数理解析研究所講究録
第 471 巻 1982 年 230-258

平面グラフの多種フローについて

東北大 工学部　松本 和彦
Kazuhiko Matsumoto
西関 隆夫
Takao Nishizeki

1. まえがき

ネットワークフロー問題に対しては多くの研究がなされている．特に，1種フローあるいは2種フローの場合には，いわゆる最大フロー・最小カット定理の成り立つことが知られており，最大1種フローあるいは最大2種フローを求める効率の良いアルゴリズムが種々工夫されている．これに対し，多種フロー問題は通信網・交通網制御，ＶＬＳＩの配線等への応用上重要であるにもかかわらず，一般のグラフでフローの種類が3個以上の場合に対しては効率の良いアルゴリズムは知られていない．しかし，無向平面グラフでソース及びシンクがすべて外面にある場合は効率の良いアルゴリズムが筆者らによって与えられている[8]．

今回は次のような2種類の平面グラフに対して，多種フロ

ーを見つける多項式時間アルゴリズムを与える：

(i) ソース・シンク対の間に枝を付加してもグラフが平面である場合.

(ii) ある2つの面の境界上にのみソース・シンク対がある場合（ただし対応するソースとシンクは同じ面の境界上にある.

これらのアルゴリズムは，与えられた要求量を持つ多種フローが存在するかどうかを判定し，もし存在するならば，具体的にその多種フローを求めている．上の(ii)の場合は，筆者らの前回のアルゴリズム(8)で取扱えるグラフを真に包含している.

2. 準備

まず記号，用語の定義をする．フローネットワーク $N=(G, P, c)$ は3つ組である．ここで

(i) $G=(V, E)$ は有限単純無向グラフであり，Vは点集合，Eは枝集合である.

(ii) Pはソース・シンク対 (s_i, t_i) の集合である．ここでソースs_iとシンクt_iはGの相異なる点である.

(iii) $c: E \rightarrow R^+$ は容量関数である．（R（またはR^+）は（正

の）実数の集合である.）

Gが平面グラフであるとき，$N=(G, P, c)$は平面ネットワークという. またNがk個のソース・シンク対を持つとき，すなわち$|P|=k$であるとき，Nはk-ネットワークという.

Nの各ソース・シンク対(s_i, t_i)には非負の要求量$d_i \geq 0$が与えられている. グラフGは無向であるが，便宜上各枝に適当な向きを付けて，枝に流れるフローの値はその向きと同じときは正，逆向きのときは負とする. k個の要求量$d_1, d_2, \cdots, d_k$を持つk種フローとは，次の(a)及び(b)を満足する関数$f_i : E \to R$の集合$\{f_1, f_2, \cdots, f_k\}$である.

(a)：容量条件.　すべての$e \in E$に対して

$$\sum_{i=1}^{k} |f_i(e)| \leq c(e).$$

(b)：保存則.　どのf_iでも，各$v \in V-\{s_i, t_i\}$に対して$IN(f_i, v)=OUT(f_i, v)$が成り立ち，また$OUT(f_i, s_i) - IN(f_i, s_i) = IN(f_i, t_i) - OUT(f_i, t_i)$が成り立つ. ここで$IN(f_i, v)$は$v$に流れ込む第$i$種のフロー$f_i$の総量を表わし，$OUT(f_i, v)$は$v$から流れ出る第$i$種のフロー$f_i$の総量を表わす.

$E(X ; Y)$は一方の端点が$X \subset V$に，他方が$Y \subset V$に含まれる枝の集合を表わすことにする. 以下にいくつかの記号を列挙する.

$$E(X) = E(X; V - X) \qquad (\text{カット})$$

$$c(X; Y) = \sum_{e \in E(X; Y)} c(e)$$

$$c(X) = \sum_{e \in E(X)} c(e) \qquad (\text{カットの容量})$$

$$D(X; Y) = \{ i \mid 1 \le i \le k, \ |\{s_i, t_i\} \cap X| = |\{s_i, t_i\} \cap Y| = 1 \}$$

$$D(X) = D(X; V - X)$$

$$d(X; Y) = \sum_{i \in D(X; Y)} d_i$$

$$d(X) = \sum_{i \in D(X)} d_i \qquad \left(\begin{array}{l}\text{カットXにより切ら}\\ \text{れる要求量の総和}\end{array}\right)$$

明らかに $E(X) = E(V - X)$, $c(X) = c(V - X)$, $D(X) = D(V - X)$, $d(X) = d(V - X)$ が成立する.

各 $X \subset V$ に対して $c(X) \ge d(X)$ が成立するとき，ネットワークNはカット条件を満足するという．明らかに，カット条件はk-ネットワークにおいて与えられた要求量を持つk種フローが存在するための必要条件であるが，必らずしも十分ではない．しかし，次の定理が証明されている．

［定理1］（Seymour[5]）　$N = (G, P, c)$ は平面k-ネットワークで，すべてのソース・シンク対(s_i, t_i)に対してs_iとt_iの間に枝を付加しても，そのネットワークが平面であるとする．このとき，要求量 $d_1, d_2, \cdots, d_k$ を持つk種フロー

がNに存在するための必要十分条件は，Nがカット条件を満足することである．

［定理2］（岡村[3]）　$N=(G, P, c)$は平面k-ネットワークで，ソース$s_1, \cdots, s_l$及びシンク$t_1, \cdots, t_l$がGの外周B_0上に，またソース$s_{l+1}, \cdots, s_k$及びシンク$t_{l+1}, \cdots, t_k$がGのある面の境界B_1上にあるとする．このとき要求量$d_1, \cdots, d_k$を持つk種フローがNに存在するための必要十分条件は，Nがカット条件を満足することである．

