1995年度日本オペレーションズ・リサーチ学会
春季研究発表会

外平面グラフ上の最大流量を求める並列アルゴリズム

02401223　豊橋技術科学大学　　*中山慎一　　NAKAYAMA Shin-ichi
01603863　豊橋技術科学大学　　増山 繁　　MASUYAMA Shigeru

1　まえがき

ネットワーク $N=(G=(V,E),s,t,c)$ を次のように定義する．G は，多重辺を持たない連結有向グラフであり，節点数 $|V|$ を n，辺数 $|E|$ を m で示す．s と t $(s\neq t)$ は V に属する特別な節点で，それぞれ，始点，終点と呼ばれる．各辺 $(i,j)\in E$ には，その上を流れる流量の上限値を表す容量 $c(i,j)\geq 0$ が与えられているとする．最大流量問題 [3] とは，ネットワーク $N=(G,s,t,c)$ においてそれぞれの辺の流量関数 $f(u,v)$ が，容量制約: $(0\leq f(u,v)\leq c(u,v))$，流量保存則: $(v\neq s,\ t$ なる各 $v\in V$ について $\sum_{u:(u,v)\in E}f(u,v)=\sum_{u:(v,u)\in E}f(v,u)$ が成り立つ)，を共に満たすという条件のもとで，始点 s から終点 t への流量値 $\sum_{u:(s,u)\in E}f(s,u)-\sum_{u:(u,s)\in E}f(u,s)$ を最大にするものを求める問題であり，今までに数々の研究がなされてきた [3].この問題を並列アルゴリズムの観点からながめると，一般のグラフに関しては，最大流量問題は P 完全 [1] であり効率の良い並列アルゴリズムが存在しないと考えられている [1].

本論文では，取り扱うグラフ G を外平面グラフ [2]，つまり，グラフの全ての節点を無限面 (グラフを平面に埋め込んだ時，グラフの外側の無限に広がった領域) の境界上に乗るように平面埋め込み可能なグラフ，に限定して，外平面グラフ G 上の任意に指定した2節点 s，t，$(s\neq t)$ 間の最大流量問題を CRCW PRAM[1] 上で，$O(n^2/\log n)$ 個のプロセッサを用いて $O(\log n)$ 時間で解く並列アルゴリズムを提案する．

2　準備

本論文で取り扱う外平面グラフ G は無向グラフである．無向グラフの場合には，無向辺 (u,v) を，それと同一の容量をもつ互いに逆方向の2つの有向辺 (u,v)，(v,u) に置き換えることによって有向グラフの場合と同様に最大流量問題が定義できる．この場合に，無向グラフにおける任意の辺 (u,v) の流量 f' は，有向グラフの流量関数 $f(u,v)$ を用いて次のように計算できる．

$f'(u,v)=\max\{0,f(u,v)-f(v,u)\}$,
$f'(v,u)=\max\{0,f(v,u)-f(u,v)\}$

以下で提案するアルゴリズムは，必ずしも，実際に無向グラフのすべての辺を2本の互いに逆方向の有向辺で置き換えるわけではないが，求める最大流量は上記の意味においてのものである．

G より $v\in V$ を取り除くと非連結になるようなカット節点 v が存在しない，すなわち，2連結な外平面グラフ G は，全ての節点を1度だけ通る閉路 H を無限面との境界として埋め込み，閉路 H を構成しない残りの辺を互いに交差しないように閉路 H の内側に埋め込むような平面埋め込みを持つ [1]．無限面との境界をなす辺のことを周辺 (side edge) と呼び，他の辺を対角辺 (diagonal) と呼ぶ．また，始点 s を節点番号1として，周辺に沿って節点番号 $1,\cdots,n$ を付けた場合，周辺全体から成る閉路 $H:s(=1),\cdots,i,i+1,\cdots,t$(終点)，$\cdots,j,j+1,\cdots,n$ において，路 $s(=1),\cdots,i,i+1,\cdots,t$ 上の辺を前周辺 (forward-side edge) と呼び，路 $t,\cdots,j,j+1,\cdots,s$ 上の辺を後周辺 (rear-side edge) と呼ぶ．但し，周辺 (s,t) が存在する場合，辺 (s,t) を前周辺とする．

3　最大流量を求める並列アルゴリズム

ここでは，入力として与えられる外平面グラフ G は2連結グラフであると仮定する．2連結とは限らない一般の外平面グラフの場合についても，2連結成分を並列に求め [3]，各2連結成分に対し本並列アルゴリズムを適用することにより容易に拡張できる．以下に，最大流量を求める並列アルゴリズム Procedure Maximum-Flow の概略を述べる．

(Step 1) 与えられた外平面グラフ G の周辺を求め [4]，始点 s を節点番号1として，周辺 (ハミルトン閉路) に沿って節点番号 $1,\cdots,n$ を付ける [4].

