グラフ・ネットワーク最適化

浅野　孝夫

1. はじめに

数理計画の様々な問題がグラフ・ネットワークを用いて定式化され解かれている。グラフ・ネットワークを用いる利点は、問題が明確になって目で見て直観的に解を求めることがある程度できること、および情報科学の分野で展開されたアルゴリズムやデータ構造を用いて大きいサイズの問題も効率的に解ける点などであろう。たとえば、次のような問題などもグラフ・ネットワークを用いて定式化され解ける。

以下の表はある年のCリーグのペナントレースの8月某日終了時点の成績および残り試合の対戦数である。この時点で最終的に中日の優勝する可能性があるかどうかを判定せよ．可能性があるときはその可能性を実現するような，各チームと他のチームとの残り試合の対戦成績の例を記せ。

チーム	勝数	負数	残り試合数
ヤクルト	66	38	26
広島	62	40	28
巨人	56	47	27
横浜	52	51	27
阪神	36	66	28
中日	33	63	34

残り対戦数	ヤクルト	広島	巨人	横浜	阪神	中日
ヤクルト	-	6	5	5	2	8
広島	6	-	7	6	5	4
巨人	5	7	-	3	6	6
横浜	5	6	3	-	6	7
阪神	2	5	6	6	-	9
中日	8	4	6	7	9	-

なお、Cリーグのルールでは、引き分けは再試合をして勝敗を必ず決めるので、引き分けは起きないものとしてよい。また、最終的に勝数が最大のチームが複数チーム存在したときには、プレーオフで優勝チームを決定するので、プレーオフに出場できれば、優勝する可能性はある。

上記の問題は、最大フロー最小カット定理に基づいて解決できる。もちろん別の方法でも解けるが、ネットワークフローに基づく方法が最も自然であるので後述する。また次の問題もネットワーク問題として解ける。ここで、$T1,\cdots,T6$は教師で$C1,\cdots,C6$は科目であり、表の数字は各教師のそれぞれの科目の得意度(10点満点)を表している。各教師に科目を重複なく1科目担当してもらい、総得意度(担当科目の得意度の総和)を最大にするのがこの問題である。これは最適割り当て問題と呼ばれるネットワーク問題の有名な問題の例である。

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$
$T1$	6	6	7	8	6	8
$T2$	6	5	3	6	4	5
$T3$	9	8	6	9	7	9
$T4$	4	5	5	6	3	4
$T5$	5	3	2	6	4	3
$T6$	9	8	8	9	6	8

グラフ・ネットワーク最適化の研究は日本でも極めて活発に実行されてきた。伊理正夫先生や冨澤信明先生による最大フローアルゴリズムや最小費用フローアルゴリズムはFord-FulkersonやEdmonds-Karpのアルゴリズムとほぼ同時に発表され、この分野の研究の端緒となっている。また藤重悟先生の強多項式最小費用フローアルゴリズムも最初のTardosの強多項式アルゴリズムとともに有名である。また最近では、最小カットを求める茨木俊秀先生・永持仁先生の研究は極めて斬新なアイデアに基づいていて、画期的な研究として全世界的に注目されている。また、福田公明先生の逆探索法や平面グラフに対する西関隆夫先生の著

あさの　たかお　中央大学理工学部情報工学科
〒112 文京区春日1-13-27

書も世界的な注目を浴びている。これらの成果は、藤重悟先生・室田一雄先生編の「離散構造とアルゴリズム、I,II,III,IV」(近代科学社)で詳説されている。拙著「情報の構造、上,下」(日本評論社)でもいくつか扱っている。このように、グラフ・ネットワーク最適化の研究に日本が重要な貢献をしているが、研究は広範にわたるので、ここでは最大フローアルゴリズムを主に最近の研究動向まで含めて解説したい。さらに興味のある読者は是非上記の文献および後述の参考文献等を参照されたい。

2. 最大フローアルゴリズム

2.1 フローとカット

有向グラフ$G=(V,E)$の各辺$e=(v,w)\in E$に正の実数の容量$cap(e)$が割り振られているネットワーク$N=(G=(V,E),cap,s,t)$において、入口$s\in V$から出口$t\in V$への流量最大のフローを求める問題が最大フロー問題である。なお、Nのフローfは、辺集合Eから非負実数集合R_+への関数で以下の2つの条件をみたすものとして定義される。

(A) 任意の辺$e\in E$に対して、$0\leq f(e)\leq cap(e)$.