$X \subset V$のとき，Gが$V-X$に含まれるすべての点と，$E(X)$に含まれるすべての枝を除去して得られるGの部分グラフを$G|X$で表わすことにする．次の2つの補題が成立する．証明は文献(4)に与えられているので省略する．

［補題1］[4]　グラフ$G=(V, E)$は連結であるとする．すべての$X \subset V$に対して$c(X) \geq d(X)$が成立するための必要十分条件は，$G|X$と$G|(V-X)$のどちらも連結であるような$X \subset V$に対して$c(X) \geq d(X)$が成立することである．

［補題2］[4]　任意のグラフ$G=(V, E)$において

$$c(X \cap Y)+c(X \cup Y)=c(X)+c(Y)-2c(X-Y; Y-X),$$

$$d(X \cap Y)+d(X \cup Y)=d(X)+d(Y)-2d(X-Y; Y-X)$$

が成立する．ただし$X, Y \subset V$である．

3．多種フローアルゴリズム(1)

本節では定理１で述べたネットワーク，すなわち各ソース・シンク対の間に枝を付け加えても平面性を失わないネットワークに対して多種フローを求めるアルゴリズムを与える．

もとのグラフを$G_0=(V, E)$とし，各ソース・シンク対の間に枝を付加したグラフをG_aと表わすことにする．またもとのネットワークを$N_0=(G_0, P, c)$，新しいネットワークを$N_a=(G_a, \phi, c_a)$とする．ただしG_aにおける，s_i-t_i間に付加された枝をe_{ai} $(1\leq i\leq k)$と表わすことにし，容量は$c_a(e_{ai})=-d_i\ (\leq 0)$とする．むろん$e\in E$については$c_a(e)=c(e)$とする．

3.1 許容性の判定

ここでは，ネットワークNに，与えられた要求量d_1, d_2, $\cdots$, d_kを持つk種フローが存在するかどうか（すなわち許容性）を判定する方法を述べる．

まず，ネットワークN_0においてカット$E_0(X)$の余裕$m_0(X)$を次のように定義する：

$$m_0(X)=c_0(X)-d_0(X)$$

（ネットワーク名N_iと同じ添字を用いて$m_i(X)$, $c_i(X)$, $d_i(X)$

などと書くことにする.）この余裕を用いると，定理1は次のように書き直せる:

　N_0において要求フローを実現する多種フローが存在するための必要十分条件は

$$m_0(X) \geq 0$$

　がすべての $X \subset V$ に対して成立することである.

これをN_aにおいて考えてみる. N_aでのカットの容量$c_a(X)$は $c_a(X) = c_0(X) - d_0(X)$ である. なぜなら，Xにソースs_iが含まれ，$V - X$ にシンクt_iが含まれていれば，G_aでは$e_{ai} \in E_a(X)$であり，ソース・シンク対がともにXに，あるいは$V-X$ に含まれているようなものは$e_{ai} \notin E_a(X)$ であるからである. したがって，N_aにおけるカットの余裕 $m_a(X)$ は

$$\begin{aligned} m_a(X) &= c_a(X) - d_a(X) \\ &= c_a(X) \quad (P_a = \phi \text{ だから}) \\ &= m_0(X) \end{aligned}$$

であり，

$$m_a(X) \geq 0$$

がすべての $X \subset V$ に対して成立することがフローが存在するための必要十分条件となる.

　さて，G_aの双対グラフ $G_a^* = (V_a^*, E_a)$ を考える. G_aの容量関数をG_a^*の長さ関数とみなす. $G|X$ 及び $G|(V-X)$ がともに連

結であるとき，$E_a(X)$ は G_a^* の閉路に対応することは明らかである．$E_a(X)$ に対応する G_a^* の閉路の長さは $c_a(X)$ に等しい．したがって，G_a^* におけるすべての閉路の長さが非負であることが，フローが存在するための必要十分条件であると言い換えることができる．したがって許容性の判定は，負の長さの閉路が G_a^* に存在するかどうかの判定（存在すれば，フローは実現できない）を行えばよい．この判定は多項式時間で行える.[7]

3.2 アルゴリズム

ここでは，ソース・シンク対間に枝を付加しても平面であるような k-ネットワークにおいて多種フローを求める多項式時間アルゴリズムを与える．入力として与えるネットワークはすでに枝を付加したネットワークである．G_0 において s_i と t_i を含む面（複数個あればそのうちの1つ）は枝 e_{ai} を付加することにより2つに分割される．G_a におけるその分割された2つの面の境界上の s_i から t_i までの e_{ai} を含まない道をそれぞれ Q_i^1, Q_i^2 とする．Q_i^j $(j=1, 2)$ の方向を，s_i から t_i への方向と定める．また G_a の各枝 e には任意に向きを与えておき，e を流れるフロー $f_i(e)$ は e と同じ向きのとき正，逆向きのとき負とする．

```
procedure MULTIFLOW I;
begin
  if 双対グラフGa* に長さが負の閉路が存在する
    then 要求フローは実現できないので STOP;
  for すべての i 及び 枝 e∈E do fi(e):=0;
  while Ga にソース・シンク対が存在する do
    begin
      Δ>0 なる (si, ti) 及び Qi^j を選ぶ (1≤i≤k,
      1≤j≤z);
      comment
        Δ = min{1/2 min{ma(e1, e2)|e1, e2 ∈ Qi^j},
                min{ca(e)|e∈Qi^j}, di};
      for 各e∈Qi^j do
        begin
          fi(e):= { fi(e)+Δ : e と Qi^j の方向が一致するとき;
                  { fi(e)-Δ : それ以外のとき;
          ca(e):= ca(e)-Δ;
          if ca(e)=0 then e を除去する
        end;
      di:= di-Δ;
      ca(eai):= -di;
```

if $d_i=0$ then (s_i, t_i) と e_{ai} を除去する

end

end.

