

R. E. Burkard, G. Róte, G. Ruhe, N. Sieber

Algorithmische Untersuchungen zu bikriteriellen kostenminimalen Flüssen in Netzwerken

1. Einleitung

Die Bestimmung optimaler Flüsse auf Netzwerken gehört zu den am häufigsten angewandten Aufgaben des Operations Research. Die wesentlichen Ursachen dafür liegen in dem hohen Veranschaulichungsgrad der Modelle vermittelt der Darstellung der physischen Größen und deren Abhängigkeiten durch einen Graphen und im Vorhandensein von effektiven Lösungsalgorithmen begründet. Mathematisch handelt es sich dabei um lineare Optimierungsaufgaben, deren Nebenbedingungen wie beim Maximalstromproblem oder beim Problem eines kostenminimalen Flusses besonders einfach und übersichtlich strukturiert sind. Mit wenigen Ausnahmen, z. B. *Klingman* und *Mote* (1982), wird bei den bisherigen Untersuchungen jedoch stets von nur einer Zielfunktion ausgegangen. Viele praktische Probleme beziehen indessen mehrere, miteinander konkurrierende Zielgrößen ein. Die Zielstellung der vorliegenden Arbeit besteht in der Darstellung der beim Entwurf und der Analyse von Algorithmen für das bikriterielle kostenminimale Flußmodell erzielten Ergebnisse. Im zweiten Abschnitt geben wir die formale mathematische Beschreibung der Problemstellung. Die im dritten Abschnitt dargestellten Komplexitätsergebnisse bilden den Ausgangspunkt für die nachfolgend beschriebenen Algorithmentwicklungen. Der vierte Abschnitt behandelt dazu zunächst einige allgemeine Resultate zur Approximation konvexer Funktionen. Weiterhin werden ein exakter Lösungsalgorithmus und verschiedene Realisierungen des Sandwich-Approximationsverfahrens beschrieben. In einem fünften Abschnitt werden numerische Resultate von Testrechnungen zu den oben angegebenen Algorithmen dargestellt.

2. Problemstellung

Es bezeichnet $G = (V, A)$ einen gerichteten, asymmetrischen zusammenhängenden Graph mit der Knotenmenge V und der Bogenmenge $A \subseteq V \times V$. Die Anzahl der Knoten sei $|V| = n$ und die Anzahl der Bögen $|A| = m$. In G sind zwei ausgezeichnete Knoten 1 und n gegeben, die die Quelle bzw. die Senke symbolisieren. Sei R die Menge der reellen Zahlen. Eine auf A erklärte Funktion $x : A \rightarrow R$ heißt ein Fluß, falls

$$\sum_{(i,k) \in A} x_{jk} - \sum_{(i,j) \in A} x_{ij} = \begin{cases} |x| & \text{für } j = 1 \\ -|x| & \text{für } j = n \\ 0 & \text{sonst} \end{cases} \quad (1)$$

für alle Knoten $j \in V$ erfüllt ist. Dabei ist

$$|x| = \sum_{(1,k) \in A} x_{1k} = \sum_{(i,n) \in A} x_{in}$$

die Stärke des Flusses x . Ein Fluß heißt zulässig, falls mit einer für jeden Bogen $(i, j) \in A$ definierten Kapazität cap_{ij} gilt:

$$0 \leq x_{ij} \leq \text{cap}_{ij} . \quad (2)$$

Die lineare Restriktionen (1) und (2) beschreiben ein konvexes Polyeder X , das Flußpolyeder genannt wird.

Zusätzlich wird angenommen, daß der Transport einer Mengeneinheit entlang des Bogens $(i, j) \in A$ die Kosten c_{ij} verursacht. Für eine festgehaltene Flußstärke $x = \text{const.}$ (i. A. wird dabei vom Maximalfluß ausgegangen) wird ein Flußdurchlauf mit minimalen Kosten gesucht. Die Aufgabe

$$\text{MCF: } \min \{c^T x : x \in X, |x| = C_1 = \text{const.}\}$$

heißt kostenminimales Flußproblem, wobei $c = (c_{ij}) \in R^m$ der Kostenvektor ist. Falls nicht anders vereinbart, beziehen wir die Forderung $|x| = C_1$ in die Definition des Polyeders X mit ein.

