Vol.2009-MPS-75 No.2
2009/9/10

最小極大フローに対する効率的なアルゴリズム

陳　文　傑[†1]　篠村　真弘[†1]　施　建　明[†1,∗1]

最小極大フロー問題とは, ネットワークでの極大フローの中で最も流量の小さいフローを見つけ出す問題である. この問題は $\mathcal{NP}$ 困難な問題であることが知られている. 本研究では, 最小極大フロー問題の目的関数と制約条件をそれぞれ線形関数を用いて定式化し, 本来の問題を凸集合上での凹最小化問題に帰着させる. 更に, 外部近似法を基にした新たなアルゴリズムを提案し, アルゴリズムを数値実験で検証する.

The Effective Algorithm For Minimum Maximal Network Flow

Wenjie Chen,[†1] Masahiro Shiromura[†1]
and Jianming Shi[†1,∗1]

The minimum maximal flow problem that is one of the $\mathcal{NP}$-hard problems is a problem of searching the minimum flow among the maximal flows in a given directed network. In this research, we express the objective function and constraints of the minimum maximal flow problem by using linear functions and formulate this problem as a concave minimization problem over a convex set. Moreover, we propose an algorithm based upon the outer approximation method and check the algorithm with numerical experiments.

1. はじめに

ネットワークフロー問題は 1950-60 年代の Ford と Fulkerson より提出されて以来, 50 年近く経っているにもかかわらず, 依然として活発に研究がされており, この分野の代表的な問題として最大流問題と最小費用流問題がある. 従来のネットワーク理論はフローをコントロールできるという条件の下で発展したが, 最小極大フロー問題はフローをコントロールできない前提で提案された問題である[1)]. 最小極大フロー問題とは, ネットワークのすべての極大フローの中で, 最も流量の小さいフローを見つけ出す問題であり, $\mathcal{NP}$ 困難な問題であることが知られている[2)].

最小極大フロー問題に関する研究としては, 1997 年に Shi と Yamamoto が最小極大フロー問題を定義し, 下界法でこの問題を解決するためのアルゴリズムを提案した[3)]. また 2003 年に Gotoh, Thoai と Yamamoto は大域最適法のアルゴリズムを示した[4)]. その後, 2007 年に Yamamoto と Zenke は外部近似法でのアルゴリズムを提案した[5)].

下界法では, ネットワークの各辺にフローの下限を加え, 全域探索法で最小極大フローを見つける. 大域最適法では, 最小極大問題を大域最適化問題に変換し, 局所探索法と分枝限定法を用い, 最小極大フローを見つける. 外部近似法では, ギャップ関数を用いて極大フローの実行可能領域を定義し, 本来の問題を $D.C.$ 最適化問題に変換し, $D.C.$ 問題に対する外部近似法と局所探索法を用いて最小極大フローを見つける. しかしながら, 下界法や外部近似法などの既存研究で計算する場合では, ネットワークの枝の本数を次元とする凸多面体の端点を計算する必要があるので, その次元が高くなると, 莫大な計算時間がかかる.

本研究では, 最小極大フロー問題の目的関数と制約条件をそれぞれ線形関数を用いて定式化し, 本来の問題をいくつかの線形計画問題を解くことに帰着させる. 更に, これに基づくアルゴリズムを提案する. 線形計画問題に対する多項式アルゴリズムは知られているので, ネットワークの枝の本数が増加しても, 最小極大フロー問題を解く時間はそれほど増加しない.

本研究報告の構成を次に示す. 第 2 節では扱う問題に関する基礎概念と理論について解説を行い, 外部近似法に関する説明をする. 第 3 節ではアルゴリズムを提案するための数式と定理について紹介し, アルゴリズムの提案と説明をする. 第 4 節では数値実験とその結果について示す. 第 5 節では結論を記述する.

2. 基礎概念と問題定義

最小極大フロー問題はネットワークにおけるフロー問題の１つであるので, ネットワークについての定義と記号をあらかじめ定める.