もし，対角辺が存在しなければ，グラフ G はハミルトン閉路 C のみから成るので Step 6 へ分岐する．

(Step 2) G の面 $C_1,C_2,\cdots,C_l$, $l\geq 1$ を求める [1].

(Step 3) 以下のように根付木 $T_G=(V_T,E_T,r)$, V_T は節点集合，E_T は辺集合，r は根，を構成する [1]. $V_T=\{\ c_i\mid c_i$ は面 C_i に対応 $\}$，辺 $(n,1)$(節点番号1は始点 s である) を含む面 C_1 に対応する節点 c_1 を木 T_G の根 r，$E_T=\{\ (c_i,c_j)\mid c_i,\ c_j,\in V_T$ それぞれに対応する面 C_i，C_j が共に辺 (i,j) をもつ $\}$ とする．

終点 t の節点番号より1小さい節点番号をもつ節点 v_{t-1} と t の周辺 (v_{t-1},t) を含む面 C_l に対応する T_G の節点を c_l とし，T_G における根 c_1 から c_l への路 p に着目する．この路 $p=c_1,c_2,\cdots,c_l$

上の辺を T_G の主線, 路 p 上の端点も含む節点を主節点と呼ぶ.

(Step 4) T_G の主線を求める. もし, 主線以外の辺が存在しなければ, Step 6 へ分岐する.

(Step 5) T_G より主節点, および, それに接続する辺を除去すると, 部分木 $T_1, \cdots, T_k$, $k \geq 1$ からなる森を得る. 各部分木 T_i において, 主節点と接続していた節点をそれぞれ木 T_i の根とする. 各 T_i, $i = 1, \cdots, k$, において, T_i の節点 $c_1^i, c_2^i, \cdots, c_l^i$ に対応する G の面 $C_1^i, C_2^i, \cdots, C_l^i$ によって生成される G の部分グラフ $G_i = (V_i, E_i)$, $V_i = \{ v \in V \mid v$ は $C_1^i, C_2^i, \cdots, C_l^i$ の少なくとも 1 つに含まれる $\}$, $E_i = \{ e \in E \mid e$ は $C_1^i, C_2^i, \cdots, C_l^i$ の少なくとも 1 つに含まれる $\}$, を構成する.

また, G_i の構成と同様に主節点 $c_1, c_2, \cdots, c_{l'}$ に対応する G の面 $C_1, C_2, \cdots, C_{l'}$ によって生成される G の部分グラフ $\tilde{G}$ を構成する.

for *all* i, $1 \leq i \leq k$ in parallel do

T_i に対応する G の部分グラフ G_i と $\tilde{G}$ との共有節点 x_i-y_i $(x_i \neq y_i)$ 間の最大流量を求め, $\tilde{G}$ の各辺 (x_i, y_i) の容量 $c(x_i, y_i)$ をそれぞれ $f(x_i, y_i)$ とする.

(Step 6) $\tilde{G}$ の始点 s, 終点 t 間の最大流量を求める. この値が G の最大流量になる. $\tilde{G}$ の最小カット [3] を求めるために, $\tilde{G}$ の修正双対グラフを構成して $\tilde{G}^*$ とする. 但し, 修正双対グラフは, $\tilde{G}$ の双対グラフ $\tilde{G}^d$[2] を求め, $\tilde{G}$ の無限面に対応する $\tilde{G}^d$ の節点 v^* を除去し, 代わりに v_{up}^*, v_{down}^* の 2 つの節点を付加し, 前周辺に対応する $\tilde{G}^d$ の辺を v_{up}^* に, 後周辺に対応する $\tilde{G}^d$ の辺を v_{down}^* にそれぞれ接続させて得られるグラフと定義する.

(Step 7) $\tilde{G}^*$ の各辺 (i^*, j^*) の長さ $l(i^*, j^*)$ を, 対応する $\tilde{G}$ の辺 (i, j) の容量 $c(i, j)$ とし, $\tilde{G}^*$ における v_{up}^* から v_{down}^* への最短距離を求める. この最短距離が G の最小カット [3] の値であり, 最大流量-最小カットの定理 [3] により, G の最大流量 val が求まる.