MCF enthält als Spezialfälle eine Reihe von Standardproblemen des Operations Research: die Bestimmung kürzester Wege, das Transportproblem und das Maximalflußproblem. Zur Lösung von MCF stehen streng polynomiale Algorithmen bereit, d. h. ihre Laufzeit ist polynomial in n und m und hängt nicht von

$$\text{CAP} = \max \{ \text{cap}_{ij} : (i, j) \in A \} \text{ oder}$$

$$\text{COST} = \max \{ c_{ij} : (i, j) \in A \}$$

ab (vgl. dazu z. B. *Goldberg, Tarjan* (1987)).

Die formal allgemeineren Fälle mit nichttrivialen unteren Kapazitätsschranken und freien rechten Seiten können durch Einführung von Umkehrbögen bzw. von zusätzlichen Bögen auf den Fall des MCF zurückgeführt werden.

Zahlreiche praktische Anwendungen lassen sich nicht adäquat durch eine Zielfunktion beschreiben. Wir untersuchen im weiteren den Fall, daß auf X zwei Kostenfunktionen $c, d : X \rightarrow R_+$ erklärt sind. Eine Lösung $x^* \in X$ heißt (funktional) effizient oder vektoroptimal, falls kein $x \in X$ existiert, so daß mit $f(x) = (c^T x, d^T x)$ gilt:

$$f(x) \leq f(x^*) \text{ und } f(x) \neq f(x^*) .$$

Die Ermittlung der effizienten Minimallösungen bezeichnen wir mit \min^* . Dann untersuchen wir das folgende Problem:

$$\text{BMCF: } \min^* \{ f(x) : x \in X \} .$$

3. Komplexitätsergebnisse

Die Charakterisierung der dem Problem BMCF innewohnenden Komplexität gibt wichtige Aufschlüsse für den nachfolgenden Algorithmentwurf. Dabei ist es zunächst von Interesse, daß zur Lösung von MCF streng polynomiale Algorithmen bereitstehen. Ein zweites wichtiges Resultat betrifft den Spezialfall des bikriteriellen Wegeproblems. Folgendes Problem gehört in die Klasse der NP-vollständigen Aufgaben:

Ermittle zum Netzwerk $G = (V, A)$ mit Bewertungen $c, d : A \rightarrow R$ und vorgegebenen Konstanten $\alpha, \beta > 0$, ob es einen gerichteten Weg P von 1 nach n gibt, für den gilt

$$c(P) = \sum_{(i,j) \in P} c_{ij} \leq \alpha$$

$$d(P) = \sum_{(i,j) \in P} d_{ij} \leq \beta .$$

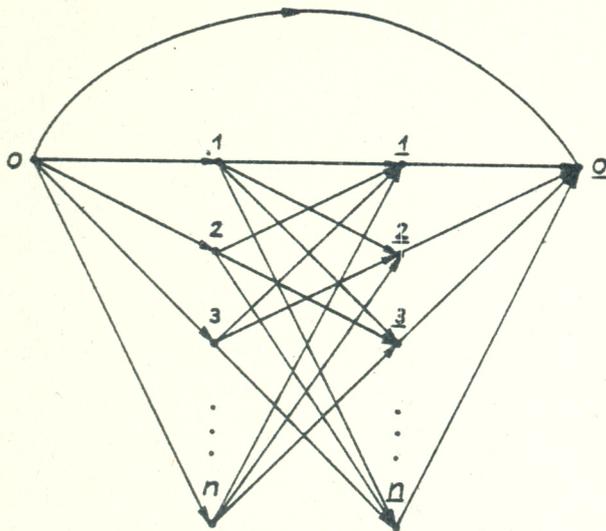


Bild 1

In Bezug auf BMCF betrachten wir zwei Fragestellungen:

- 1) Ermittle die Anzahl $\#_1$ der effizienten Basislösungen mit unterschiedlichen Bildpunkten.
- 2) Ermittle die Anzahl $\#_2$ der effizienten Basislösungen, die in unterschiedliche Eckpunkte des konvexen Polyeders aller Bildpunkte abgebildet werden.

