予算制約をもつネットワークデザイン問題の近似解法

片　山　直　登

1　はじめに

ネットワークデザイン問題は，様々な条件の下で，ネットワークの形状と多品種のものの流れを求める問題の総称である。この問題には様々な応用分野が存在しており，主な対象として，道路ネットワーク設計，通信ネットワーク設計，輸送路線計画，航空ネットワーク計画や，上下水道・ガス・電力ネットワーク設計などを挙げることができる。

予算制約をもつネットワークデザイン問題（budget network design problem, *BND*）は，与えられたデザイン費用の予算の範囲内で構成できる適切なネットワークを求める基本的な問題である。目的関数は線形のルーティング費用であり，線形のデザイン費用を予算制約にもつモデルとなる。

一方，線形のルーティング費用と線形のデザイン費用の和を最小にするネットワークを求める問題は固定費用をもつネットワークデザイン問題（fixed charge network design problem, *FND*）とよばれ，この問題に対しては高速で誤差が小さい貪欲解法である Minoux 法が開発されている。

本研究では，*FND* に対する Minoux 法とその改良法を *BND* に適用した解法を提案する。また，提案した解法を用いた数値実験を行い，解法の特性を検討する。

2　従来の研究

BND に対して，多くの厳密解法と近似解法が提案されている。厳密解法では下界平面法とよばれる緩和問題を用いた方法，Lagrange 緩和を用いた方法や分枝限定法などがある。一方，近似解法では，貪欲解法であるフォワード法やバックワード法などがある。

Boyce-Farhi-Weischedel[3] は，ルーティング費用の総和を最小にするネットワークが

すべてのアークを含むネットワークであり，そのルーティング費用は*BND*の下界値となる性質を用いた下界平面を示している。Hoang[6]は，あるアークを取り除いたときに，この両端点を始点と終点とする品種のフローは迂回したパスを通るため，ルーティング費用が増加するという考え方を用いた下界平面を示している。Gallo[5]は，アークの両端を始点・終点としない品種のフローに対するルーティング費用の増加量も考慮し，この増加量を下界平面の係数値に加算していく方法を示している。Ahuja-Murty[1]は，2本目のアークを取除いたときの最小ルーティング費用の増加量が1本目のアークを取除いたときの増加量より大きな場合には，この差分を下界平面の2本目のアークの係数に加算できること考慮した下界平面を示している。片山直登-春日井[16]は，フロー保存式をLagrange緩和した問題が強い下界平面となることを用いた下界値を示している。

Scott[10]は，下界値による限定操作を行なわないバックトラック法を提案している。分枝限定法では，Boyce-Farhi-Weischedel[3]がBoyce-Farhi-Weischedeの下界平面を用いた分枝限定法を示しており，Hoang[6]はHoangの下界平面を利用した分枝限定法を示している。また，Dionne-Florian[4]はHoangの下界平面を利用し，Boffey-Hinxman[2]は分枝変数の選択に工夫をした方法を示している。

*BND*では，目的関数であるルーティング費用を算出するために全ノード間の最小費用路（最短路）を求める必要があり，繰り返し目的関数値を評価するためには多くの計算が必要となる。このような理由から，近似解法としては主に貪欲的な解法が用いられている。Scott[10]は，最小木などの初期ネットワークからアークを順に加えていくフォワード法と，全アークを含む初期ネットワークからアークを順に取り除くバックワード法を示している。西村昂-日野泰雄[13]は，近傍・入れ換えグラフと組合せ目的関数を用いて，フォワード法とバックワード法を改良している。森津秀夫[14]は，実行可能化・連結網化を用いた分枝限定法や簡易フォワード・バックワード法などを提案している。

Dionne-Florian[4]は，疑似緩和問題を用いた分枝限定法を示している。また，Wong[12]は，閾値を用いた近似解法を示している。ノード数が十分に多いとき，この閾値ヒューリスティック法で得られる解では，その目的関数値と下界値（最適値）の比が1に収束することが示されている。

一方，*FND*に対しては，効率的な貪欲解法が提案されている。Minoux[7,8,9]は高速化したバックワード法を示し，片山[15]はMinoux法の改良を行なっている。