(B) s,tを除く任意の頂点$v\in V$に対して、

$$\sum_{e=(u,v)\in\delta^-(v)} f(e)=\sum_{e=(v,w)\in\delta^+(v)} f(e).$$

ここで$\delta^-(v)$ ($\delta^+(v)$)はvを終点(始点)とするNの辺の集合を表す。(A)は各辺を流れるフローの値が容量以下であることを示している。(B)は入口・出口を除いて、流量の保存則(流入量=流出量)が成立していることを示している。従って、(B)は、

$$K=\sum_{e=(s,w)\in\delta^+(s)} f(e)-\sum_{e=(u,s)\in\delta^-(s)} f(e)$$

とおけば、

$$K=\sum_{e=(u,t)\in\delta^-(t)} f(e)-\sum_{e=(t,w)\in\delta^+(t)} f(e)$$

が成立することを意味している。すなわち、sから流出するフローの正味量Kはtに流入するフローの正味量に等しい。このKをフローfの流量といい、$val(f)$と書く。$s\in X, t\in V-X$となる点の部分集合Xに対して

$$E(X,V-X)=\{(v,w)\in E\mid v\in X, w\in V-X\}$$

をs,tカットという。s,tカット$E(X,V-X)$の容量を$cap(X)$で表す($cap(X)=\sum_{e\in E(X,V-X)} cap(e)$)。すると任意のフロー$f$と任意の$s,t$カット$E(X,V-X)$に対して、

$$val(f)\leqq cap(X)$$

が成立する。流量最大のフローfと容量最小のs,tカットを選ぶとこの不等式は等式で成立する。これが有名な最大フロー・最小カット定理である。

定理1. ネットワーク$N=(G=(V,E),cap,s,t)$において、最大のフローの流量と最小のs,tカット容量は等しい。

2.2 Ford-Fulkersonのアルゴリズム

最大フローを求めるFord-Fulkersonのアルゴリズムは、ゼロフローから出発してフローfを更新していくが、その際フローfの残余容量ネットワーク$N(f)$を用いて、入口sから出口tへのパス$P=P(s,t)$を求め、そのパスPに沿ってできるだけフローを流してfを増加し更新して、最終的に$N(f)$にsからtへのパスがなくなるまで繰り返す、というものである。フローfに関する残余容量ネットワーク$N(f)$は以下のようなものである。各辺$e=(v,w)\in E$に対して、$f(e)<cap(e)$ならばさらに$cap_f(e)=cap(e)-f(e)$だけ流せるのでそれが$e=(v,w)$の残余容量であり、$f(e)>0$ならば$f(e)$だけ減らせるので逆向きの仮想的な辺$e^R=(w,v)$を考え、その辺$e^R=(w,v)$に沿って$f(e)$だけ流せるので辺$e^R=(w,v)$の残余容量は$cap_f(e^R)=f(e)$となる。従って、仮想的な辺の全体を$E^R=\{e^R\mid e\in E\}$とすると、フローfに関する残余容量ネットワーク$N(f)=(G(f)=(V,E(f)),cap_f,s,t)$は

$$\begin{aligned} E(f) &= \{e\in E\mid f(e)<cap(e)\} \\ &\quad \cup\{e^R\in E^R\mid f(e)>0\}, \\ cap_f(e) &= cap(e)-f(e)\quad (f(e)<cap(e)), \\ cap_f(e^R) &= f(e)\quad (f(e)>0) \end{aligned}$$

として定義できる。

フローfに関する残余容量ネットワーク$N(f)$におけるsからtへのパス$P(s,t)$は増加パスと呼ばれる。実際、$P(s,t)$上の辺の残余容量の最小値を$\Delta(P)$とし、各辺$e\in E$を流れるフローfを

$$f'(e) := \begin{cases} f(e) + \Delta(P) & (e \in P(s,t)) \\ f(e) - \Delta(P) & (e^R \in P(s,t)) \\ f(e) & (e, e^R \notin P(s,t)) \end{cases}$$

と更新すれば、f'もNのフローとなり、かつ$val(f') = val(f) + \Delta(P)$となるので流量を増加できる($\Delta(P) > 0$に注意されたい)。従って、$f$が$N$の最大フローならば、$f$に関する増加パスは存在しない。逆もいえる。