Eine Antwort auf 1) wird durch Hansen (1980) gegeben, der vermittels des bikriteriellen Wegeproblems nachweist, daß $\#_1$ exponentiell von $n = |V|$ abhängen kann. Für das dabei verwendete pathologische Beispiel I gilt $\#_2(I) = 2^n$, i. A. gilt $\#_1(I) \geq \#_2(I)$ bezüglich beliebiger Instanzen I von BMCF. Vermittels des von Zadeh (1973) eingeführten Graphen $G_n = (V_n, A_n)$ von Bild 1 und spezieller Ansätze für die Kostenfunktionen konnte in Ruhe (1988b) nachgewiesen werden, daß auch $\#_2$ exponentiell von n abhängen kann, wenn gleich diese Schranke in praktischen Fällen kaum erreicht wird.

4. Algorithmen

4.1. Einführung

Zur Lösung multikriterieller Netzwerkflußprobleme lassen sich die Verfahren der parametrischen linearen Optimierung bzw. der Vektoroptimierung anwenden. Dabei sind jedoch eine Reihe von Vereinfachungen möglich, die sich aus der speziellen Struktur des Problems als Flußproblem in einem Netzwerk ergeben.

Für lineare Optimierungsaufgaben mit mehreren Zielfunktionen sind seit vielen Jahren Lösungskonzepte bekannt. Die einfachste Methode ist es, das lineare Optimierungsproblem für jede der K Zielfunktionen unabhängig voneinander zu lösen. Man erhält so eine optimale Entscheidungstabelle (Dück 1978), die wertvolle Erkenntnisse für die Entscheidungsfindung beinhaltet.

Eine andere Vorgehensweise besteht darin, eine gewichtete Summe der K linearen Zielfunktionen aufzustellen und danach die Optimierung mit eben dieser Summe als Ersatzzielfunktion vorzunehmen. Das Ergebnis erscheint in Abhängigkeit der Gewichte, deren praktische Festlegung bekanntlich recht problematisch ist. Wiederum anders ist die Methode, daß man $K-1$ Zielfunktionen als Nebenbedingung formuliert und die so umgewandelte Aufgabe als lineares Optimierungsproblem mit einer Zielfunktion behandelt. Es ist auch möglich, für die Zielfunktionen eine Prioritätsreihenfolge vorzugeben und danach Lösungen zu ermitteln.

In *Klingman* und *Mote* (1982) wird eine Netzwerkflußvariante zur multikriteriellen linearen Optimierungsaufgabe vorgestellt. Dabei wird im wesentlichen die der multikriteriellen Optimierungsaufgabe entsprechende lineare mehrparametrische Optimierungsaufgabe betrachtet und dann unter Ausnutzung der speziellen Struktur der Nebenbedingungen des Flußproblems der multikriterielle Simplexalgorithmus zur Bestimmung der effizienten Lösungen angewandt. Ferner wird eine Netzwerkflußvariante für die Lösung mit einer gewichteten Zielfunktion beschrieben und am Beispiel der Bestimmung des kürzesten Weges bei mehreren Zielfunktionen demonstriert. Dabei ist eine interaktive Lösungsprozedur entwickelt worden.

Parametrische und multikriterielle Probleme hängen eng zusammen: Multikriterielle Probleme lassen sich in parametrische überführen, wobei es zu jeder effizienten Lösung des multikriteriellen Problems einen Satz von Parametern gibt, so daß diese Lösung auch Optimallösung des so entsprechenden parametrischen Problems (mit spezieller Wahl der Parameter) ist.