2.1 ネットワークの基礎概念

ネットワーク $N = (V, E, s, t, c)$ は, 有向グラフ $G = (V, E)$ の各枝 e に正の実数の容量 $c(e)$ が割り振られているグラフである. ただし, V は $m + 2$ 個の点集合であり, E は n 個の枝集合であり, s, t はネットワークのソース, シンクであり, c は各枝の容量である. また,

†1 室蘭工業大学情報工学科
Department of Computer Science and Systems Engineering,Muroran Institute of Technology
∗1 現在, この著者は科学文部省に研究補助金を支えされている.Email:shi@mmm.muroran-it.ac.jp

© 2009 Information Processing Society of Japan

$(s,t) \in V$ である.

定義 1

ネットワーク N の枝集合から非負実数集合への関数 $f : E \to \boldsymbol{R}_+$ が以下の条件を満たすとき, f を N のフローという.

D1.1) 任意の枝 $e \in E$ に対して, $c\,(e) \geq f\,(e) \geq 0$ が成り立つ.

D1.2) s, t を除く任意の V の点 v に対して, 流量保存則 (流入量=流出量) が成り立つ. 数式で書くと, 次のようになる.

$$\sum_{e=(u,v)\in E} f(e) = \sum_{e=(v,w)\in E} f(e)$$

また, s から流出するフローの正味量 F は t に流入するフローの正味量に等しい. この F をフロー f の流量といい, $\mathrm{val}\,(f)$ と書く.

s を含み, t を含まない点の部分集合 U に対するカット $C\,(U, V-U)$ を (s,t) カットという. (s,t) カット $C\,(U, V-U)$ の容量を $\mathrm{cap}\,(U)$ で表す.

2.2 最小極大フロー問題の形式

$m \times n$ 次元の行列 $\boldsymbol{A}$ の各要素を次のように定義しておく.

$$a_{v,e} = \begin{cases} 1 & (\text{点 } v \text{ から出る枝 } e \text{ の場合}) \\ -1 & (\text{点 } v \text{ に入る枝 } e \text{ の場合}) \\ 0 & (\text{それ以外}) \end{cases}$$

そうすると, フローの条件 **D1.1)** と **D1.2)** を満たすフローは集合 $\boldsymbol{X}$ で書き換えられる.

$$\boldsymbol{X} = \{\boldsymbol{x} \in \boldsymbol{R}^n \mid \boldsymbol{A}\boldsymbol{x} = 0, c \geq \boldsymbol{x} \geq 0\} \qquad (1)$$

ただし, $\boldsymbol{R}^n$ は n 次元の列ベクトルの集合であり, $\boldsymbol{x}$ はフローである.
即ち, $\boldsymbol{X}$ はフローの実行可能領域である.

本研究では, 議論を簡潔するため, 各枝の容量 $c(e)$ は正の整数であることを仮定する. また, ソース s に入る経路とシンク t から出る経路が存在しないこととサイクルが存在しないことも仮定する.

さらに, $\boldsymbol{d}(\in \boldsymbol{R}^n)$ の各要素を次のように定義しておく.

$$d_e = \begin{cases} 1 & (\text{ソース } s \text{ から出る枝 } e \text{ の場合}) \\ -1 & (\text{ソース } s \text{ に入る枝 } e \text{ の場合}) \\ 0 & (\text{それ以外}) \end{cases}$$

上記の定義より, 従来の最大流問題は次の形式となる.

$$\left|\begin{array}{ll} \max & \langle \boldsymbol{d}, \boldsymbol{x} \rangle \\ s.t. & \boldsymbol{x} \in \boldsymbol{X} \end{array}\right.$$

ただし, $\langle \cdot, \cdot \rangle$ は内積である.
極大フローを次のように定義する.

定義 2

ネットワーク N に対して, 以下の **D2.1)** と **D2.2)** を同時に満たす N の実行可能フロー $\boldsymbol{z}$ が存在しないとき, N の実行可能フロー $\boldsymbol{x}$ を N の極大フローという.

D2.1) 任意の枝 $e \in E$ に対して, $\boldsymbol{z} \geq \boldsymbol{x}$ が成り立つ.

D2.2) $\boldsymbol{z} \neq \boldsymbol{x}$ が成り立つ.