3　問題の定式化

需要の発生地点や集中地点，交差点，コンピュータ，集線装置，施設などのように点で表すことのできるものをノードとよび，道路，路線，通信回線などのように点と点を結ぶ線をアークとよぶ。車両，貨物，通信などのように，ネットワーク上には決められ

たノード間を移動するものが存在する。単位時間当たりの移動量を需要量とよび，アーク上のこの流れをフロー，需要の出発ノードを始点，到着ノードを終点とよぶ。一般的には品種はものの種類を表すが，本研究では同一の始点・終点をもつものを同一の品種とし，始点または終点が異なるものは異なる品種として区別する。

アークには，デザイン費用とルーティング費用が与えられている。デザイン費用は，道路建設費用，回線設置・レンタル費用などのようにアークを設置・使用するときに発生する固定費用を表す。ルーティング費用は，ものがアーク上を移動するときに発生する費用で，走行費用（時間），輸送費用（時間），通信費用（時間）などのようなフロー量に依存する変動費用（時間）を表す。また，使用できるデザイン費用に上限が与えられており，これを予算とよぶ。

ノード集合をN，アーク集合をA，品種の集合をKとする。アーク(i,j)を設置するか否かを表すデザイン変数をy_{ij}とし，アーク(i,j)上の品種kのフロー量を表すアークフロー変数をx_{ij}^kとする。アーク(i,j)の設置によって発生するデザイン費用をf_{ij}，品種kの単位当たりのルーティング費用をc_{ij}^k，デザイン費用のための予算をBとする。品種kの需要量をd^kとし，ノードnが品種kの始点であれば$-d^k$，終点であればd^k，それ以外では0である定数をd_n^kと表す。

アークフロー変数を用いたBNDの定式化を示す。

$$(BND)\quad \text{最小化}\quad \sum_{(i,j)\in A}\sum_{k\in K}(c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) \tag{1}$$

$$\text{条件}\quad \sum_{i\in N}x_{in}^k - \sum_{j\in N}x_{nj}^k = d_n^k \quad for\ all\ n\in N,\ k\in K \tag{2}$$

$$\sum_{(i,j)\in A}f_{ij}y_{ij}\leq B \tag{3}$$

$$x_{ij}^k\leq d^k y_{ij},\ x_{ji}^k\leq d^k y_{ij} \quad for\ all\ k\in K,\ (i,j)\in A \tag{4}$$

$$y_{ij}\in\{0,1\} \quad for\ all\ (i,j)\in A \tag{5}$$

$$x_{ij}^k\geq 0,\ x_{ji}^k\geq 0 \quad for\ all\ k\in K,\ (i,j)\in A \tag{6}$$

(1)式は，ルーティング費用の総和の最小化を表す。(2)式はフロー保存式とよばれる式であり，ノードnにおける流入量と流出量の差が，ノードnが品種kの始点であれば$-d^k$，終点であればd^k，その他の中継ノードであれば0となることを表し，始点から出たフローが必ず終点に到着することを表す。(3)式は，選択したアークのデザイン費用の合計が予算以下であることを表す予算制約である。(4)式は，アーク(i,j)が存在するときにのみフローの存在を許す強制制約式であり，アーク(i,j)上のフロー量とデザイン変数の関係を表す。(5)式はデザイン変数の0-1条件，(6)式はアークフロー変数の非負条件である。

4　従来の近似解法

BND は，多品種最小費用フロー（最短路）問題とナップサック問題を含む混合整数計画問題である。このため，ノード数が10～15程度の小規模な問題であれば，一般の汎用パッケージで最適解が得られる可能性がある。しかし，ネットワークが完全グラフの場合で，先に示した定式化を用いた場合には，ノード数が30程度であっても，0－1変数であるアーク変数が400，フロー変数は30万を超え，一般的な汎用パッケージを用いてそのまま解くことは困難である。一方，Lagrange 緩和法では強い下界値を求めることができるが，ノード数が100，アーク数が5000程度を超えると膨大な計算時間が必要となる。したがって，規模の大きな問題では，最適解を求める代わりに十分良い実行可能解を短時間で求める近似解法が必要となる。