なお，$m_a(e_1, e_2)$ $(e_1, e_2 \in Q_i^j)$ は

$$m_a(e_1, e_2) = \min\{m_a(X) \mid X \subset V, E_a(X) \cap Q_i^j = \{e_1, e_2\}\}$$

と定義される．この値は双対グラフ G_a^* における e_1 と e_2 を含む，最も短い閉路の長さである．この値は多項式時間で計算できる.(2)

3.3 アルゴリズムの正当性及び計算時間

上記のアルゴリズムが正しく動作するためには，枝 e $(e \in Q_i^j)$ の容量を減らした，あるいは e を除去した直後のネットワーク N_a' においてカット条件が満足されなければならない．まずこのことを示す．もとのネットワーク $N_a=(G_a, \phi, c_a)$ において枝 $e \in Q_i^j$ が除去されなかった，すなわち容量 $c_a(e)$ (すべての $e \in Q_i^j$) が減ったネットワーク N_a' で $E_a' = E_a$ と仮定する（そうでない場合も同様に示せる）．N_a' における容量，要求量，余裕をそれぞれ c'，d'，m' と書くことにする．N_a においては $m_a(X) \geq 0$ が成立している．すべての $e \in Q_i^j$ に対して $c_a(e) := c_a(e) - \Delta$，また $d_i := d_i - \Delta$，$c_a(e_{ai}) := -d_i$

としたとき，

$$c'(X) = \begin{cases} c_a(X) - 2\Delta & : |E_a(X) \cap Q_i^j| = 2 \text{ のとき} \\ c_a(X) & : \text{それ以外のとき} \end{cases}$$

である．$m'(X) = c'(X)$ であるから，カットの余裕 $m'(X)$ が $m_a(X)$ より変更を受けるのは $|E_a(X) \cap Q_i^j| = 2$ なる X のみであり，しかも $m'(X) = m_a(X) - 2\Delta$ である．Δ の選び方より $m'(X) \geq 0$ である．

N_a がカット条件を満足する限り，$\Delta > 0$ となるような道 Q_i^j が存在することを示したのが補題3である．

［補題3］　N_a はカット条件 $m_a(X) = c_a(X) \geq 0$ を満足する平面ネットワークとする．このとき $e_1, e_2 \in Q_i^j$ に対して $m_a(e_1, e_2) > 0$ であるようなソース・シンク対 (s_i, t_i) 及び道 Q_i^j が存在する．

（証明）　どの (s_i, t_i) に対しても $m_a(e_1, e_2) = 0$ となるような枝 $e_1, e_2 \in Q_i^j$ が存在すると仮定する．すなわち Q_i^j ($j = 1$ 及び 2) に沿っては (s_i, t_i) 間のフロー f_i を流せないと仮定する．いま，付加した枝 e_{al} ($l = 1, \cdots, k$) を含まないような Q_i^j を選ぼう．N_a はカット条件を満足しているので，定理1により要求フローを実現する多種フローが存在する．(s_i, t_i) 間のフロー f_i が流れる枝だけからなる s_i から t_i までの道 R_i のうち，閉路 $\{e_{ai}\} \cup R_i$ の内部の面数が最小になるよ

うな道をR_iと定める(図1). Q_i^dとR_iによって囲まれるe_{ai}を含まない側の領域(いくつかの面の集合，図1の斜線部)が必ず存在する. ここで，あらゆるフローの流れ方のうちで，このようなQ_i^dとR_iによって囲まれる領域の面の数がいちばん少なくなるような流れ方とソース・シンク対(s_i, t_i)及びQ_i^dを選ぶ. このように選んだ領域はいくつかの部分(各部分は隣接する面の集合で，異なる部分に含まれる面は隣接していない)に分かれることがあり，それらをS_1, $\cdots$, S_hとする(図1ではS_1, S_2). S_b($1 \leq b \leq h$)の内部(R_i上を除く)にソース(またはシンク)が含まれているとき，それに対応するシンク(またはソース)は必ずS_bの中に含まれることがG_aの平面性からわかる. またどのS_bの中にも少なくとも1個はソース・シンク対が存在することが以下のように示せる. S_bの境界でかつQ_i^dにも含まれる枝にはf_i以外のフロー$f_{i'}$が流れている. ソース・シンク対$(s_{i'}, t_{i'})$がS_bに含まれていないと仮定する. フローf_iの経路(R_i)と$f_{i'}$の経路を交換して得られる多種フローに対して新しくR_iを定めると，R_iとQ_i^dの囲む領域の面の数がより小さくなり，矛盾する(図2).

ここで，任意のS_bの中のソース・シンク対(s_ℓ, t_ℓ)を考える. フローf_ℓの流れる経路(R_ℓ)がR_iと交叉している場合

には，R_l と R_i の枝の交換（すなわちフロー f_i と f_l の入れ換え）を行うことにより，f_l の経路 R_l が S_b の内部におさまるようにできる（図3）．R_l が R_i と交叉していなければ R_l は S_b の内部に含まれている．いずれにしろ R_l と $Q_l^{j'}$ に囲まれる e_{al} を含まない領域の面の数は明らかに S_b の面の数よりも小さくなっており，これは仮定に矛盾する．

