6S−6

最大流問題の近年の算法の実際的評価

吉田勝人　中森眞理雄
東京農工大学

1．はじめに

最大流問題を解く算法の研究は，最近の 10 年間では Goldberg と Tarjan のプッシュリラベル法（以下 P/R 法）とそれに対する多くの改良によって大きな発展をみている．10 年前までの算法の実際的な効率に対しては，[1] にくわしい．本稿はプッシュリラベル法の各種算法の計算機実験による比較である．[1] で良い結果のでている Dinic，Karzanov 法とあわせ，実現した算法を表 1 に示す．n，m はそれぞれ点の数，枝の数であり，U は最大容量である．

表 1．実現した算法

算法	計算量
1. Dinic (DNC)	$O(n^2m)$
2. Karzanov (KARZ)	$O(n^3)$
3. Sleator, Tarjan (ST)	$O(nm\log n)$
4. P/R FIFO (PRF)	$O(n^3)$
5. P/R 最遠点 (PRHD)	$O(n^2m^{1/2})$
6. P/R 最近点スケーリング (PRSSD)	$O(nm+n^2\log U)$
7. P/R 最遠点スケーリング (PRSHD)	$O(nm+n^2(\log U/\log\log U)$
8. P/R 動的木 (PRSLFT)	$O(nm\log(n^2/m))$

2．プッシュリラベル法

各点 v はそれぞれ超過量 e(v) と距離ラベル d(v) を持つ．距離ラベルを用いて e(v)>0 の超過点 v に次の二つの操作（P/R 操作）を繰り返し適用していくことによって算法は進行する．

プッシュ：e(v)>0 ($v\in V$)，d(v)=d(w)+1 ($(v,w)\in E$) のとき，v から w にフローを流す．

リラベル：e(v)>0 だが，プッシュがもうできないとき距離ラベルを変更する．

超過点がなくなったとき算法は終了しそのときの e(t) が最大流である．複数存在する超過点の中からどれを選択するか，どれだけの量を流すかなどによって多くの算法がある．

3．プッシュリラベル法の実現

3.1 実現方法

算法はすべて FORTRAN で作成した．

P/R 操作を行う各算法は超過量と距離ラベルの値を格納するために，それぞれ大きさ n の配列を用意する．また各算法ごとに P/R 操作の対象となる点（候補点）を格納する構造を用意する．ここで超過点と候補点を区別して使っている．例えば，PRSHD では超過点の中で e(v)>Δ/k の点だけが候補点となる．次に各算法の候補点の格納構造を述べる．

(1)PRF

先頭ポインタと後尾ポインタを持った大きさ n の配列による FIFO キューを使う．

(2)PRHD

同じ距離ラベルを持つ候補点をそれぞれリスト表現で格納する．これには，大きさ n の配列を二つ用意する．一つはその距離ラベルを持つ候補点へのポインタ用（なぜ大きさ 2n ではないのかは後述）であり，もう一つは候補点の格納用である．

(3)PRSSD

e(v)>Δ/k の点だけを候補点にする．格納構造は最遠点法と同様である．Δが変わったときに次の候補点を e(v) の配列を走査して求める．超過点をすべて候補点として維持しておくのは複雑な格納構造と操作を必要とするからである．

(4)PRSHD

PRHD と同様のものに加えて，大きさ n の配列を用いる．Δ/k だけフローを流すのを保証するために使うスタック用である．

(5)PRSLFT

六つの大きさ n の配列で動的木を表現し，候補点は PRHD と同様のものに格納する．

3.2 距離ラベルの計算

算法を効率よくするために最初の距離ラベルは，d(s)=n，d(t)=0 とし，残りの点は残余グラフ上で t からの幅優先探索によって計算する．つまり t からの最短路の値がその点の距離ラベルとなる．探索の途中，e(v)>0 (PRSHD なら e(v)>Δ/k) の点を見つけたら候補点として格納する．距離ラベルの計算が終了した後 P/R 操作により算法が進行するが，途中，d(v)>n となった点は候補点には入れない．候補点がなくなったら（つまりもう t に流せなくなったら），d(s)=0，d(t)=n とし，今度は s から幅優先探索によって距離ラベルを計算する．これにより最遠点選択では，距離ラベルのポインタ用の配列が大きさ n ですむ．しかし，これだけでは到底 DNC や KARZ との比較の対象となるだけの効率は達成できない．

そこで，積極的に算法の途中である時点になったら距

Practical Evaluation of Recent Maximum Flow Algorithms

Katsuhito YOSHIDA and Mario NAKAMORI

Department of Computer Science, Tokyo University of Agriculture and Technology

離ラベルを再計算することを試みる．このある時点を筆者はリラベルの回数（rc）で設定した．すなわち距離ラベルを計算するときに候補点となった点の個数（vc）に対し，rc>vc となった時点で距離ラベルの再計算を行う．これはあくまでも，経験的な予測による設定であり，理論的な裏付けがあるわけではない．図 1 に 2 種類のネットワークに対して vc の係数を変えたときの速度の変化を示す．測定に使用した計算機は NEC ACOS1000 である．プログラムは，最適化は行わず，ネットワークの入出力を除いた部分のミリ秒単位の計測であり，それぞれ 10 回の計算の平均である．容量は 1 から 100 までの整数である．また図の上段に示した 10 個の数値は，その二つのネットワークに対して，算法の開始時とすべての超過点が d(v)≧n となったときの 2 回だけ距離ラベルの計算を行ったときの実行時間（単位はミリ秒）である．