Ferner haben multikriterielle Probleme wichtige Konvexitätseigenschaften, die algorithmisch genutzt werden können. So ist im Raum der Zielfunktionswerte eines multikriteriellen Problems die den effizienten Lösungen entsprechenden Kurve (Fläche) konvex.

4.2. Approximation konvexer Funktionen

Motiviert durch parametrische min-Kostenflußprobleme wurde von *Burkard*, *Hamacher* und *Rote* (1987) eine Theorie zur Approximation konvexer Funktionen entwickelt, die sowohl bei parametrischen als auch bei bikriteriellen Flußproblemen eine Anwendung finden kann. Bekannterweise ist die Optimalwertfunktion $z(t)$ eines parametrischen min-Kostenflußproblems eine konvexe Funktion im Parameterintervall $[0, T]$. Man kann nun $z(t)$ durch stückweise lineare, konvexe Funktionen $l(t)$ und $u(t)$ global approximieren mit

$$l(t) \leq z(t) \leq u(t) \text{ für alle } t \in [0, T].$$

Dabei soll einerseits die Berechnung der Funktionen $l(t)$ und $u(t)$ möglichst einfach sein, andererseits der Abstand der Funktionen bzgl. eines vorgegebenen Fehlermaßes möglichst klein werden.

Dazu wird $[0, T]$ unterteilt in

$$0 = t_0 < t_1 < t_2 < \dots < t_n = T$$

und $z(t)$ wird in $[t_{i-1}, t_i]$ ($i = 1, 2, \dots, n-1$) durch

$$u(t) = z(t_{i-1}) + [(z(t_i) - z(t_{i-1})) / (t_i - t_{i-1})] \cdot (t - t_{i-1})$$

ersetzt. Ferner wird $l(t)$ definiert durch

$$l(t) = \max \{ z(t_{i-1}) + z_{i-1}^+ \cdot (t - t_{i-1}), z(t_i) + z_i^- \cdot (t - t_i) \},$$

wobei z_{i-1}^+ die rechtsseitige Ableitung von z im Punkt t_{i-1} und z_i^- die linksseitige Ableitung von z im Punkte t_i ist. Dieses Vorgehen wird als Sandwich-Algorithmus bezeichnet.

Frei ist noch, wie die Unterteilung (t_i) des Intervalls gewählt wird. Wählt man als Unterteilung die Mittelpunkte jener Intervalle, in denen der Fehler noch größer als eine vorgegebene Schranke $\varepsilon > 0$ ist, so läßt sich folgendes zeigen:

Satz (Burkard, Hamacher, Rote (1987)):

Die Anzahl M der Funktionsauswertungen z , z^+ und z^- , die benötigt werden zur Berechnung einer oberen und unteren ε -Approximation $u(t)$ bzw. $l(t)$ von $z(t)$ ist beschränkt durch

$$M \leq \max \{ 2, \lfloor 3/2 \sqrt{T(z^-(T) - z^+(0))/2\varepsilon} \rfloor + 1 \}. \quad (3)$$

4.3. Ein exaktes Verfahren für BMCF

In Anlehnung an *Aneja, Nair* (1979) geben wir für das bikriterielle Problem kostenminimaler Flüsse ein Vorgehen an, bei dem Übergänge zwischen nicht notwendig benachbarten Eckpunkten des Bildbereiches erfolgen.

Sei Y_{eff} die Menge der effizienten Eckpunkte des Bildbereiches.

Satz:

Sei $y^r, y^s \in Y_{\text{eff}}$. Dann gilt mit

$$y^t = \arg \min \{ |y_2^s - y_2^r| \cdot y_1 + |y_1^s - y_1^r| \cdot y_2 : y = (y_1, y_2) \in Y \} \quad (4)$$

und $y^t \neq l \cdot y^r + (1-l) \cdot y^s$; $l \in [0,1]$, daß auch $y^t \in Y_{\text{eff}}$.