最も基本的な近似解法として，バックワード法が知られている。ここで，バックワード法のアルゴリズムを示しておく。

バックワード法

ステップ1　すべてのアークを含むネットワークを初期解とし，ルーティング費用をアークの長さとした各品種の始点と終点間の最短路を求め，この経路上に各需要量を流したものをフロー解とする。

ステップ2　現在のネットワークに含まれているすべてのアークに対して，次の計算を行なう。

このアークをネットワークから削除したときの総ルーティング費用の増加量を計算する。もし，ネットワークが非連結になれば，増加費用を∞とする。

ステップ3　総ルーティング費用の増加量が最小であるアークを求め，このアークをネットワークから削除する。現在のネットワークのデザイン費用の合計が予算以下になればステップ4へ，そうでなければステップ2へ戻る。

ステップ4　現在のネットワークに含まれていないアークで，このアークをネットワークに付加したときにデザイン費用の合計が予算を超えないすべてのアークに対して，次の計算を行う。そのようなアークが存在しなければ，終了する。

このアークをネットワークに付加したときの総ルーティング費用の減少量を計算する。

ステップ5　総ルーティング費用の減少量が最大となるアークを求め，このアークをネットワークに付加し，ステップ4へ戻る。

ステップ4と5では，予算の残余分に対してフォワード法を行なっている。バックワード法は，貪欲解法の中で最も良い解法の一つである。しかし，バックワード法では，

1本のアークを付加や削除したネットワーク上における最短路問題を繰り返し解くことが必要となる。このように構造が少し変化したネットワークに対しては，Murchland法(Murchland's algorithm)[11]を用いることによって，すべてのノード間の最短路問題を効率的に求めることができる。Murchland法を用いると，アークを付加したネットワークでは $O(|N|^2)$，アークを削除したネットワークでは $O(|N|^3)$(経験的には $O(|N|^2)$ で，全ノード間の最短路を求めることができる。このとき，バックワード法の計算量は $O(|A|^2|N|^3)$ であり，密で規模の大きな問題に適用することは困難が伴う。

一方，*FND* に対して，Minoux[7,8,9]は高速かつ誤差が小さい近似解法を示している。*FND* と *BND* の相違点は，デザイン費用を目的関数に含むか，予算として制約条件にもつかである。したがって，*FND* に有効である Minoux 法を *BND* に適用すれば，効果的な解法となることが期待できる。

Minoux 法では，はじめに各アーク (i, j) に対して，このアークを削除したときのノード i，j 間の最小ルーティング費用パスに，現在アーク (i, j) 上を流れているフローを流し代えたときの目的関数の減少量を求め，この値をアークを削除したときの目的関数値を減少量の評価値としている。しかし，アークを削除したときに，必ずしも i，j 上のフローはノード i，j 間の最小ルーティング費用パス上を流れるとは限らないため，この評価値は近似値となる。この評価値の大きい順にアークの削除リストを作成し，実際に削除の順になるときに目的関数値の減少量の近似値を再評価する。これがリスト中で最大であるときに限りアークを削除し，そうでなければ評価値にしたがって，このアークを削除リストの適切な位置に格納する。

FND に対する Minoux 法を示しておく。

FND に対する Minoux 法

ステップ1　すべてのアークを含むネットワークを初期解とし，ルーティング費用をアークの長さとした各品種の始点と終点間の最短路を求め，この経路上に各需要量を流したものをフロー解とする。

ステップ2　ネットワークに含まれているすべてのアークについて，減少量の近似的な評価値を計算する。

ステップ3　減少量の評価値の正の大きい順に，該当するアークと減少量の評価値をリストに格納する。

ステップ4　リスト内の減少量の評価値が正で最大のアークを選ぶ。減少量の評価値が正であるアークがなければ，終了する。

ステップ5　選択したアークに対して，現在のネットワーク上における減少量の評価値を再計算する。この値がリスト内で，正で最大であればステップ6へ。そうでなければ，減少量の評価値にしたがって，このアークをリスト内の適切な位置に移動し，