定義 2 を満たすフローの集合を $\boldsymbol{X_E}$ で表すと, $\boldsymbol{X_E}$ は極大フローの集合である[6].

$$\boldsymbol{X_E} = \{\boldsymbol{x} \in \boldsymbol{X} \mid \boldsymbol{z} \geq \boldsymbol{x} \text{ かつ } \boldsymbol{z} \neq \boldsymbol{x} \text{ のような } \boldsymbol{z} \in \boldsymbol{X} \text{ が存在しない }\} \qquad (2)$$

$\boldsymbol{X_E}$ を用いて, 最小極大フロー問題は次のようになる.

$$(MMF)\left|\begin{array}{ll} \min & \langle \boldsymbol{d}, \boldsymbol{x} \rangle \\ s.t. & \boldsymbol{x} \in \boldsymbol{X_E} \end{array}\right. \qquad (3)$$

上記の問題形式より, 最小極大フローは $\boldsymbol{X_E}$ の中で最小流量フローである.

3. 提案するアルゴリズム

まずは, 提案するアルゴリズムに使用される数式と記号を紹介する. 次に, アルゴリズムを提案する. 最後に, このアルゴリズムの正当性と収束性について証明する.

3.1 数 式 紹 介

次のような関数 $\boldsymbol{r}(\boldsymbol{x})$ を定義する.

$$\boldsymbol{r}(\boldsymbol{x}) := \max\{\langle \boldsymbol{e}, \boldsymbol{y} - \boldsymbol{x} \rangle \mid \boldsymbol{y} \geq \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X}\} \qquad (4)$$

© 2009 Information Processing Society of Japan

Vol.2009-MPS-75 No.2
2009/9/10

ただし, $\boldsymbol{e}$ はすべての要素が 1 である n 次元列ベクトルである.
上記の式 (4) より, 任意の $\boldsymbol{x} \in \boldsymbol{X}$ に対して $\boldsymbol{r}(\boldsymbol{x}) \geq 0$ であることが明らかである. なぜならば, 制約条件の $\boldsymbol{y} \geq \boldsymbol{x}$ より, $\boldsymbol{r}(\boldsymbol{x})$ 関数の値は $\langle \boldsymbol{e}, \boldsymbol{y} - \boldsymbol{x} \rangle \geq 0$ となる.

定理 1

$\boldsymbol{r}(\boldsymbol{x})$ は線形区分的な凹関数である.

証明：$\boldsymbol{r}(\boldsymbol{x})$ 関数は線形計画の双対定理とパラメトリック線形計画の性質を満たすため, 定理が成立する[7]. □

定理 2

$\boldsymbol{r}(\boldsymbol{x}) = 0$ が成立する必要十分条件は $\boldsymbol{x} \in \boldsymbol{X_E}$ である.

証明：まず, 必要性を証明する.
$\boldsymbol{x} \notin \boldsymbol{X_E}$ と仮定する. 即ち, フロー $\boldsymbol{x}$ は極大フローではない. 極大フローの定義 (定義 2) より, $\boldsymbol{y} \geq \boldsymbol{x}, \boldsymbol{y} \neq \boldsymbol{x}$ を満たす $\boldsymbol{X}$ の実行可能解 $\boldsymbol{y}$ は存在する. 即ち, $\boldsymbol{y} - \boldsymbol{x} \neq 0$ が成り立つ. よって, フロー $\boldsymbol{y}$ は $\boldsymbol{r}(\boldsymbol{x})$ 関数 (式 4) の実行可能領域に含まれる. $\boldsymbol{r}(\boldsymbol{x})$ 関数の値は $\langle \boldsymbol{e}, \boldsymbol{y} - \boldsymbol{x} \rangle \neq 0$ であることが成り立つ. 即ち, $\boldsymbol{r}(\boldsymbol{x}) \neq 0$ が成り立つ. 背理法より, 定理の必要性が成立する.
同様に, 十分性は明らかである. □
定理 2 より, MMF 問題を次の形式に書き直すことができる.

$$(MMF) \left| \begin{array}{ll} \min & \langle \boldsymbol{d}, \boldsymbol{x} \rangle \\ s.t. & \boldsymbol{x} \in \boldsymbol{X}, \boldsymbol{r}(\boldsymbol{x}) = 0 \end{array} \right. \tag{5}$$

系 1

MMF 問題において, $\langle \boldsymbol{d}, \boldsymbol{x} \rangle$ が最小になるのは $\boldsymbol{r}(\boldsymbol{x}) = 0$ のときに限ることは明らかである. 即ち, $\boldsymbol{r}(\boldsymbol{x})$ は MMF 問題の極大フローであるかを判断する関数である.