Auf obigen Satz basiert ein Verfahren zur Lösung von BMCF.

Zur Ermittlung der Startlösungen verwenden wir den

Satz:

Mit $x^1 \in \arg \text{lexmin} \{ (c^1x, c^2x) : x \in X \}$ und
 $x^2 \in \arg \text{lexmin} \{ (c^2x, c^1x) : x \in X \}$

gilt: $y^i = f(x^i) \in Y_{\text{eff}}$ für $i = 1, 2$.

Auf dieser Grundlage ist in *Ruhe* (1988a) ein Algorithmus angegeben, der in $O(|Y_{\text{eff}}| n^2 (m + n \log n) \log n)$ Operationen die Menge Y_{eff} bestimmt. Die Vorgehensweise ist auf andere stetige Probleme mit zwei Zielfunktionen übertragbar. Die in jeder Iteration zu lösende Teilaufgabe ist im Falle von BMCF vom Typ MCF und muß für andere bikriterielle Probleme durch das Lösen der entsprechenden Teilaufgabe ersetzt werden.

4.4. Approximationsverfahren für bikriterielle Flußprobleme

Für bikriterielle Flußprobleme kann ebenfalls die in Abschnitt 4.2. beschriebene Approximationsmethode angewandt werden. Dabei empfiehlt es sich jedoch, nicht den Fehler $\max |u(t) - l(t)|$ zu minimieren (der eine Zielfunktion bevorzugt), sondern den projektiven Abstand zwischen $u(t)$ und $l(t)$.

Es sei $L := \{(t, l(t)) : t \in [0, T]\}$ und $U := \{(t, u(t)) : t \in [0, T]\}$.

Dann ist der projektive Abstand Δ zwischen L und U festgelegt durch

$$\Delta = \max \{ \sup_{x \in L} \inf_{y \in U} \|y - x\|, \sup_{y \in U} \inf_{x \in L} \|y - x\| \} .$$

In *Ruhe* (1988a) werden Approximationen $l(t)$ und $u(t)$ mit $\Delta \leq \varepsilon$ ermittelt. Der Rechenaufwand zur Ermittlung dieser Approximationen beträgt

$$O((\Delta^0/\varepsilon) n^2 (m + n \log n) \log n) , \quad (5)$$

wobei Δ^0 den projektiven Abstand bezüglich der durch die Anstiege in den linken und rechten Randpunkten gegebenen ersten Approximation bedeutet.

In *Fruhwirt, Burkard, Rote* (1989) wird ebenfalls dieses Problem untersucht. Die Autoren diskutieren 3 Regeln zur Wahl des Zwischenpunktes eines Intervalls, in dem Δ größer als das vorgegebene ε ist, nämlich die

- Sehnenregel (siehe auch *Ruhe* (1988a))
- Winkelhalbierungsregel
- mittlere Steigungsregel.

Dabei konnte folgendes gezeigt werden

Satz (Fruhwirt, Burkard, Rote 1989):

Die Anzahl M der Funktionsauswertungen die nötig ist, um den projektiven Abstand Δ kleiner als ein vorgegebenes $\varepsilon > 0$ zu machen, beträgt bei der Winkelhalbierungsregel

$$M \leq \max \{2, \lfloor 3/2 \sqrt{T \alpha / 2\varepsilon} \rfloor + 1\} \quad (6)$$

bzw. bei der mittleren Steigungsregel

$$M \leq \max \{2, \lfloor 3/2 \sqrt{T \cdot D / 2\varepsilon} \rfloor + 1\} . \quad (7)$$

Dabei ist D die Differenz der Anstiege im linken und rechten Randpunkt des Intervalls $[0, T]$ sowie α der Winkel, der in der 1. Approximation von $l(t)$ auftritt. Für die Sehnenregel konnte Rote (1988) folgendes Ergebnis mit einer neuen Beweismethode zeigen:

Satz (Rote 1988):

Neben den Auswertungen in den Endpunkten des Intervalles $[0, T]$ benötigt die Sehnenregel maximal M_0 zusätzliche Auswertungen, wobei gilt

$$M_0 = \begin{cases} 0 & \text{für } A \alpha / \varepsilon^2 \leq 64 \\ \lfloor 4/\sqrt{3} (\sqrt[4]{A \alpha / \varepsilon^2} - 1) \rfloor & \text{für } A \alpha / \varepsilon^2 > 64. \end{cases} \quad (8)$$

Dabei ist A die Fläche des Dreieckes, das in der 1. Iteration von $l(t)$ und $u(t)$ gebildet wird.

4.5. Ein interaktives Vorgehen für mehr als zwei Kriterien.

Sowohl das in 4.3. beschriebene exakte als auch die in 4.4. dargestellten approximativen Verfahren sind zunächst nur für zwei Kriterien formuliert (eine Verallgemeinerung erscheint prinzipiell möglich). Um auch für mehr als zwei Kriterien eine praktikable Methode anzubieten, wurde ein interaktives Vorgehen auf der Grundlage der Parametervariation in der Zielfunktion realisiert. Bei dem vorliegenden Computerprogramm können bis zu 10 Kriterien Berücksichtigung finden.

5. Numerische Ergebnisse

5.1. Untersuchungen zur Anzahl der Knickstellen

Obwohl die worst-case-Untersuchungen gemäß Abschnitt 3. im Falle bikriterieller kostenminimaler Flüsse eine exponentiell von $n = |V|$ abhängende Anzahl von effizienten Eckpunkten des Bildbereiches ausweisen, zeigen die experimentellen Untersuchungen für den ‚Normalfall‘ ein weitaus günstigeres Verhalten.

Zu dem in Abschnitt 4.3. beschriebenen Vorgehen zur Lösung von BMCF wurde ein Computer-Programm geschrieben (Turbo-Pascal) und für zwei Klassen von Graphen am Rechner PC 1715 getestet.

- Die in Abschnitt 3. dargestellten pathologischen Graphen mit zufällig erzeugten Kosten
- Grid-Graphen für $n = 4$ und 5 .

Die beiden ganzzahligen Kostenvektoren wurden als Realisierungen von gleichverteilten Zufallsgrößen bestimmt. Als Parameterintervall für die Zufallsgrößen wurden

- a) [0,5], [0,10] und [0,50] bzw.
 b) [0,5], [0,10] und [0,20]

verwendet. Die in den Tabellen 1 und 2 angegebenen Werte sind Durchschnittsgrößen von jeweils 5 gerechneten Beispielen.

Tabelle 1
Kardinalität von Y_{eff}

n	[0,5]	[0,10]	[0,50]
3	1,6	2,0	1,8
4	2,2	2,4	1,8
5	2,2	2,6	1,4

Tabelle 2
Kardinalität von Y_{eff}

n	[0,5]	[0,10]	[0,20]
4	5,6	4,0	7,0
5	8,4	10,8	9,6

Darüber hinaus wurden für eine Reihe von Standardtestbeispielen großer Dimension, die durch das dafür bereitgestellte Programmsystem NETGEN (siehe *Klingman, Napier & Stutz 1974*) erzeugt wurden, die Anzahl $\#_2 = |Y_{\text{eff}}|$ der effizienten Eckpunkte des Bildbereiches bestimmt.