（証明終）

以上より MULTIFLOW I は多種フローを正しく求めることがわかる．

次に計算時間について述べることにする．ソース・シンク対に順番を付けておき，その順に各 Q_i^j を調べる．$\Delta>0$ と定まるような Q_i^j は必ず存在することが補題3に示されているから，高々 $2k$ 個の Q_i^j について調べれば $\Delta>0$ となるものが見つかる．$\Delta=\frac{1}{2}ma(e_1,e_2)>0$ の場合，Q_i^j にフローを流す操作をしたのち，同じ Q_i^j にフロー f_i が流れることはない．したがって補題3により，枝が除去される（$\Delta=c(e)$ の場合）ことなく，かつソース・シンク対が除去される（$\Delta=d_i$）こともなく，すべての Q_i^j について $\Delta=0$ または $\Delta=\frac{1}{2}ma(e_1,e_2)>0$ となることはあり得ない．したがって，$\Delta>0$ と定まる回数は高々 $2k\times|E|$ 回である．$ma(e_1,e_2)$ の計算は多項式時間でできるので，結局 MULTIFLOW I は多

項式時間で終了することがわかる.

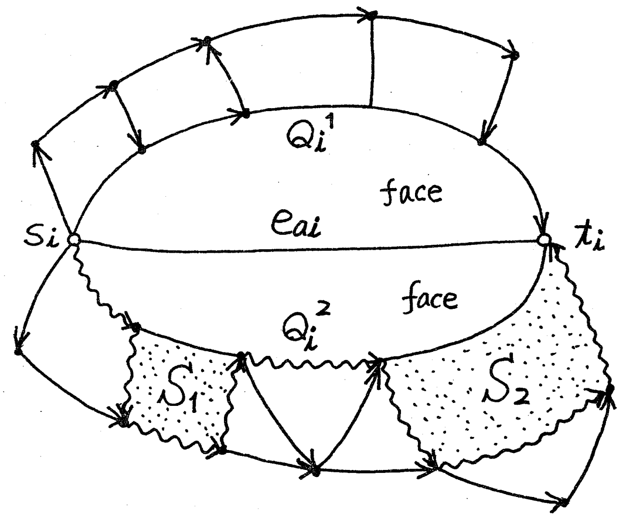

図 1.

矢印がフロー f_i の流れる枝と,フローの方向を表わす. 矢印のない枝にはフロー f_i は流れていない. 〰〰 は R_i を表わす.

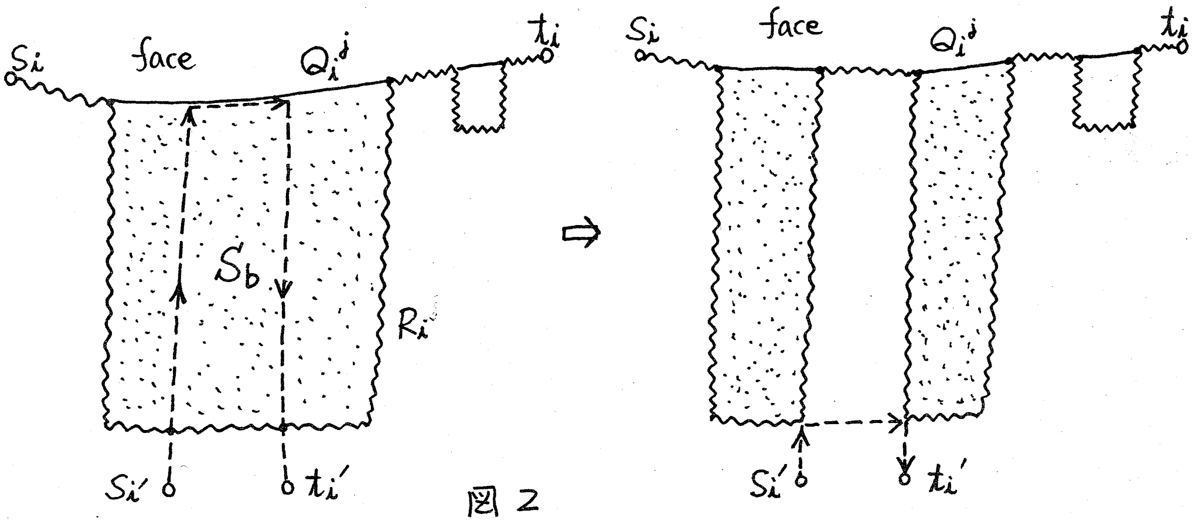

図 2

---- f_i' の経路

〰〰 R_i

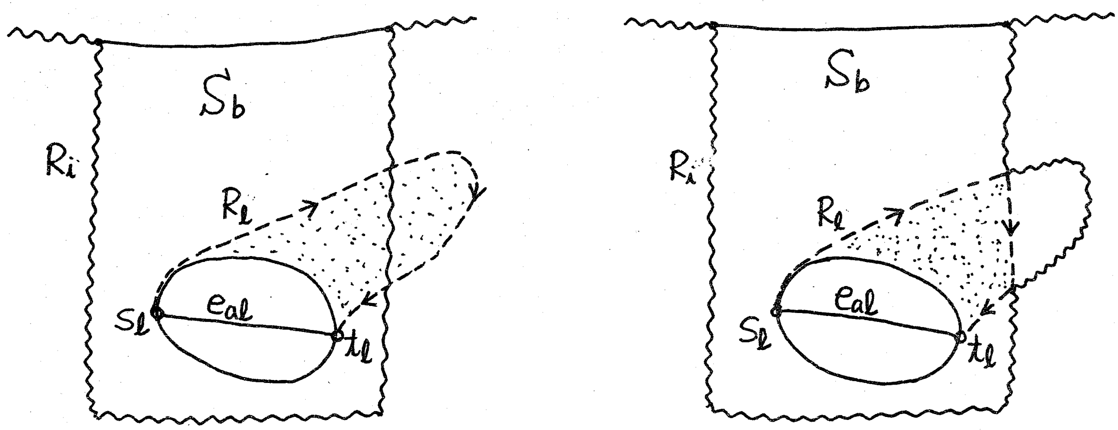

図3

〰〰 R_i

----- R_ℓ

4. 多種フローアルゴリズム(2)