ある実数 $t > 0$ に対して, 次のペナルティ関数 $P(t)$ を定義する.

$$P(t) := \min\{\langle \boldsymbol{d}, \boldsymbol{x} \rangle + t \cdot \boldsymbol{r}(\boldsymbol{x}) \mid \boldsymbol{x} \in \boldsymbol{X}\} \tag{6}$$

$P(t)$ に対して, 次の定理 3 が成り立つ.

定理 3

十分に大きい t に対して, もし $\boldsymbol{x}^* \in \arg\min P(t)$ ならば, $\boldsymbol{r}(\boldsymbol{x}^*) = 0$ が成り立つ.

証明：$\boldsymbol{r}(\boldsymbol{x}^*) \neq 0$ と仮定する. $\boldsymbol{r}(\boldsymbol{x}) \geq 0$ より, $\boldsymbol{r}(\boldsymbol{x}^*) > 0$ が成り立つ. また, ある $\boldsymbol{y} \in \boldsymbol{X_E}$ が存在すると仮定する. 定理 2 より, $\boldsymbol{r}(\boldsymbol{y}) = 0$ が成り立つ.
$t = (\max\{\langle \boldsymbol{d}, \boldsymbol{x} \rangle \mid \boldsymbol{x} \in \boldsymbol{X}\} + 1)/\boldsymbol{r}(\boldsymbol{x}^*)$ とおく. 明らかに $t > 0$ が成り立つ.
よって,

$$\begin{aligned} \langle \boldsymbol{d}, \boldsymbol{x}^* \rangle + t \cdot \boldsymbol{r}(\boldsymbol{x}^*) &= \langle \boldsymbol{d}, \boldsymbol{x}^* \rangle + \max\{\langle \boldsymbol{d}, \boldsymbol{x} \rangle \mid \boldsymbol{x} \in \boldsymbol{X}\} + 1 \\ &> \max\{\langle \boldsymbol{d}, \boldsymbol{x} \rangle \mid \boldsymbol{x} \in \boldsymbol{X}\} \\ &\geq \langle \boldsymbol{d}, \boldsymbol{y} \rangle \\ &= \langle \boldsymbol{d}, \boldsymbol{y} \rangle + t \cdot \boldsymbol{r}(\boldsymbol{y}) \end{aligned}$$

上記の式より, $\boldsymbol{x}^*$ が $P(t)$ の最適解ではないことになる.
背理法より, 定理が成立する. □
パラメーター $\alpha > 0$ とおく. 次の実行可能領域を $\boldsymbol{X}(\alpha)$ を定義する.

$$\boldsymbol{X}(\alpha) := \boldsymbol{X} \cap \{\boldsymbol{x} \in \boldsymbol{R}^n \mid \langle \boldsymbol{d}, \boldsymbol{x} \rangle \geq \alpha\} \tag{7}$$

また, $\boldsymbol{X}(\alpha)$ を基にした問題 $Q(\alpha)$ を次のように定義する.

$$Q(\alpha) := \min\{\langle \boldsymbol{d}, \boldsymbol{x} \rangle \mid \boldsymbol{x} \in \boldsymbol{X}(\alpha)\} \tag{8}$$

$Q(\alpha)$ 関数の役割はアルゴリズムの各段階において制約される領域の中で, 最小流量フローを見つけるということである.

3.2 アルゴリズムの提案

これから提案するアルゴリズムでは, $\boldsymbol{r}(\boldsymbol{x})$ 関数は実行可能領域 $\boldsymbol{X}$ の下で定義される関数であることより, $\boldsymbol{x}$ は常に $\boldsymbol{X}$ に含まれる. また, 提案するアルゴリズムでは, 問題 (5) をいくつかの線形計画問題に帰着させることにより, n が大きくなっても, 見つける時間はそれほど多くはならない.

提案する MMF へのアルゴリズム

ステップ 0: $\boldsymbol{r}(0)$ を解き, $\boldsymbol{r}(\boldsymbol{y}^1) = 0$ のような $\boldsymbol{y}^1$ を得る. パラメーター $\varepsilon > 0$ を設定し, $k := 1$ とおく.