定理2。ネットワーク$N = (G = (V, E), cap, s, t)$において、フロー$f$が最大フローであるための必要十分条件は$f$に関する増加パスが存在しないことである。

Ford-Fulkersonのアルゴリズムより、各辺eの容量$cap(e)$が整数のときは、残余容量も常に整数となり、従って、更新で得られるフロー$f(e)$も整数となるので、次の整数性定理が得られる。

定理3。(整数性定理)　ネットワーク$N = (G = (V, E), cap, s, t)$において、各辺$e$の容量$cap(e)$が整数ならば、フロー$f(e)$も整数となるような最大フロー$f$が存在する。

2.3 Dinicのアルゴリズム

Ford-Fulkersonのアルゴリズムには欠点があり、辺の容量が無理数であると反復が有限回で終了しなかったり、誤った値に収束したりする。また辺の容量がすべて整数であるとしても反復回数が辺の容量に比例するときもある。その意味で、Ford-Fulkersonのアルゴリズムは多項式オーダのアルゴリズムではない。これに対して、Edmonds-Karpは辺数最小の増加パスを選ぶことでこれらの欠点を解消した。Dinic(ディニッツ)はこの考えをさらに進めて、残余容量ネットワーク$N(f)$を幅優先探索して、各頂点vの入口sからの距離(辺数最小のパスに含まれる辺数)$level[v]$を求めて

$$E_L(f) = \{(v, w) \in E(f) \mid level[w] = level[v] + 1\}$$

の辺からなるレベルネットワーク$N_L(f) = (V, E_L(f))$を構成しレベルネットワーク$N_L(f)$上での極大フローf''を求め、

$$f(a) := \begin{cases} f(a) + f''(a) & (a \in E_L(f)) \\ f(a) - f''(a^R) & (a^R \in E_L(f)) \end{cases}$$

と更新しながら、$N(f)$にsからtへのパスがなくなるまで繰り返して、最終的にNの最大フローを求める方法を提案した。

$N(f)$の極大フローf''が$O(mn)$ $(n = |V|, m = |E|)$の手間で求められること、および反復ごとにsからtへのパスの長さが1以上増加することより反復回数が$O(n)$であることがいえて、Dinicの最大フローアルゴリズムの計算時間は$O(mn^2)$となる。

定理4。ネットワーク$N = (G = (V, E), cap, s, t)$の最大フローは、Dinic法により$O(mn^2)$の手間で求めることができる。

2.4 Dinic以降のアルゴリズム

次の表はこれまでに提案された代表的な最大フローアルゴリズムの性能一欄である[1]。いずれも高速化のためデータ構造に工夫を凝らしているが、実用的なレベルでの性能評価は今後の計算機実験によって明らかにされるものと思われる。

発表年	著者	手間
1951	Dantzig	mn^2U
1955	Ford-Fulkerson	mnU
1970	Edmonds-Karp	m^2n
1970	Dinic	mn^2
1974	Karzanov	n^3
1977	Cherkassky	$n^2m^{1/2}$
1980	Galil-Naamad	$mn(\log n)^2$
1983	Sleator-Tarjan	$mn\log n$
1986	Goldberg-Tarjan	$mn\log(n^2/m)$
1990	Cheriyan et al.	$n^3/\log n$
1990	Alon	$mn + n^{8/3}\log n$
1992	King et al.	$mn + n^{2+\epsilon}$
1993	Phillips et al.	$mn(\log_{m/n} n + \log^{2+\epsilon} n)$
1994	King et al.	$mn\log_{m/(n\log n)} n$

上の表は、最初の2つを除いて強多項式オーダのアルゴリズムである。実際上生じる問題では、容量は整数に近似されて、整数の容量をもつネットワークとして扱われることが多い。そのような場合は、容量の最大値をUとおいてUを表現するためのビット数$\log U$を用いて計算の手間を見積もるのが自然である。このような観点からの最大フローアルゴリズムも多数提案されている。強多項式オーダのアルゴリズムと性能を比較するため、通常$\log U = O(\log n)$の相似仮定(similarity assumption)が用いられている。さらに、この相似仮定のもとで$\log n$の多項式で表現される部分は無視して得られる性能(の限界)を球場限界(ball-park bound)という。たとえば、$O(mn\log U)$などは、球場限界はmnとなる($O^*(mn)$と表示)。