本節では定理2で述べたネットワーク，すなわちソース$s_1, \cdots, s_\ell$及びシンク$t_1, \cdots, t_\ell$が外周B_0上に，ソース$s_{\ell+1}, \cdots, s_k$及びシンク$t_{\ell+1}, \cdots, t_k$が外周以外のある1つの面の境界B_1上にある場合について多種フローを求めるアルゴリズムを与える．

4.1 許容性の判定

ここではネットワークNに，与えられた要求量$d_1, \cdots d_k$を持つk種フローが存在するかどうか（すなわち許容性）を判

定する方法を述べる.

まず $X \subset V$ に対して, カット $E(X)$ の余裕 $m(X)$ を $m(X) = c(X) - d(X)$ と定義する. また, $e_1, e_2 \in B_i$ ($i=0$ または 1) に対して

$$m(e_1, e_2) = \min\{m(X) \mid X \subset V,\ E(X) \cap B_i = \{e_1, e_2\}\}$$

と定義する. ただし, $e_1 \neq e_2$ とは限らない. また $E(X) \cap B_i = \{e_1, e_2\}$ なる $X \subset V$ が存在しないときには $m(e_1, e_2) = \infty$ とする. (したがって e が橋枝でなければ $m(e, e) = \infty$ である) $m(e_1, e_2)$ は B_i の枝 e_1, e_2 を含むカットの余裕の最小値である.

定理1と補題1から次の補題を得る.

[補題4]　平面 k-ネットワーク $N=(G, P, c)$ において, 要求量 $d_1, \cdots, d_k$ を持つ k 種フローが存在するための必要十分条件は, すべての $e_1, e_2 \in B_i$ ($i=0$ 及び 1) に対して $m(e_1, e_2) \geq 0$ なることである.

(証明)　$G|X$ と $G|(V-X)$ のどちらも連結であるときには, $|E(X) \cap B_i|$ ($i=0, 1$) は 0, 1, 2 のいずれかである. ソース及びシンクはすべて B_0 あるいは B_1 上にあるので, $|E(X) \cap B_0| = |E(X) \cap B_1| = 0$ のときは明らかに $c(X) \geq d(X) = 0$ である. したがって補題1により, $|E(X) \cap B_i| = 1, 2$ なる X についてだけ $c(X) \geq d(X)$ が成立すれば, すなわちすべての $e_1, e_2 \in$

B_i $(i=0,1)$ に対して $m(e_1,e_2)\geq 0$ であれば，カット条件が成立する． （証明終）

上の補題から，許容性の判定には，すべての $e_1,e_2\in B_i$ $(i=0,1)$ について $m(e_1,e_2)\geq 0$ かどうかを調べればよい．これは以下のように多項式時間で行うことができる．

まず考えるべきカットは次の(i)～(iii)に場合分けできる．

(i) $E(X)\cap B_0\neq\phi$，かつ $E(X)\cap B_1=\phi$．

(ii) $E(X)\cap B_1\neq\phi$，かつ $E(X)\cap B_0=\phi$．

(iii) $E(X)\cap B_0\neq\phi$，かつ $E(X)\cap B_1\neq\phi$．

いま X，$Y\subset V$ が(i)の条件を満たし，かつ $E(X)\cap B_0=E(Y)\cap B_0$ のとき，$d(X)=d(Y)$ である．したがって $e_1,e_2\in B_0$ を固定したとき，$E(X)\cap B_0=\{e_1,e_2\}$ なる(i)を満たすどのような $X\subset V$ に対しても $d(X)$ は一定である．その値を $d_0(e_1,e_2)$ と書くことにする．同様に，$e_1,e_2\in B_1$ を固定したとき，$E(X)\cap B_1=\{e_1,e_2\}$ なる(ii)を満たすどのような X に対しても $d(X)$ は一定であり，その値を $d_1(e_1,e_2)$ と書くことにする．次に X，$Y\subset V$ がともに(iii)を満たし，$E(X)\cap B_i=E(Y)\cap B_i$ $(i=0$ 及び $1)$ のとき，$d(X)=d(Y)$ である．したがって $e_1,e_2\in B_0$，$e_3,e_4\in B_1$ を固定したとき，$E(X)\cap B_0=\{e_1,e_2\}$ かつ $E(X)\cap B_1=\{e_3,e_4\}$ であれば，そのようなどの X に対しても $d(X)$ は一定である．その値を $d_2(e_1,e_2;e_3,e_4)$

と書くことにする．このとき $e_1, e_2 \in C_0$ に対して余裕（m_0 と書くことにする）は

$$m_0(e_1, e_2) = \min\{c_0(e_1, e_2) - d_0(e_1, e_2),$$
$$\min\{c_2(e_1, e_2; e_3, e_4) - d_2(e_1, e_2; e_3, e_4) \mid e_3, e_4 \in B_1\}\},$$

$e_3, e_4 \in B_1$ に対しては余裕（m_1 と書くことにする）は

$$m_1(e_3, e_4) = \min\{c_1(e_3, e_4) - d_1(e_3, e_4),$$
$$\min\{c_2(e_1, e_2; e_3, e_4) - d_2(e_1, e_2; e_3, e_4) \mid e_1, e_2 \in B_0\}\}$$