© 2009 Information Processing Society of Japan

Vol.2009-MPS-75 No.2
2009/9/10

ステップ 1: $\alpha_k := \langle \boldsymbol{d}, \boldsymbol{y}^k \rangle - \varepsilon$ とおく. $Q(\alpha_k)$ を解き, $\boldsymbol{q}^k$ を得る.
ステップ 2: もし $\boldsymbol{r}(\boldsymbol{q}^k) = 0$ ならば, $\boldsymbol{y}^k := \boldsymbol{q}^k, k := k+1$ とおき, ステップ 1 に戻る.
ステップ 3: $\max\{\langle \boldsymbol{e}, \boldsymbol{w} \rangle \mid \langle \boldsymbol{d}, \boldsymbol{w} \rangle = \langle \boldsymbol{d}, \boldsymbol{q}^k \rangle, \boldsymbol{w} \in \boldsymbol{X}\}$ を解き, $\boldsymbol{w}^k$ を得る.
ステップ 4: もし $\boldsymbol{r}(\boldsymbol{w}^k) = 0$ ならば, $\boldsymbol{y}^k := \boldsymbol{w}^k, k := k+1$ とおき, ステップ 1 へ.
ステップ 5: $\boldsymbol{v}^k \in \boldsymbol{X_E} \cap (\boldsymbol{X} \cap \{\boldsymbol{x} \mid \langle \boldsymbol{d}, \boldsymbol{x} \rangle = \langle \boldsymbol{d}, \boldsymbol{q}^k \rangle\}$ を探索する.
ステップ 6: もし $\boldsymbol{v}^k$ が存在しなければ, 終了する.($\boldsymbol{q}^{k-1}$ は最適解である.)
ステップ 7: 存在するならば, $\boldsymbol{y}^k := \boldsymbol{v}^k, k := k+1$ とおき, ステップ 1 に戻る.

提案するアルゴリズムによって見つけられる最小極大フローを $\boldsymbol{x}^*$ とおく.

次に, 2 次元のイメージ図 1 を用いて最小極大フローを見つける手順を説明する.

まずは, 初期値を設定しておく. $\boldsymbol{X}$ はフローの実行可能領域であり, $\boldsymbol{X_E}$(太線) は極大フローの実行可能領域である. また, 最大フロー $\boldsymbol{y}^1$ を見つける (図 1 の (a)). 次に, $\boldsymbol{y}^1$ の値を値 ε だけ減少させ, 新しいフローと極大フロー領域 $\boldsymbol{X}(\alpha_1)$ を得る. $Q(\alpha_1)$ 関数で最小の点を見つけ, $\boldsymbol{r}(\boldsymbol{x})$ 関数でこの点は極大フローであるかを判断する (図 1 の (b)). そうであるならば, 同様に繰り返していく (図 1 の (c) と (d)). そうでないならば, $\max\{\langle \boldsymbol{e}, \boldsymbol{w} \rangle \mid \langle \boldsymbol{d}, \boldsymbol{w} \rangle = \langle \boldsymbol{d}, \boldsymbol{q}^k \rangle, \boldsymbol{w} \in \boldsymbol{X}\}$ でこのフローの最大点 $\boldsymbol{x}^*$ を見つけ, $\boldsymbol{r}(\boldsymbol{x}^*)$ 関数で極大フローであるかを確認する (図 1 の (e)). これを ε だけ減少させた後のフローに極大フローが含まれなくなるまで続け, そうなったならば, その値以下には極大フローが存在しないことになるので, 直前の $\boldsymbol{q}^{k-1}$ を $\boldsymbol{x}^*$ とする (図 1 の (f)). 以上の動作より, フロー $\boldsymbol{x}^*$ は MMF 問題 (5) の最小極大フローであることは明らかである.

