer erreichen, wenn man T durch L^2 und $G(P, K, l, c)$ im 0-ten Schritt durch $G_{\perp 1}(P, K, l, c)$ ersetzt.), hat man eine Lösung von Problem III. Algorithmus II modifiziert Überlegungen zur Berechnung k-kürzester Wege (siehe z.B. LAWLER [13, 1976]) unter Berücksichtigung dynamischer Gesichtspunkte. Man überlege:

Sei $M(h)$ die im h-ten Schritt berechnete Menge M , bevor Operation h.3 durchgeführt wird, sei $W(w(\rho), w(\rho)^i, G(\rho) [P_i, P_{i+1}, D_\rho(P_i)])$ die Menge der Wege der Gestalt $w(a, b, \tau)$ in $G(\rho) [P_i, P_{i+1}, D_\rho(P_i)]$, $P_i, P_{i+1} \in P_{W(\rho)}$, die mit $v(\rho) \cup w(\rho)^i$, $i \in I_{W(\rho)} \setminus \{m(\rho)\}$, beginnen. Für $W_h(\rho, i)$, $\rho \leq h$ ($h \geq 1$ Iterationsindex), $i \in I_{W(\rho)} \setminus \{m(\rho)\}$, mit $W_1(1, i) = W(v(1), w(1)^i, G(1) [P_1, P_{i+1}, D_1(P_1)])$ und

$$(13) \quad W_{h+1}(\rho, i) = \begin{cases} W_h(\rho, i) & \rho \leq h, \rho \neq j(h), i \neq i(h) \\ \emptyset & \rho = j(h), i = i(h) \\ W(v(h+1), w(h+1)^i, G(h+1) [P_i, P_{i+1}, D_{h+1}(P_i)]), & \begin{matrix} \rho = h+1 \\ i \in I_{W(h+1)} \setminus \{m(h+1)\} \end{matrix} \end{cases}$$

und

$$(14) \quad M_h = \bigcup_{\rho \leq h} \bigcup_{i \in I_{W(\rho)} \setminus \{m(\rho)\}} W_h(\rho, i)$$

gilt

Lemma 2:

- (i) $W_h(\rho_1, i_1) \cap W_h(\rho_2, i_2) = \emptyset$ für $(\rho_1, i_1) \neq (\rho_2, i_2)$
- (ii) $M(h) \subset M_h$
- (iii) $M_h \cap \bigcup_{v=0}^{h-1} \{\hat{w}_v\} = \emptyset$

Beweis:

(i) Man sieht unmittelbar

$$(15) \quad W_h(\rho, i_1) \cap W_h(\rho, i_2) = \emptyset \quad \text{für } i_1 \neq i_2$$

Für $h=1$ liefert (15) schon (i). Unter Annahme der Gültigkeit für h braucht man für $h+1$ wegen (13) und (15) nur z.B. $W_{h+1}(h+1, i)$ und $W_{h+1}(\rho, j)$ für $\rho \leq h$, $\rho \neq j(h)$, $j \neq i(h)$ zu überprüfen. In diesem Fall gilt aber $W_{h+1}(\rho, j) = W_h(\rho, j)$ und $W_{h+1}(h+1, i) \subset W_h(j(h), i(h))$.

(ii) Für $h=1$ gilt $M(1) = \bigcup_{i \in I_{w(1)} \setminus \{m(1)\}} \{v(1) \cup w(1) \overset{i}{\cup} w_{1i}\} \subset \bigcup_{i \in I_{w(1)} \setminus \{m(1)\}} W_1(1, i) = M_1$.