となる．ただし

$$c_0(e_1, e_2) = \min\{c(X) \mid X \subset V, E(X) \cap B_0 = \{e_1, e_2\}, E(X) \cap B_1 = \phi\},$$
$$c_1(e_3, e_4) = \min\{c(X) \mid X \subset V, E(X) \cap B_1 = \{e_3, e_4\}, E(X) \cap B_0 = \phi\},$$
$$c_2(e_1, e_2; e_3, e_4) = \min\{c(X) \mid X \subset V, E(X) \cap B_0 = \{e_1, e_2\},$$
$$E(X) \cap B_1 = \{e_3, e_4\}\}$$

である．

さて，c_0，c_1，c_2 の計算法について述べることにする．まず $c_i(e_1, e_2)$ $(i=0, 1)$ の計算法について述べる．G の境界 B_j（$i=0$ のとき $j=1$，$i=1$ のとき $j=0$）上のすべての点を 1 点に縮約して自己ループを取り除いて得られるグラフを $G_j = (V_j, E_j)$ とする．G_j の双対グラフ $G_j^* = (V_j^*, E_j)$ を作り，G_j の容量関数を G_j^* の長さ関数とみなす．このとき，G_j の e_1, e_2 を含む最小カットの値，すなわち G における $c_i(e_1, e_2)$ の値は G_j^* の e_1 と e_2 を結ぶ自明でない最短道の長さに等しい．これ

はDijkstra法[1]を用いることによって多項式時間で求まる．したがってすべての$e_1, e_2 \in B_i$ $(i=0, 1)$に対して$c_i(e_1, e_2)$は多項式時間で求まる．次に$c_2(e_1, e_2; e_3, e_4)$の求め方を述べる．Gの双対グラフ$G^*=(V^*, E)$を作り，Gの容量関数をG^*の長さ関数とみなす．GのB_iに囲まれる面に対するG^*の点をv_i $(i=0, 1)$とする．また$e \in B_0$，$e' \in B_1$とする．G^*におけるv_0からv_1までのeとe'を含む単純な道の点集合（ただしv_0，v_1を除く）を$V(e, e')$と表わすことにし，その道の長さを$\ell(e, e')$と表わすことにする．このとき

$$c_2(e_1, e_2; e_3, e_4) = \min\{L(e_1, e_3; e_2, e_4),\ L(e_1, e_4; e_2, e_3)\}$$

である．ただし

$$L(e_1, e_3; e_2, e_4) = \min\{\ell(e_1, e_3) + \ell(e_2, e_4) \mid e_1, e_2 \in B_0, e_3, e_4 \in B_1, V(e_1, e_3) \cap V(e_2, e_4) = \phi\}$$

である．Lは多項式時間で求めることができる[6]．したがって，すべての$e_1, e_2 \in B_0$，$e_3, e_4 \in B_1$に対して$c_2(e_1, e_2; e_3, e_4)$も多項式時間で求めることができる．

d_0, d_1, d_2も簡単に多項式時間で求まるから，許容性の判定は多項式時間で求まる．

4.2　アルゴリズム

ここではℓ個のソース・シンク対がB_0上に，残りがB_1上に

ある平面k-ネットワークNにおいて多種フローを求める多項式時間アルゴリズムを与える．なお，Gの枝には適当に向きを与えておき，各枝eを流れるフロー$f_i(e)$はeの向きとフローの流れの向きが一致するときには正，一致しないときは負の値を持つものとする．

<u>procedure</u> MULTIFLOW II ;

<u>begin</u>

  <u>if</u> $m(e_1, e_2) < 0$ ($e_1, e_2 \in B_0$ または $e_1, e_2 \in B_1$) なるe_1, e_2が存在する <u>then</u> 要求フローは実現できないので STOP ;

  <u>for</u> すべての i $(1 \leq i \leq k)$ 及び $e \in E$ <u>do</u> $f_i(e) := 0$;

  <u>for</u> $\ell = 0$ 及び 1 <u>do</u>

    <u>begin</u>

      <u>for</u> 各 $e \in B_\ell$ <u>do</u> DECREASE (e) ;

      <u>while</u> B_ℓ 上にソース・シンク対が存在する <u>do</u>

        <u>begin</u>

          B_ℓ 上の (s_i, t_i) で $\Delta > 0$ なるものを選ぶ ;

          <u>comment</u> e_0 を s_i または t_i に接続する B_ℓ 上の枝 (一般性を失うことなく s_i に接続しているものとする) とし, Q を s_i から t_i までの e_0 を含む B_ℓ 上の道としたとき,

          $\Delta = \min\{\frac{1}{2}\min\{m(e_0, e) | e \in Q\},\ d_i,\ c(e_0)\}$;

```
        f_i(e0) := { f_i(e0)+Δ : e0がs_iから出る向きのとき；
                   { f_i(e0)−Δ : それ以外のとき；
        新しいソース・シンク対(s_i', t_i')を作り，s_i'をe0のs_iでない
        端点に，t_i'を点t_iに指定する；
        d_i' := Δ；
        d_i := d_i − Δ；
        if d_i = 0 then (s_i, t_i)を除去する；
        if s_i' = t_i' then (s_i', t_i')を除去する；
        c(e0) := c(e0) − Δ；
        if c(e0) = 0 then
          begin
            e0を除去する；
            for 新しくB_lに加えられた各枝e do DECREASE(e)
          end
      end
    end
end.

procedure DECREASE(e0);
begin
  comment e0 ∈ B_l;
```

```
mm := min{m(e₀, e) | e ∈ Bℓ};
if mm ≥ c(e₀) then
  begin
    e₀を除去する;
    for 新しくBℓに加えられた各枝e do DECREASE(e)
  end
else c(e₀) := c(e₀) − mm
end.
```