ステップ4へ戻る。

ステップ6　選択したアークをネットワークとリストから削除し，ステップ4へ戻る。

5　*BND*の近似解法

*FND*と*BND*の相違点は，デザイン費用を目的関数に含むか，予算として制約条件にもつかである。また，*BND*の実行可能性は，ネットワークが連結し，かつデザイン費用が予算以下であることで判定できる。したがって，Minoux法を*BND*に適用することは比較的容易である。そこで，最大ネットワークから順番にアークを取り除き，現在のデザイン費用の合計が予算以下になれば計算を終了するという操作を行なえば良いことになる。

一方，*BND*ではアークを取り除いたときに目的関数である総費用の減少量を評価値として用いている。しかし，*FND*では，アークを取り除いたときに総ルーティング費用は増加（非減少）するため，総ルーティング費用の増加量を近似的に評価して，この値の小さい順の削除リストを作成し，この順にアークの削除を検討すれば良いことになる。そこで，*FND*に対するMinoux法と同様に，アーク（i, j）を削除したときのノードi, j間の最小ルーティング費用パスに，現在アーク（i, j）上を流れているフローを流し代えたときの目的関数であるの総ルーティング費用の増加量を求め，この値をアークを削除したときの総ルーティング費用の増加量の近似的な評価値とする。

*BND*に対するMinoux法のアルゴリズムを示す。

Minoux法

ステップ1　すべてのアークを含むネットワークを初期解とし，ルーティング費用をアークの長さとした各品種の始点と終点間の最短路を求め，この経路上に各需要量を流したものをフロー解とする。

ステップ2　ネットワークに含まれているすべてのアークについて，このアークを取り除いたときの総ルーティング費用の増加量の近似的な評価値を計算する。

ステップ3　増加量の評価値の小さい順に，該当するアークと増加量の評価値をリストに格納する。

ステップ4　リスト内の増加量の評価値が最小のアークを選ぶ。

ステップ5　選択したアークに対して，現在のネットワーク上における増加量の評価値を再計算する。この値がリスト内で最小であればステップ6へ行く。そうでなければ，増加量の評価値にしたがって，このアークをリスト内の適切な位置に移動し，ステップ4へ戻る。

ステップ6　選択したアークをネットワークとリストから削除する。現在，ネットワー

クに含まれているアークのデザイン費用の合計が予算以下になれば終了し，そうでなければステップ 4 へ戻る。

バックワード法とは異なり，この Minoux 法には予算以下となった場合のフォワード法を加えていない。しかし，バックワード法と同様に予算の残りに対してフォワード法を行なうことも容易である。しかし，ここでは計算量の関係から，あえてフォワード法を加えていない。

Minoux 法は，初期計算を除けば，アーク (i, j) が実際に削除の対象となる場合に限り増加量の近似的な評価値を計算するだけであるので，非常に高速な解法となる。計算量は $O(|A|^2|N|^2)$ であるが，平均的には $O(|A||N|^2)$ である。

Minoux 法では，目的関数である総ルーティング費用の増加量を近似値として求めている。これは，その時点における厳密な目的関数の増加量を求めるためには全対全の最短路問題を厳密に解く必要があるため，計算量の増加を抑える必要があるためである。しかし，1 本のアークが削除されたネットワーク上の最短路問題は，Murchland 法[11] を用いると $O(|N|^2)$ で求めることができる。したがって，Murchland 法を用いてアークを削除したときの目的関数の変化量を厳密に評価しても，計算量は同じになる。この解法の計算量も $O(|A|^2|N|^2)$ であるが，平均的には $O(|A||N|^2)$ である。

Minoux 法における目的関数の増加量を厳密に評価するように改良した方法のアルゴリズムを示す。

Minoux 法の改良法

ステップ 1　すべてのアークを含むネットワークを初期解とし，ルーティング費用をアークの長さとした各品種の始点と終点間の最短路を求め，この経路上に各需要量を流したものをフロー解とする。