一般に、Dinic法などの増加パスを求めてフローを増加していく方法は、フローを様々な流量をもつパ

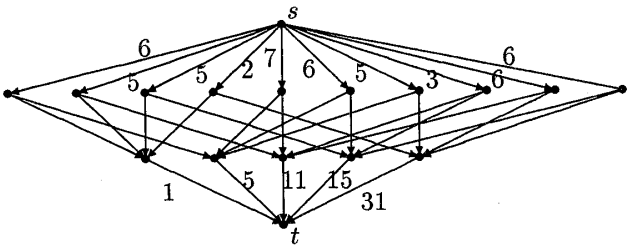

図1. 中日の優勝可能性判定のためのネットワーク

スの和として表現しているとみなせる。従って、フローのパス分解に基づく方法といえるが、パスの総数は$\Omega(mn)$になるため、パス分解法に基づくアルゴリズムでは最悪の場合$\Omega(mn)$の手間はどうしてもかかることになる。これが、パス分解法の障壁といわれるものであるが、これを克服しているのはCheriyan et al.のアルゴリズムだけである。しかしそれも、密ネットワークに限定されている。これに対して、1997年のFOCS(IEEE Symposium on Foundations of Computer Sciences) の会議で、相似仮定の下ではあるが、パス分解法の障壁$O(mn)$を大幅にクリアする$O^*(\min\{m^{3/2}, mn^{2/3}\})$の最大フローアルゴリズムが提案されている。それについては4節で紹介する。

2.5 優勝可能性判定問題への応用

図1は第1節で紹介した中日の優勝可能性を判定するためのネットワークである。上から2段目の点は左から、ヤクルト対広島、ヤクルト対巨人、ヤクルト対横浜、ヤクルト対阪神、広島対巨人、広島対横浜、広島対阪神、巨人対横浜、巨人対阪神、横浜対阪神に対応する。入口sからそれらの点に向かう辺の容量は残り対戦数になっている。一方、上から3段目の点は左から、ヤクルト、広島、巨人、横浜、阪神に対応する。中日が残り試合全勝しても67勝にしかならないので、他のどれかのチームが68勝以上してしまうと中日は優勝できない。そこで上から3段目の点から出口tに向かう辺の容量は、対応するチームが67勝になるまでの勝ち数となっている。他の残りの辺の容量は∞とみなしてよい。このネットワークで最大フローfを求めその値$val(f)$が、s,tカット$E(s,V-s)$の容量$cap(s)$に等しいときは、すべての残り試合を消化することができて、中日が優勝するためのシナリオがフローfで表現されている。残念ながら、この例では、最大フローの流量50となり、$cap(s) = 51$となり、中日が優勝するとすると残り試合1試合消化できなくなってしまう。従ってこの時点で中日の優勝の可能性は完全に消滅している。次の例で阪神の優勝する可能性はどうだろうか。

チーム	勝数	負数	残り試合数
ヤクルト	66	38	26
広島	58	40	32
巨人	56	47	27
横浜	50	47	33
阪神	36	63	31
中日	33	64	33

残り対戦数	ヤクルト	広島	巨人	横浜	阪神	中日
ヤクルト	-	6	5	5	2	8
広島	6	-	7	10	5	4
巨人	5	7	-	3	6	6
横浜	5	10	3	-	9	6
阪神	2	5	6	9	-	9
中日	8	4	6	6	9	-

前と同様にネットワークを作り最大フローを求めると今度は最大フローの流量60となり、$cap(s) = 60$となり、阪神が優勝する可能性があると判断できる。そのときの対戦成績の例は以下の通りである(各行がそのチームの対戦勝ち数である)。

	ヤクルト	広島	巨人	横浜	阪神	中日
ヤクルト	-	1	0	0	0	0
広島	5	-	4	0	0	0
巨人	5	3	-	3	0	0
横浜	5	10	0	-	0	2
阪神	2	5	6	9	-	9
中日	8	4	6	4	0	-

上記の問題では、一般にNチーム存在するとき、ネットワークの頂点数n、辺数mはともに$O(N^2)$となり、Dinicの方法で解くと手間は$O(mn^2) = O(N^6)$となりそうだが、第3段目の点数が$O(N)$であるのでsからtへのパスの長さは最大で$O(N)$であり、したがって、反復回数も$O(N)$となり、全体の手間は$O(N^5)$である。より厳密に解析すれば、おそらくさらによい手間が得られるであろう。