Aus der angenommenen Gültigkeit von (ii) für h folgt

$M(h) \setminus \{\hat{w}_h\} \subset M_h \setminus \{w_h(j(h), i(h))\}$, denn wegen (i) gilt

$$w \in M(h) \setminus \{\hat{w}_h\} \implies w \notin w_h(j(h), i(h))$$

Dann folgt für $h+1$ aber $M(h+1) = M(h) \setminus \{\hat{w}_h\} \cup \bigcup_{i \in I_{w(h+1)} \setminus \{m(h+1)\}} \{v(h+1) \cup w(h+1) \overset{i}{\cup} w_{h+1i}\} \subset$

$$\subset M_h \setminus \{w_h(j(h), i(h))\} \cup \bigcup_{i \in I_{w(h+1)} \setminus \{m(h+1)\}} \{w_{h+1}(h+1, i)\} = M_{h+1}.$$

(iii) Man hat $M_1 \cap \{\hat{w}_0\} = \emptyset$. Bei angenommener Gültigkeit von (iii) für h erhält man für $h+1$ mit der entsprechenden Aufspaltung wie unter (ii)

$$(M_h \setminus \{w_h(j(h), i(h))\} \cup \bigcup_{i \in I_{w(h+1)} \setminus \{m(h+1)\}} \{w_{h+1}(h+1, i)\}) \cap (\bigcup_{\mu=0}^{h-1} \{\hat{w}_\mu\} \setminus \{\hat{w}_h\}) = \emptyset$$

wenn man die entsprechenden Überlegungen für die einzelnen Durchschnittsterme beachtet.

Da wegen der eingangs angeschriebenen Voraussetzungen über die Struktur der betrachteten Graphen nur endlich viele Wege zur Verfügung stehen, ist Algorithmus II wegen Lemma 2 endlich. Dabei ist eine Reduktion der Schranken L^1, L^2 bzw. ein Abbruch des Verfahrens nach Kenntnis einer suboptimalen Lösung, die einer geeigneten Kostenschranke genügt, empfehlenswert. Bei der Anschrift von Algorithmus II ist auf die Einbeziehung eines Hilfsverfahrens zur Markierung effizienter Punkte verzichtet werden, weil der Algorithmus in der vorliegenden Form prinzipiell auch auf ein Ersatzprogramm anwendbar wäre (bei Kenntnis der transformierten Kostenvorbewertung). Entsprechende (Effizienz-) Markierungen können jedoch leicht berücksichtigt werden (z.B. durch komponentenweisen Vergleich). Zur Verdeutlichung dient das folgende

Beispiel:

Für ein Beispiel einfachster Form bestehe der Graph $G(P, K, l, c)$ aus den Punkten $P = \{a, p, q, b\}$ und den Kanten $K = \{(a, p), (a, q), (p, a), (p, q), (p, b), (q, p), (q, b)\}$. Sei $T = 10$. Die zeitabhängige Entfernungs- und Kostenstruktur sei gegeben durch die folgende Tabelle, wobei der erste Wert die Benutzungsdauer $l_{ij}(t)$, der zweite die Kosten $c_{ij}(t)$ für die entsprechende Kante $(i, j) \in K$ bei Benutzung zum Zeitpunkt t beschreibt (Für fehlende Wertepaare existieren keine Verbindungen).

Kanten	(a,p)	(a,q)	(p,a)	(p,a)	(p,b)	(q,p)	(q,b)
Zeit t	0	3,2			<u>4,2</u>	3,3	3,5
	1	1,4	3,2	2,3	<u>4,1</u>	<u>4,3</u>	<u>4,5</u>
	2	1,3	<u>5,1</u>	1,4		2,4	3,5
	3			3,1	<u>4,6</u>	3,1	3,5
	4		2,3			2,3	
	5	3,3			<u>4,6</u>	<u>4,2</u>	
	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						

dungen.). Als Verbotsintervalle für Wartezeiten in den Punkten seien $(V_a = [1,2], V_p = [3,6] \cup [8,9], V_q = [5,9], V_b = \emptyset)$ festgelegt.