4.3 アルゴリズムの正当性および計算時間

上記のアルゴリズムが正しく動作するためには，枝e_0の容量を減らした，あるいはe_0を除去した直後のネットワークN'においてカット条件が満足されなければならない．まずこのことを示す．Nにおいてe_0が除去されなかった，すなわち容量$c(e_0)$が減ったネットワークN'で$E'=E$と仮定する（そうでない場合も同様に示せる）．N'における容量，要求量，余裕をそれぞれc', d', m'と書くことにする．Nにおいては$m(X) \geq 0$が成立している．procedure MULTIFLOW II において$c(e_0) := c(e_0) - \Delta$，$d_i' := 0$，$d_i := d_i - \Delta$としたとき，$N'$において

$$c'(X)=\begin{cases} c(X) & : e_0 \notin E(X)\cap B_\ell \text{ のとき,} \\ c(X)-\Delta & : \text{それ以外のとき,} \end{cases}$$

$$d'(X)=\begin{cases} d(X) & : e_0 \notin E(X)\cap B_\ell \text{ のとき,} \\ d(X)+\Delta & : E(X)\cap B_\ell \text{ が } e_0 \text{ と } Q \text{ の枝からなるとき,} \\ d(X)-\Delta & : \text{それ以外のとき,} \end{cases}$$

である．よってカットの余裕 $m'(X)=c'(X)-d'(X)$ が $m(X)$ より変更を受けるのは $E(X)\cap B_\ell=\{e_0, e\}$ となるような枝 $e\in Q$ が存在するときのみであり，しかも $m'(X)=m(X)-2\Delta$ である．Δ の選び方より $m'(X)\geq 0$ である．procedure DECREASE (e_0) についても同様に $m'(X)\geq 0$ がいえる．

N がカット条件を満足する限り，B_0 及び B_1 上に，$\Delta>0$ と定まる枝 e_0 が必ず存在することを示したのが補題 7 であるが，その証明のために次の補題 5，6 が必要である．ただし補題 5，6 の証明は文献(3)に与えられているので省略する．

［補題 5］(3)　各 $X\subset V$ に対して $c(X)\geq d(X)$ が成立しているとする．もし $X_1, X_2\subset V$ が $m(X_1)=m(X_2)=0$ であり，$D(X_1-X_2\,;\,X_2-X_1)=\phi$ であれば，$m(X_1\cap X_2)=m(X_1\cup X_2)=0$ が成立し，また $E(X_1-X_2\,;\,X_2-X_1)=\phi$ が成立する．

［補題 6］(3)　各 $X\subset V$ に対して $c(X)\geq d(X)$ が成立しているとする．$X_1, X_2\subset V$ は $m(X_1)=m(X_2)=0$ を満足するとし，Y は $G-(X_1\cup X_2)$ のある連結成分の点の部分集合であるとす

る．$D(X_1 \cap X_2 ; Y)=\phi$ ならば $D(Y ; V-(X_1 \cup X_2 \cup Y))=\phi$ である．

［補題7］ $N=(G, P, c)$ はカット条件を満足する連結な平面ネットワークとする．$B_l\ (l=0, 1)$ 上にソース・シンク対があるならば，次の条件を満たすような，あるソース s_i（または t_i）に接続する枝 $e_0 \in B_l$ が存在する：

すべての $e \in Q$ に対して $m(e_0, e_1)>0$ が成立する．ここで，$e_0=(s_i, v)$ としたとき，Q は v から t_i まで行く B_l 上の道である．

（証明） $l=0$ として一般性を失わない．補題が成立しないと仮定する．$V(B_0)-V(B_1)$ 上に（$V(B_i)$ は B_i 上の点集合を表わす）ソースあるいはシンクがあればその任意の1つを，なければ $V(B_0)$ 上の任意の1つを選ぶ．それはソースであるとして一般性を失わない．それを s_h としよう．s_h に接続する枝 $e_1, e_2 \in B_0$ に対して次の条件(i), (ii), (iii), (iv)を満たす $X_i \subset V$（$i=1$ 及び2）が存在する：

(i) $e_i \in E(X_i)$

(ii) $t_h \notin X_i$.

(iii) $m(X_i)=0$.

(iv) $G|X_i$ 及び $G|(V-X_i)$ がともに連結である．

$X_1 \cap V(B_1)=\phi$ あるいは $X_2 \cap V(B_1)=\phi$ であることを示そう．

$i=1$ 及び 2 に対して $X_i \cap \nabla(B_1) \neq \phi$ と仮定する。このとき $X_1 \cup X_2$ は s_h と t_h を分離する．すなわち s_h と t_h は $G-(X_1 \cup X_2)$ の異なる連結成分に含まれる．$t_h \notin X_1$ かつ $t_h \notin X_2$ であるので $X_1 \cap X_2 \cap \nabla(B_0) = \phi$ である．まず $D(X_1 \cap X_2 ; \{s_h\}) = \phi$ が成立していることを示そう．$X_1 \cap X_2 \cap \nabla(B_1) = \phi$ のときは明らかに成立している．よって $X_1 \cap X_2 \cap \nabla(B_1) \neq \phi$ とする．このときもし $s_h \in \nabla(B_0) \cap \nabla(B_1)$ ならば s_h, t_h の選び方からして $t_h \in \nabla(B_0) \cap \nabla(B_1)$ であり，$X_1 \cap X_2 = \phi$ になってしまうから，$s_h \in \nabla(B_0) - \nabla(B_1)$ である．このとき明らかに $D(X_1 \cap X_2 ; \{s_h\}) = \phi$ である．このようにして $D(X_1 \cap X_2 ; \{s_h\}) = \phi$ が成立していることが示せた．したがって補題 6 より $D(\{s_h\} ; \nabla - (X_1 \cup X_2 \cup \{s_h\}) = \phi$ となるが，これは矛盾である．以上により $i=1$ または 2 に対して $X_i \cap \nabla(B_1) = \phi$ であることが証明できた．