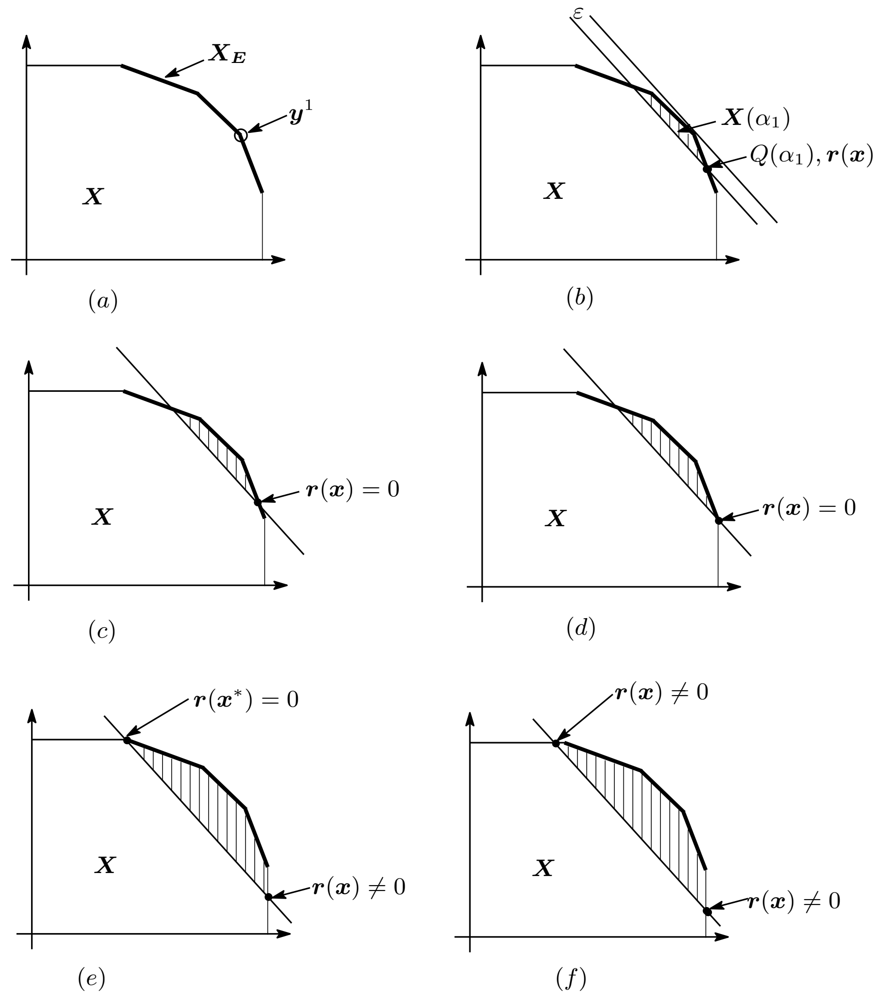

図 1 最小極大フローを見つける手順

© 2009 Information Processing Society of Japan

定理 4

もし $\boldsymbol{x}^*$ が MMF 問題の最適解であれば, $\boldsymbol{x}^*$ も $Q(\langle \boldsymbol{d}, \boldsymbol{x}^* \rangle)$ の最適解である.

証明：$\boldsymbol{q}^{k-1}$(つまり $\boldsymbol{x}^*$) は MMF 問題の最適解とする. このとき, ステップ 1 の $Q(\langle \boldsymbol{d}, \boldsymbol{x}^* \rangle)$ にて探索領域は $\boldsymbol{X}(\langle \boldsymbol{d}, \boldsymbol{x}^* \rangle) = \boldsymbol{X} \cap \{\boldsymbol{x} \in \boldsymbol{R} \mid \langle \boldsymbol{d}, \boldsymbol{x} \rangle \geq \langle \boldsymbol{d}, \boldsymbol{x}^* \rangle\}$ である. よって領域内の任意の $\boldsymbol{x}$ に対しても, $\langle \boldsymbol{d}, \boldsymbol{x} \rangle \geq \langle \boldsymbol{d}, \boldsymbol{x}^* \rangle$ が成立する. $\boldsymbol{x}^*$ 自身も領域内に含まれるため, $Q(\langle \boldsymbol{d}, \boldsymbol{x}^* \rangle) = \min\{\langle \boldsymbol{d}, \boldsymbol{x} \rangle \mid \boldsymbol{x} \in \boldsymbol{X}(\langle \boldsymbol{d}, \boldsymbol{x}^* \rangle)\} = \langle \boldsymbol{d}, \boldsymbol{x}^* \rangle$ となる. よって, $\boldsymbol{x}^*$ も $Q(\langle \boldsymbol{d}, \boldsymbol{x}^* \rangle)$ の最適解となる. □

続いて, 提案するアルゴリズムの正当性と収束性について証明する.

定理 5

提案するアルゴリズムが MMF 問題 (5) の最適解を見つけるためのステップ数は有限である.