ステップ 2　ネットワークに含まれているすべてのアークについて，このアークを取り除いたときの総ルーティング費用の増加量を厳密に計算する。

ステップ 3　増加量の小さい順に，該当するアークと増加量をリストに格納する。

ステップ 4　リスト内の増加量が最小のアークを選ぶ。

ステップ 5　選択したアークに対して，現在のネットワーク上における増加量を再計算する。この値がリスト内で最小であればステップ 6 へ行く。そうでなければ，増加量にしたがって，このアークをリスト内の適切な位置に移動し，ステップ 4 へ戻る。

ステップ 6　選択したアークをネットワークとリストから削除する。現在，ネットワークに含まれているアークのデザイン費用の合計が予算以下になれば終了し，そうでなければステップ 4 へ戻る。

表 1　問題の規模

ノード数	アーク変数	品種数	フロー変数
10	45	45	4050
20	190	190	72200
30	435	435	378450
40	780	780	1216800
50	1225	1225	3001250
60	1770	1770	6265800
70	2415	2415	11664450
80	3160	3160	19971200
90	4005	4005	32080050
100	4950	4950	49005000

表 2　近似解の誤差(%)と計算時間の比較：予算＝最小木の 2 倍

ノード数	Scott 法		Minoux 法		改良法	
	誤差(%)	計算時間(s)	誤差(%)	計算時間(s)	誤差(%)	計算時間(s)
10	0.22	0.1	0.30	0.0	0.30	0.0
15	0.61	0.1	0.60	0.1	0.60	0.0
20	0.49	0.3	0.53	0.1	0.53	0.1
25	0.81	1.2	0.88	0.0	0.88	0.0
30	0.69	3.2	0.64	0.1	0.64	0.0
35	0.87	7.8	0.87	0.2	0.88	0.1
40	0.98	16.7	1.10	0.3	1.09	0.2
45	1.14	36.3	1.09	0.4	1.11	0.3
50	1.48	74.6	1.53	0.5	1.53	0.3
55	1.43	136.6	1.69	0.7	1.68	0.5
60	1.34	232.6	1.66	1.0	1.62	0.7
65	1.54	383.7	1.66	1.3	1.63	1.0
70	1.75	598.1	1.80	1.7	1.83	1.3
75	1.85	893.1	2.02	2.5	2.05	2.1
80	1.84	1305.2	1.87	3.1	1.92	2.1
85	2.08	1914.1	2.18	4.0	2.17	2.7
90	2.06	2692.9	2.20	5.4	2.18	3.2
95	2.55	3716.9	2.54	6.7	2.57	4.1
100	2.47	5016.6	2.48	8.8	2.52	4.9

6　数値実験

Scott のバックワード法 (Scott 法)，Minoux 法 (Minoux 法) と Minoux 法の改良法

表3　近似解の誤差(%)と計算時間の比較：予算＝最小木の4倍

ノード数	Scott法		Minoux法		改良法	
	誤差(%)	計算時間(s)	誤差(%)	計算時間(s)	誤差(%)	計算時間(s)
10	0.01	0.0	0.01	0.0	0.01	0.0
15	0.08	0.1	0.08	0.0	0.08	0.0
20	0.09	0.3	0.10	0.0	0.10	0.0
25	0.12	0.8	0.10	0.1	0.10	0.0
30	0.11	2.4	0.11	0.1	0.11	0.0
35	0.14	6.0	0.14	0.2	0.14	0.1
40	0.15	13.0	0.18	0.3	0.17	0.2
45	0.19	26.2	0.24	0.4	0.25	0.2
50	0.26	50.2	0.28	0.5	0.28	0.3
55	0.25	89.6	0.29	0.7	0.29	0.6
60	0.27	151.1	0.29	0.9	0.30	0.7
65	0.28	245.0	0.32	1.2	0.33	1.1
70	0.31	381.5	0.34	1.6	0.33	1.2
75	0.33	574.3	0.38	2.1	0.37	2.0
80	0.31	840.0	0.36	2.7	0.37	2.4
85	0.41	1221.6	0.43	3.5	0.43	2.7
90	0.35	1732.5	0.40	4.9	0.38	4.0
95	0.44	3626.4	0.49	6.5	0.49	6.5
100	0.48	3262.9	0.53	8.7	0.53	7.2