さて，$X_i \cap \nabla(B_1) = \phi$ であり，$|X_i \cap \nabla(B_0)|$ が最小となるようなソース s_h と X_i を選ぶ．ただし条件 (i)～(iv) を満たすとする．ここで $i=1$ としても一般性を失わない．このとき次のようなソース・シンク対 (x, y) （x と y のどちらか一方がソースで他方がシンク）が存在する：

$$x \in \nabla(B_0) \cap X_1,$$

$$y \in \nabla(B_0) - X_1.$$

いま w を $E(X_1) \cap B_0$ の端点で $w \notin X_1$ かつ $w \neq s_h$ なる点とす

る．上述の対(x, y)として，wからB_0上をt_h，s_hの順に出会う方向に回ったとき最初にyに出会うような対を選ぶ．命題が成立しないと仮定しているので，次のような点集合X_3が存在する：

$m(X_3)=0$，$G|X_3$および$G|(V-X_3)$が連結，$x, y \notin X_3$，$e \in E(X_3)$（ここで$e=(x, z)$はxに接続するB_0上の枝とする）．

X_3を上の条件を満たす最小の点集合とする．(x, y)の選び方と$X_1 \cap V(B_1)=\phi$より$D(X_1-X_3; X_3-X_1)=\phi$．したがって補題5より$E(X_1-X_3; X_3-X_1)=\phi$，かつ$m(X_1 \cap X_3)=0$である．また$G|(X_1 \cap X_3)$および$G|(V-(X_1 \cap X_3))$は連結である．なぜなら，$G|(X_1 \cap X_3)$が非連結ならばzを含む連結成分以外の各連結成分はB_0上の点を含まず，もちろんB_1上の点も含まない．すなわちソースやシンクを含まない．これらの各成分の点集合を$U_i\ (i \geq 1)$で表わせば$c(U_i)>0$，$d(U_i)=0$である．点zを含む成分の点集合をU_0と表わすことにすると

$$
\begin{aligned}
0 = m(X_1 \cap X_3) &= c(X_1 \cap X_3) - d(X_1 \cap X_3) \\
&= \{c(U_0) - d(U_0)\} + \sum_{i \geq 1} c(U_i) > 0
\end{aligned}
$$

となり矛盾である．したがって$G|(X_1 \cap X_3)$は連結である．このことから$G|(V-(X_1 \cap X_3))$が連結であることもわかる．

$X_1 \cap X_3 \subset X_1 - \{x\}$であるから，$X_3 \not\subset X_1$ならば$X_3$の最小性に

反し，$X_3 \subset X_1$ ならば X_1 の最小性に反する．　　　　（証明終）

以上より MULTIFLOW II は多種フローを正しく求めることがわかる．

次に計算時間について述べることにする．まずソース・シンク対の個数であるが，これは分裂を繰り返すことにより多量に増えるおそれがある．しかし，(s_i, t_i) と (s_j, t_j) が重なったとき，これらを1つのソース・シンク対とみなせば，アルゴリズムのどの段階においてもソース・シンク対の総数は高々 $O(n^2)$ 個である．(s_i, t_i) と e_0 を選んだとき Δ は多項式時間で求めることができる．ソース・シンク対を選ぶ順番を適当に定めておけば，多項式時間で $\Delta > 0$ なる e_0 を見つけることができる．次に MULTIFLOW II の while 文の中で $\Delta > 0$ となることが何回起こるか，すなわち"フローを流す操作"の繰り返しの回数を調べる．B_ℓ 上の各枝に対して procedure DECREASE(e) を行っているので，各 e に対して $m(e, e') = 0$ となる枝 $e' \in B_\ell$ が存在する．したがって (s_i, t_i) 間のフローを e_0 に流す操作をしたのち，新しい対 (s_i', t_i') 間のフローが e_0 に流れる（すなわち e_0 を逆もどりする）ことはない．したがって枝 e_0 について $\Delta > 0$ となる回数は高々 $O(n^2)$ 回である．したがってアルゴリズム全体で $\Delta > 0$ となる回数は高々 $O(n^3)$ 回となる．したがって MULTIFLOW II は多項

式時間で終了することが示された．

5. むすび

本文では，2種類の平面グラフに対して，要求フローを実現する多種フローを求める多項式時間アルゴリズムを与えた．しかし，計算時間は効率が良いとはいえず，改善の余地があり，検討中である．

文　献

(1) A.V. Aho, J.E. Hopcroft and J.D. Ullman : The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass. (1974).

(2) E.L. Lawler : Combinatorial Optimization : Networks and Matroids, Holt, Rinehart and Winston (1976).

(3) H. Okamura : Edge-disjoint paths in planar graphs, Discrete Applied Math., to appear.

(4) H. Okamura and P.D. Seymour : Multicommodity flows in planar graphs, Journal of Combinatorial Theory, Series B 31, pp. 75-81 (1981).

(5) P.D. Seymour : On odd cuts and planar multicommodity

flows, Proc. London Math. Soc. (3) 42, pp.178-192 (1981).

(6) J.W. Suurballe : Disjoint paths in a network, Networks 4, pp. 125-145 (1974)

(7) R.L. Tobin : Minimal complete matchings and negative cycles, Networks 5, pp. 371-387 (1975).

(8) 松本，西関，斎藤：平面グラフの多種フローを求める多項式時間アルゴリズム，信学論(A) 採録決定.