4 辺の流量を求める並列アルゴリズム

G の各辺の流量を求める並列アルゴリズムでは, まず, G の最大流量 val を基に $\tilde{G}$ の各辺の流量 $f(x, y)$ を決定し, 次に, $\tilde{G}$ における $\tilde{G}$ と各部分グラフ G_i の共有辺 (x_i, y_i) の流量 $f(x_i, y_i)$ を基に, 各 G_i の各辺の流量を並列に求めることにより, G の各辺の流量を決定する.

以下に, 辺の容量を求める並列アルゴリズムの概略を示す (流れの向きの決定法については容易なので省略する).

Procedure Flow-Determination

begin

(Step 1) { $\tilde{G}$ の各辺の流量を決定する. }

$\tilde{G}$ の内面中の面に対応する $\tilde{G}^*$ の節点を $v_1^*, \cdots, v_l^*$ とする. $\tilde{G}^*$ の各辺 (i^*, j^*) の長さ $l(i^*, j^*)$ を, 対応する $\tilde{G}$ の辺 (i, j) の容量 $c(i, j)$ とし, v_{up}^* から各 v_i^*, $i = 1, \cdots, l$, への最短距離を求め, その値を $l(v_i^*)$, $i = 1, \cdots, l$, に格納する. ただし, v_{up}^*, v_{down}^* の値は, $l(v_{up}^*) = 0$, $l(v_{down}^*) = val$ とする.

$\tilde{G}$ の辺 (u, v) の流量 $f(u, v)$ は, 対応する (交わる) $\tilde{G}^*$ の辺を (u^*, v^*) とすると, $f(u, v) = |l(u^*) - l(v^*)|$ で求まる.

if 部分グラフ G_i が存在しない then 停止.

(Step 2)

for *all* i, $1 \leq i \leq k$ in parallel do

begin

$\tilde{G}$ と G_i の共有節点を x_i, y_i とし, Step 1 で求めた $\tilde{G}$ における辺 (x_i, y_i) の流量を $f(x_i, y_i)$ とする. T_i に節点 v_r^*, 及び, G_i の周辺の数 n' 個の節点 $v_{1'}^*, \cdots, v_{n'}^*$ を追加し, v_r^* と T_i の根, 及び, 面 $\tilde{C}_j$ に対応する T_i の節点 v_j と $\tilde{C}_j$ の周辺 $e_{k'}$ に対応する追加した節点 $v_{k'}$ を辺で接続した木 T_i^* を構成する. T_i^* の辺 (u^*, v^*) において, u^* を v^* の親とする. T_i^* の根から u^*, v^* への距離をそれぞれ $len(u^*)$, $len(v^*)$ とする. (u^*, v^*) に対応する G_i の辺 (u, v) の流量 $f(u, v)$ は次式で求まる.

$$f(u, v) = \begin{cases} len(v^*) - len(u^*): & len(v^*) \leq f(x_i, y_i) \text{ のとき} \\ f(x_i, y_i) - len(u^*): & len(v^*) > f(x_i, y_i) \text{ かつ} \\ & len(u^*) < f(x_i, y_i) \text{ のとき} \\ 0: & len(v^*) > f(x_i, y_i) \text{ かつ} \\ & len(u^*) \geq f(x_i, y_i) \text{ のとき} \end{cases}$$

end

end.

以上から次の定理が得られる.

[定理 1] Procedure Maximum-Flow と Procedure Flow-Determination は, それぞれ $O(n^2/\log n)$ 個のプロセッサを用いて $O(\log n)$ 時間で実行可能である. 但し, n は G の節点の個数である. □ *(証明略)*

参考文献

[1] A. Gibbons and W. Rytter: *Efficient Parallel Algorithms*, Cambridge University Press, 1988.

[2] F. Harary: *Graph Theory*, Addison-Wesley, 1969.

[3] J. van Leeuwen: *Graph Algorithms, in: J. van Leeuwen, eds. Handbook of Theoretical Computer Science*, Elsevier Science Publishers B.V., 1990.

[4] 中山慎一, 増山繁: "外平面グラフ上の st-最短経路を求める並列アルゴリズム", 1994 年度日本 OR 学会秋季研究発表会アブストラクト集, pp.240-241.