(改良法) を数値実験によって比較する。100×100の平面上にランダムにノードを発生し，ノード間のユークリッド距離を整数化した値をデザイン費用する。ルーティング費用はすべての品種で同一とし，デザイン費用に比例するものとする。予算制約は，デザイン費用をアークの重みとした最小木のデザイン費用の2～8倍に設定する。すべての2ノード対間にアークが使用可能とし，すべての2ノード対間に品種が流れるものとし，すべての需要量を1とする。ノード数は10から100までとし，同一ノード数では10組のデータを使用する。使用計算機はIBM PC互換機，使用言語はMicrosoft Fortan Power Station Ver. 4である。また，複数の計算機を使用しているため，CPU Celeron 450 Mz, メモリ64 Mb, OS Windows 2000機の計算時間に換算している。なお，近似解の誤差を定めるために，Lagrange緩和法[16]を用いて下界値を算出する。

表1にノード10から100までの問題の変数規模を示す。ノード数自体は少数でも，決定すべき変数は大規模であることがわかる。予算＝最小木の2倍の場合における，各解法によって得られた近似解の平均目的関数値の誤差と1問当りの平均計算時間を表2に示す。誤差は，“100×(近似解の平均目的関数値－*Lagrange*緩和法による下界値)/*Lagrange*緩和法による下界値”である。同様に，予算＝最小木の4倍の結果を表3に，予算＝

表 4　近似解の誤差(%)と計算時間の比較：予算＝最小木の 6 倍

ノード数	Scott 法		Minoux 法		改良法	
	誤差(%)	計算時間(s)	誤差(%)	計算時間(s)	誤差(%)	計算時間(s)
10	0.01	0.1	0.01	0.1	0.01	0.0
15	0.02	0.0	0.02	0.1	0.02	0.0
20	0.03	0.2	0.03	0.1	0.03	0.1
25	0.03	0.8	0.04	0.0	0.04	0.0
30	0.04	2.3	0.06	0.1	0.06	0.0
35	0.06	5.9	0.07	0.1	0.07	0.1
40	0.07	12.7	0.07	0.2	0.07	0.1
45	0.07	25.9	0.09	0.3	0.08	0.3
50	0.09	49.7	0.10	0.5	0.10	0.4
55	0.09	88.8	0.11	0.7	0.11	0.5
60	0.08	150.3	0.10	0.9	0.10	0.7
65	0.11	245.7	0.12	1.3	0.12	0.9
70	0.11	379.4	0.11	1.7	0.11	1.2
75	0.10	572.8	0.13	2.0	0.13	1.6
80	0.10	842.1	0.12	2.9	0.12	2.0
85	0.14	1402.5	0.16	4.0	0.16	2.5
90	0.12	2531.7	0.14	4.8	0.14	4.1
95	0.17	2795.9	0.19	7.3	0.19	5.6
100	0.18	3252.1	0.21	8.4	0.21	4.8

最小木の 6 倍の結果を表 4 に，予算＝最小木の 8 倍の結果を表 5 に示す。

Scott 法で求められた近似解は優れており，予算が最小木の 2 倍では最適解（下界値）との差である誤差が最大でも2.5%であり，予算が最小木の 4 倍を超えると誤差が0.5%以下となる。一方，Minoux 法や改良法は，Scott 法と比べて最大でも0.3%程度悪い解となっている。また，Minoux 法と改良法の間では，ほとんど誤差に差が見られない。

計算時間においては，Scott 法はノード80を超えると膨大な時間を必要とするため，大規模な問題に対してそのまま適用することは困難である。一方，Minoux 法と改良法では，ノード100においても10秒以内の短時間で近似解を算出しており，大規模な問題に対しても有効な解法であることが分かる。Minoux 法に比べて，改良法は80%から半分程度の計算時間となっている。これは，近似値に比べて厳密な目的関数値の変化量を求めることによって，アルゴリズムのステップ 5 においてリスト中でその値が最小となることが多くなり，再計算の回数が減少するためと考えられる。*FND* における Minoux 法[16]では，改良法の方が誤差が大きいが計算時間が10%程度増加することと比べると，*BND* における Minoux 法ではその振る舞いが異なっている。