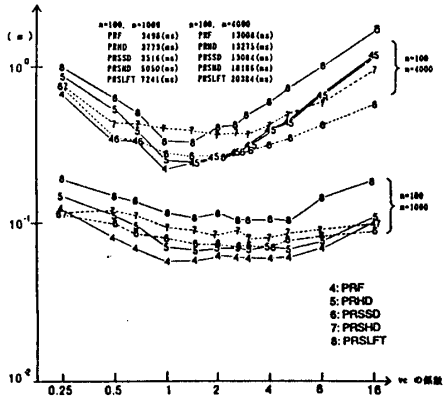

図 1．vc の係数による速度の変化

距離ラベルを何回か計算することにより大幅な効率の改善を得た．

密なネットワークでは距離ラベルの計算は約 3〜5 回，疎なネットワークでは約 4〜6 回行われている．この図は，ネットワーク中で右余曲折の P/R 操作を繰返しながらフローを移動することと，幅優先探索で適切な距離ラベルを求めることとのトレードオフの比較となっている．疎なネットワークで各算法が最速となる vc の係数に幅があるのは，適切な距離ラベルを求めても，その計算にかかる時間と相殺していることを示す．PRSSD, PRSHD は超過量をできるだけネットワーク中に散らそうという性質のものなので，ある程度 P/R 操作が行われたほうがよく，そのために係数に幅がある．PRSLFT は最遠点選択を採用しているので，PRHD と同じような動きをしている．動的木の操作に手間がかかっているために，効率はよくないということをこの図より見ることができる．

次節の計算結果は PRF, PRHD, PRSLFT は係数を 1，PRSSD, PRSHD は係数を 3 としたときのものである．

4．計算結果

図 2, 3 にランダムに作成したネットワークでの計算結果を示す．容量は 1 から 100 までの整数であり，それぞれ 10 回の計算の平均である．

P/R 法の中では PRF が最速である．しかし，KARZ 以上の性能は得られていない．距離ラベルをいつ計算するのかを判定する条件を変えることによって，また別の状況が生まれることも考えられる．

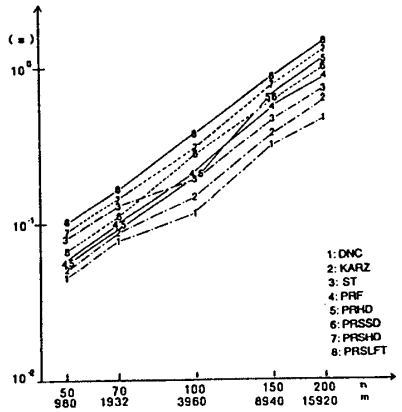

図 2．計算結果（m=2/5*n(n-1)）

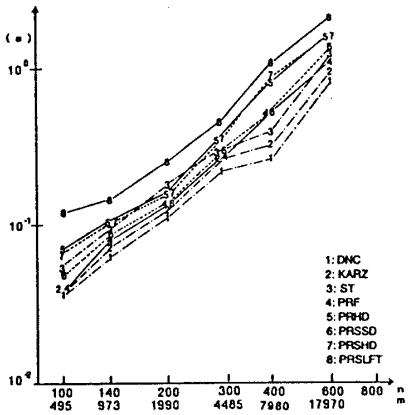

図 3．計算結果（m=1/20*n(n-1)）

5．おわりに

今日の計算の複雑さの理論の例にもれず，最大流問題においても理論的な手間が実際の計算時間を十分に反映しているわけではないことが明らかとなった．ただし，別の種類のネットワークについても，さらに調べてみる必要がある．

参考文献

[1] Imai, H. : On the Practical Efficiency of Various Maximum Flow Algorithms, J. Oper. Res. Soc. Japan, Vol. 26, No. 1, pp. 61-83, (1983).

[2] Goldberg, A. V. and Tarjan, R. E. : A New Approach to the Maximum-Flow Problem, J. ACM, Vol. 35, No. 4, pp. 921-940, (1988).

[3] Ahuja, R. K., Orlin, J. B. and Tarjan, R. E. : Improved Time Bounds for the Maximum Flow Problem, SIAM. J. Comput., Vol. 18, No. 5, pp. 939-954, (1989).

[4] Sleator, D. D. and Tarjan, R. E. : A Data Structure for Dynamic Trees, J. Comput. Syst. Sci., Vol. 26, pp. 362-391, (1983).