証明：

(正当性)　アルゴリズムの最終段階のステップ 5 からステップ 7 の中で, フロー $\boldsymbol{q}^k$ に対して, もしステップ 4 での $\boldsymbol{r}(\boldsymbol{w}^k) \neq 0$ ならば, 流量が $\langle \boldsymbol{d}, \boldsymbol{q}^k \rangle$ であるフローの中に, 極大フローが存在するかを確認する. もしそのようなフローが存在しないならば, フローの整数性と $\boldsymbol{X_E}$ の連結性より, 明らかに前段階の $\boldsymbol{r}(\boldsymbol{x}) = 0$ を満たすフロー $\boldsymbol{q}^{k-1}$ が最適解となる.

(収束性)　アルゴリズムのステップ 0(初期化) で, $\boldsymbol{r}(\boldsymbol{y}^1) = 0$ のようなフロー $\boldsymbol{y}^1$ を見つける. フロー $\boldsymbol{y}^1$ が $\boldsymbol{y}^1 \leq \boldsymbol{c}$ を満たすため, $\boldsymbol{y}^1$ の値は有限であることは自明である. ステップ 1 で, $\boldsymbol{y}^1$ の値は整数 $\varepsilon > 0$ ずつ減少していく. また, 最小のフローの値は 0 となることとフローの整数性より, 極大フローの判断回数は多くとも $\langle \boldsymbol{d}, \boldsymbol{y}^1 \rangle$ 回を超えない. さらに, 流量一定のフローは極大フローであるかを探索する回数は有限であることより, ステップ 5 の回数は有限であることは明らかである. □

4. 数 値 実 験

本節では, 提案したアルゴリズムについて, 最適解がわかっている 3 つのネットワークで検証を行う. アルゴリズムのパラメーターを $\varepsilon = 1$ と設定し, それぞれの実験結果についてまとめる. 実験環境を以下に示す.

表 1　実験環境

使用言語	CPU	Memory	OS
MATLAB6.5.1	Pentium(R) Dual 1.60 GHz	0.99GB	Windows XP

例題 4.1 ($|\boldsymbol{V}| = 4, |\boldsymbol{E}| = 5$) のネットワーク

ネットワークでは, 点集合 $\boldsymbol{V}$ の数は 2+2 であり, 枝集合 $\boldsymbol{E}$ の数は 5 であり, 各枝上の数値は各枝の容量である (図 2 の (4.1.a)). プログラムを実行し, 発見した最小極大フローの結果を以下に示す (図 2 の (4.1.b)).

- 最小極大フロー $\boldsymbol{x}^*$=$(4,4,2,2,6)^\top$,
- 最小極大フローの値 val($\boldsymbol{x}^*$)= 8.

例題 4.2 ($|\boldsymbol{V}| = 6, |\boldsymbol{E}| = 10$) のネットワーク

ネットワークでは, 点集合 $\boldsymbol{V}$ の数は 4+2 であり, 枝集合 $\boldsymbol{E}$ の数は 10 であり, 各枝上の数値は各枝の容量である (図 2 の (4.2.a)). プログラムを実行し, 発見した最小極大フローの結果を以下に示す (図 2 の (4.2.b)).

- 最小極大フロー $\boldsymbol{x}^*$=$(6,3,4,2,0,0,7,1,2,8)^\top$,
- 最小極大フローの値 val($\boldsymbol{x}^*$)=9.

例題 4.3 ($|\boldsymbol{V}| = 8, |\boldsymbol{E}| = 14$) のネットワーク

ネットワークでは, 点集合 $\boldsymbol{V}$ の数は 6+2 であり, 枝集合 $\boldsymbol{E}$ の数は 14 であり, 各枝上の数値は各枝の容量である (図 2 の (4.3.a)). プログラムを実行し, 発見した最小極大フローの結果を以下に示す (図 2 の (4.3.b)).

- 最小極大フロー $\boldsymbol{x}^*$=$(6,4,4,2,3,5,3,6,2,0,2,0,6,4)^\top$,
  ただし, 最小極大フローがいくつかあり, すべての枝の流量が整数ではない最小極大フローも存在する. ここで, すべての枝の流量が整数である最小極大フローを示す.
- 最小極大フローの値 val($\boldsymbol{x}^*$)=10.