表5　近似解の誤差(%)と計算時間の比較：予算＝最小木の8倍

ノード数	Scott法		Minoux法		改良法	
	誤差(%)	計算時間(s)	誤差(%)	計算時間(s)	誤差(%)	計算時間(s)
10	0.00	0.1	0.00	0.0	0.00	0.0
15	0.01	0.0	0.01	0.1	0.01	0.0
20	0.01	0.2	0.01	0.1	0.01	0.1
25	0.01	0.8	0.01	0.1	0.01	0.0
30	0.02	2.3	0.02	0.1	0.02	0.1
35	0.02	5.7	0.03	0.2	0.03	0.0
40	0.04	12.5	0.04	0.2	0.04	0.1
45	0.03	25.3	0.04	0.3	0.04	0.1
50	0.04	48.9	0.05	0.5	0.05	0.3
55	0.04	87.6	0.04	0.6	0.04	0.4
60	0.04	148.2	0.05	0.9	0.05	0.6
65	0.05	240.9	0.06	1.2	0.06	0.9
70	0.05	375.8	0.06	1.7	0.06	1.2
75	0.05	566.7	0.07	2.0	0.07	1.5
80	0.04	831.2	0.06	2.7	0.06	1.9
85	0.06	1202.0	0.08	3.7	0.08	2.4
90	0.06	1695.6	0.08	5.3	0.08	3.0
95	0.07	2369.2	0.08	6.3	0.08	3.9
100	0.09	3233.7	0.10	8.5	0.10	4.8

7　おわりに

本研究では，予算制約をもつネットワークデザイン問題に対する2種類の近似解法を提案した。提案したMinoux法とその改良法では，Scott法と0.3%程度以内の差の近似解を算出できることを示した。また，提案した近似解法は，ノード100までの問題に対して10秒以内で近似解を求めることができるため，大規模な問題に対しても有効な解法であることを示した。

参考文献

1）R. Ahuja and V. Murty. New lower planes for the network design problem. *Networks*, Vol. 17, pp. 113-127, 1987.

2）T. Boffey and A. I. Hinxman. Solving the optimal network problem. *European Journal of Operational Research*, Vol. 3, pp. 386-393, 1979.

3）D. Boyce, A. Farhi, and R. Weischedel. Optimal network problem: Branch-and-bound algorithm. *Environment and Planning*, Vol. 5, pp. 519-533, 1973.

4) R. Dionne and M. Florian. Exact and approximate algorithms for optimal network design. *Networks*, Vol. 9, pp. 37-59, 1979.

5) G. Gallo. Lower planes for the network design problem. *Networks*, Vol. 13, pp. 411-425, 1983.

6) H. Hoang. A computational approach to the selection of an optimal network. *Management Science*, Vol. 19, pp. 488-498, 1973.

7) M. Minoux. Multiflots de coût minimal avec fonctions de coût concaves. *Annales des Télécommunications*, Vol. 31, pp. 77-92, 1976.

8) M. Minoux. Optimization et planification des reśeaux de telecommunications. In *Optimization Techniques, Lecture Notes in Computer Science 40*, pp. 419-430. Springer Verlag, 1976.

9) M. Minoux. Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, Vol. 19, pp. 313-360, 1989.

10) A. Scott. The optimal network problem: Some computational procedures. *Transportation Research*, Vol. 3, pp. 201-210, 1969.

11) P. Steenbrink. *Optimization of Transport Networks*. John Wiley & Sons, 1974.

12) R. Wong. Probabilistic analysis of a network design problem heuristic. *Networks*, Vol. 15, pp. 347-363, 1985.

13) 西村昂, 日野泰雄, 最適ネットワーク構成に関する一考察. 土木学会論文報告集, Vol. 250, pp. 85-97, 1976.

14) 森津秀夫. 最適交通網構成手法に関する基礎的手法. 神戸大学, 1984.

15) 片山直登. ネットワークデザイン問題の近似解法. 流通経済大学流通情報学部開校記念論文集, pp. 171-191, 1997.

16) 片山直登, 春日井博. ラグランジュ緩和法を用いた予算制約をもつネットワークデザイン問題の解法. 日本経営工学会誌, Vol. 46, pp. 21-27, 1995.