Tabelle 3
Parameter der NETGEN-Testbeispiele und Kardinalität von $|Y_{\text{eff}}|$

Problem	$ V $	$ A $	$ x $	$ Y_{\text{eff}} $
1	100	900	600	143
2	100	1000	900	189
3	100	1000	4000	280
4	400	4000	3600	853
5	400	4000	3600	863
6	400	4000	15000	1286
7	800	8000	6000	1539
8	800	8000	12500	2179
9	800	8000	30000	2872

Die Berechnungen zeigen, daß die Anzahl der effizienten Eckpunkte des Bildbereiches weit davon entfernt ist, exponentiell mit der Knotenanzahl der Netzwerke zu wachsen. In keinem Beispiel übersteigt die Anzahl der Eckpunkte die Bogenanzahl des Netzwerkes. Für praktische Probleme dürfte im allgemeinen das in 4.5. geschilderte interaktive Verfahren ausreichen.

5.2. Vergleich von approximativen Verfahren

Mit den in Abschnitt 4.4. vorgestellten Regeln zur Wahl des Zwischenpunktes und der noch darzustellenden LTR-Regel wurden Vergleichsrechnungen durchgeführt. In Abhängigkeit von der vorgegebenen Genauigkeit $\varepsilon > 0$ wurden CPU-Zeit und maximale Höhe (*sth*) des zur Speicherung der Zwischenlösungen notwendigen Stacks für die in Tabelle 3 beschriebenen Testbeispiele gegenübergestellt. Außerdem enthält Tabelle 4 Aussagen zur Anzahl $|Z_{\text{eff}}|$ der jeweils ermittelten effizienten Lösungen. Die LTR-Regel benutzt eine obere Schranke M für die Anzahl der Iterationen der Winkelhalbierungsregel und führt Rotationen des Bildbereiches Y um $F * \alpha / M$ durch. Dabei ist α der Winkel, der durch das aktuelle Segment festgelegt ist, und $F > 1$ eine empirisch zu bestimmende Konstante (bzgl.

weiterer Details siehe *Fruhwirt, Burkard, Rote* 1989). Als Grundroutine zur Lösung von MCF wurde der primale Simplexalgorithmus nach *Ahrens & Finke* (1980) benutzt. Bei geringfügigen Vorteilen in der Rechenzeit ist die LTR-Regel vor allem hinsichtlich des Parameters *sth* (Anzahl der zu speichernden Zwischenlösungen) am günstigsten.

Tabelle 4
Vergleich der a) Winkelhalbierungsregel
b) Sehnenregel
c) LTR-Regel
für die in Tabelle 2 charakterisierten Testbeispiele

Pro- blem	ε/Δ^0	a)			b)			c)		
		$ Z_{\text{eff}} $	CPU (sek)	sth	$ Z_{\text{eff}} $	CPU (sek)	sth	$ Z_{\text{eff}} $	CPU (sek)	sth
1	0.01	13	3.8	3	15	4.0	3	13	3.2	2
	0.001	37	6.9	4	33	6.8	5	40	6.0	2
	0.0001	85	12.8	5	87	13.3	6	88	11.3	2
2	0.01	14	5.0	3	16	5.0	3	13	3.6	2
	0.001	38	8.1	5	35	7.6	5	37	6.2	2
	0.0001	96	15.2	5	98	15.7	6	98	12.7	2
3	0.01	15	5.2	3	16	5.2	3	14	4.2	2
	0.001	39	8.6	5	38	8.3	5	40	6.9	2
	0.0001	101	16.2	6	100	15.7	6	105	13.1	2
4	0.01	13	31.7	3	16	36.3	3	14	26.4	2
	0.001	43	57.6	5	39	55.3	5	48	44.0	2
	0.0001	124	105.9	6	121	103.6	6	131	79.4	2
5	0.01	14	34.4	3	16	36.2	3	13	25.7	2
	0.001	42	58.8	5	39	54.5	5	46	43.5	2
	0.0001	123	108.3	6	123	107.9	6	130	83.5	2
6	0.01	16	41.3	3	17	4.7	4	13	30.7	2
	0.001	43	65.8	5	35	61.4	5	42	45.7	2
	0.0001	129	118.1	4	127	120.8	6	126	84.3	2
7	0.01	13	91.9	3	16	102.7	3	14	74.6	2
	0.001	43	156.4	5	38	146.9	5	47	112.4	2
	0.0001	123	262.6	6	125	277.3	7	139	202.2	2
8	0.01	13	106.0	3	17	123.8	4	15	93.0	2
	0.001	43	181.8	5	36	166.7	5	44	127.1	2
	0.0001	130	310.1	7	129	316.7	7	146	224.5	2
9	0.01	14	136.0	3	16	146.2	3	13	104.5	2
	0.001	43	211.1	5	36	203.0	5	44	149.0	2
	0.0001	130	346.7	6	127	349.7	7	152	249.5	2