Bei Vorgabe genügend großer Schranken $L^1 > 4, L^2 = T, C > 21$ findet der Algorithmus II die Lösungen (geordnet nach den Ankunftszeiten, die aus dem letzten Wert des zweiten Tupels abzulesen sind)

$w_1 = ((a,p,a,q,p,b), (0,2,3,6,7,8), (1,2,4,6,7)), l^1(w_1) = 2, l^2(w_1) = 8, c(w_1) = 20$

$w_2 = ((a,p,q,b), (0,3,7,9), (0,3,7))$

$l^1(w_2) = 4, l^2(w_2) = 9, c(w_2) = 15$

$w_3 = ((a,p,q,b), (0,2,7,9), (1,3,7))$

$l^1(w_3) = 4, l^2(w_3) = 9, c(w_3) = 17$

$w_4 = ((a,q,p,q,b), (0,4,6,8,9), (1,4,6,8))$

$l^1(w_4) = 3, l^2(w_4) = 9, c(w_4) = 21$

$w_5 = ((a,q,p,b), (0,4,9,10), (1,5,9))$

$l^1(w_5) = 4, l^2(w_5) = 10, c(w_5) = 9$

Aufgrund der Algorithmusstruktur wirkt sich vor allem eine Reduktion der Schranken L^1, L^2 rechenaufwandverringend aus.

Wählt man $L^1 = 3$ (und $L^2 = T, C > 21$), findet der Algorithmus (die in der Graphenstrukturtablelle unterstrichenen Werte fallen dann heraus) in 2 Schritten die Wege w_1 und w_4 . w_4 wird als nicht effizient verworfen, w_1 ist in diesem Fall perfekte Lösung.

Wählt man $L^1 > 4, L^2 = 9, C = 20$ erhält man w_1, w_2, w_3 . w_3 wird als nicht effizient verworfen, w_1 löst I und II, w_2 löst III.

Wählt man $9 < C < 15$ (und $L^1 = 4, L^2 = T$) hätte der Algorithmus trotzdem auch $w_1, \dots, w_4$ zu berechnen (und als nicht zulässig zu verwerfen). w_5 ist in diesem Fall perfekte Lösung.

Literatur:

- [1] COOKE, K.L., HALSEY, E.: The shortest route through a network with time-dependent internodal transit times, *J.Math. Analysis Appl.* 14 (1966) 493-498.
- [2] DIJKSTRA, E.W.: A note on two problems in connexion with graphs, *Num.Math.* 1 (1959) 269-271.
- [3] DREYFUS, S.E.: An appraisal of some shortest-path algorithms, *J.ORSA* 17(1969) 395-412.
- [4] FORD, L.R., Fulkerson, D.R.: Constructing maximal dynamic flows from static flows, *J.ORSA* 6(1958) 419-433.
- [5] GARFINKEL, R.S., RAO, M.R.: The bottleneck transportation problem, *NRLQ* 18(1971) 465-472.
- [6] GAUL, W.: On constrained shortest route problems, *Lect. Notes in Economics and Math. Systems* 117 (1975) 83-91.
- [7] GAUL, W.: Zur Lösung von Wegproblemen mit zeitabhängigen Beschränkungen, *ZAMM* 57(1977) T323-T325.
- [8] HALPERN, J.: Shortest route with time-dependent length of edges and limited delay possibilities in nodes, *ZOR* 21 (1977) 117-124.
- [9] HALPERN, J., PRIESS, I.: Shortest paths with time constraints on movement and parking, *Networks* 4(1974) 241-253.
- [10] HARTUNG, J.: Über Ersatzprogramme bei der Vektoroptimierung, *ORV* 18(1974) 129-133.
- [11] KLAFSZKY, E.: Determination of shortest path in a network with time-dependent edge-length, *Math. Operationsforschung und Statistik* 3(1972) 255-257.
- [12] KOSMOL, P.: Zur Vektorwertigen Optimierung, *ORV* 15(1973) 77-84.
- [13] LAWLER, E.L.: *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston (1976).
- [14] ZELENY, M.: *Linear Multiobjective Programming*, *Lect. Notes in Economics and Math. Systems* 95(1974).