© 2009 Information Processing Society of Japan

Vol.2009-MPS-75 No.2
2009/9/10

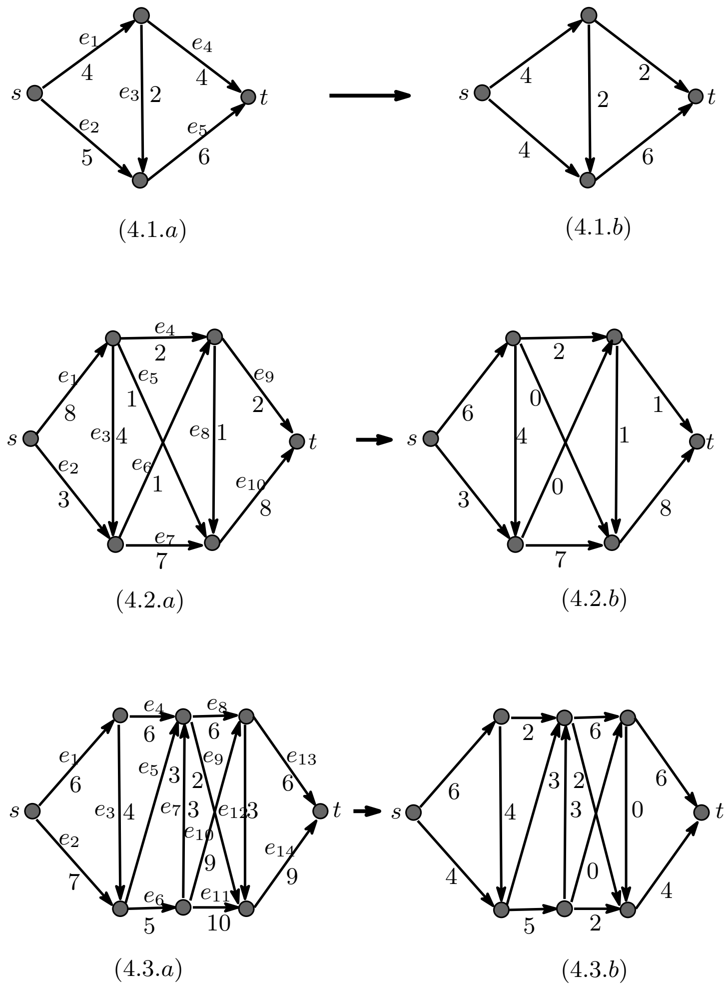

図 2 最小極大フローの実行例

5. 結　　論

最小極大フロー問題を一連の線形計画問題に変換し, 外部近似法に基づく $\mathcal{NP}$ 困難な MMF 問題に対するアルゴリズムを提案した. 最適解のわかっている例題に対して, 実験で正しい解を得られることが確認できたことより, 提案したアルゴリズムが正常に動作していることを確認した.

参 考 文 献

1) 浅野孝夫:情報の構造ネットワークアルゴリズムとデータ構造, 日本評論社 (1994).
2) Lawler, E.: *Combinatorial optimization networks and matroids*, Dover publications, INC, Mineola, NY (2001).
3) Shi, J. and Yamamoto, Y.: A global optimization method for minimum maximal flow problem, *ACTA Mathematica Vietnamica*, Vol.22, No.1, pp.271–287 (1997).
4) Gotoh, J., Thoai, N. and Yamamoto, Y.: Global optimization method for solving the minimum maximal flow problem, *Optimization Methods and Software*, Vol.18, No.4, pp.395–415 (2003).
5) Yamamoto, Y. and Zenke, D.: Out approximation method for the minimum maximal flow problem, *Journal of Operations Research Society of Japan*, Vol.50, No.1, pp.14–30 (2007).
6) Shigeno, M., Takahashi, I. and Yamamoto, Y.: Minimum maximal flow problem: an optimization over the efficient set, *Journal of Global Optimization*, Vol.25, No.1, pp.425–443 (2003).
7) Marey, K.G.: *Linear Complementary Linear and Nonlinear Programming Sigma Series in Applied Mathematics 3*, Heldermann Verlag,Berlin (1988).
8) Horst, R. and Tuy, H.: *Global Optimization Deterministic Approaches*, Springer Verlag, (1993).

© 2009 Information Processing Society of Japan