Neben der rechentechnischen Dominanz der LTR-Regel belegen die numerischen Daten, daß die Zahl der effizienten Lösungen bis zu einer relativen Genauigkeit von $\varepsilon/\Delta^0 = 0.001$ in überschaubaren Größenordnungen bleibt und das approximative Vorgehen somit eine praktikable Methode darstellt.

LITERATUR

- Ahrens, J. H.; Finke, G.* (1980): Primal transportation and transshipment algorithms. *Zeitschrift für Operations Research* 24, 1–42.
- Aneja, Y. P.; Nair, K. P.* (1979): Bicriteria transportation problem, *Management Science* 25, 73–78.
- Burkard, R. E.; Hamacher, H. W.; Rote, G.* (1987): Approximation of convex functions and applications in mathematical programming. Technische Universität Graz, Inst. für Mathematik, Report 89–1987.
- Carstensen, P. J.* (1983): Complexity of some parametric integer and network programming problems. *Mathematical Programming* 26, 64–75.

- Dück, W.* (1978): Optimierung unter mehreren Zielen. Berlin: Akademie-Verlag.
- Fruhwirt, B.; Burkard, R. E.; Rote, G.* (1989): Approximation of convex curves with application to the bicriterial minimum cost flow problem. *European Journal of Operational Research* 42, 326–338.
- Goldberg, A. V.; Tarjan, R. E.* (1987): Solving minimum-cost flow problems by successive approximation. *Proceedings of the 19th ACM STOC*, 7–18.
- Hansen, P.* (1980): Bicriterion path problems. In: *Multiple criteria decision making-Theory and applications. Lecture Notes in Economics*, 177, 109–127.
- Klingman, D.; Napier, A.; Stutz, J.* (1974): A program for generating large scale capacitated assignment, transportation, and minimum cost flow problems. *Management Science* 20, 814–821.
- Klingman, D.; Mote, J.* (1982): Solutions approaches for network flow problems with multiple criteria. *Advances in Management Studies* 1, 1–30.
- Rote, G.* (1988): Quadratic convergence of the sandwich algorithm for approximating convex functions. Technische Universität Graz, Institut für Mathematik, Report 118–1988.
- Ruhe, G.* (1988a): Flüsse in Netzwerken – Komplexität und Algorithmen, Dissertation B. Technische Hochschule Leipzig, Sektion Mathematik und Informatik.
- Ruhe, G.* (1988b): Complexity results for multicriterial and parametric network flows using a pathological graph of Zadeh. *Zeitschrift für Operations Research* 32, 9–27.
- Ruhe, G.; Sieber, N.* (1986): Zur Komplexität zweier vektorieller Netzwerkflußprobleme. *Proceedings 18. Jahrestagung „Mathematische Optimierung“*, Seminarberichte HUB, Sektion Mathematik, Nr. 85.
- Zadeh, N.* (1973): A bad network problem for the simplex method and other minimum cost flow problems. *Mathematical Programming* 5, 255–266.

Verfasser :

o. Prof., Dr. phil. *R. E. Burkard*, Dr.-Ing. *G. Rote*,
Institut für Mathematik,
Technische Universität Graz,

Dr. sc. nat. *G. Ruhe*, o. Prof. Dr. rer. nat. *N. Sieber*,
Sektion Mathematik und Informatik,
Technische Hochschule Leipzig.