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$
$T1$	4	4	3	2	4	2
$T2$	4	5	7	4	6	5
$T3$	1	2	4	1	3	1
$T4$	6	5	5	4	7	6
$T5$	5	7	8	4	6	7
$T6$	1	2	2	1	4	2

3. 最適割り当て問題

第1節で述べた得意度の表で、コスト= 10−得意度として書き換えると上の表が得られる。ここで各行iを左端点iに対応させ、各列jを右端点jに対応させて完全2部グラフ$K_{6,6}$を考え、左端点iから右端点jに向かう有向辺のコストを上の表のij要素を割り当てる。そして入口sから各左端点へ向かう辺と各右端点から出口tへ向かう辺を加え、それらの辺のコストを0とする。すべての辺の容量を1とし、sからtへの流量6の費用最小のフローを求める。流量1なら話は単純で単にsからtへの(辺のコストを長さと見なして) 最短パスを求める問題になる。このフローfに対して残余容量ネットワーク$N(f)$を作り、逆向きの辺e^R(に沿ってフローを流すと費やしたコストが戻ってくるので) のコストを$-cost(e)$と定め、またsからtへの最短パスを求め、フローfを更新する。こうして、各流量に対して費用最小のフローが得られる。これは伊理によって提案された方法であるが、辺の長さが負のものが存在するので、最短パスを求める通常のDijkstra法が適用できない。冨澤は、距離関数という最小費用フロー問題の双対問題の変数に対応する変数を導入して、負の長さの辺をなくし通常のDijkstra法ができるようにした。この手法を用いて上記の割り当て問題を解くと、完全2部グラフの辺eでフロー$f(e) = 1$の辺の集合Mは最適割り当て問題の解になる。上の例では、$M = \{(T1,C6),(T2,C2),(T3,C5),(T4,C3),(T5,C4),(T6,C1)\}$となり、総得意度=40(コスト=20)となる。

最適割り当て問題は、この例のように、教師数N、科目数Nのときは最短パス問題をDijkstra法で$O(N)$回解けば得られるので手間は(N^3)である。一般の最小費用フロー問題に対しても藤重らが強多項式オーダのアルゴリズムを与えている。

4. 最近の最大フローアルゴリズム

相似仮定の下でパス分解法の障壁$O(mn)$を大幅にクリアする$O^*(\min\{m^{3/2}, mn^{2/3}\})$の最大フローアルゴリズムについて紹介する。これは1997年のFOCSの会議でGoldberg-Raoによって提案されたものである[1]。まず、いくつかの記法を導入する。便宜上、辺$e = (i,j)$に対して逆向きの辺$e^R = (j,i)$が存在しないときは、仮想的にe^Rを考えその容量を0と考え、常に逆向きの辺があるものとする。もちろん逆向きの辺e^Rが元から存在するときは容量はそのままである。そして、フローfは$f(e) = 0$あるいは$f(e^R) = 0$のいずれかをつねにみたすものとする。したがって、各辺の残余容量は$cap_f(e) = cap(e) - f(e) + f(e^R)$、$cap_f(e^R) = cap(e^R) - f(e^R) + f(e)$となる。残余容量ネットワーク$N(f)$は残余容量が正の辺のみからなる。各辺$e$に0,1の長さを割り振る関数$\ell$が与えられたとき、各点$i$のラベル$d(i)$を用いて各辺$(i,j)$の簡約コスト$\ell_d(i,j) = \ell(i,j) + d(j) - d(i)$を定義し、つねに簡約コストが非負であるようにする。$d(t) = 0$として、すべての辺の簡約コストを非負にできるとき、点iのラベル$d(i)$を距離ラベルと呼ぶ。長さ関数ℓに対する各点iの距離$d_\ell(i)$をiからtまでの最短距離として定義する。するとd_ℓは距離ラベルとなり、任意の距離ラベルdに対して$d \le d_\ell$ (すべての点iで$d(i) \le d_\ell(i)$) が成立する。

最大フローの流量と現在の流量の差の上界をFを用いて評価する。辺の容量の最大値をUとおけば最初、$F \le \sum_{(s,w)\in\delta^+(s)} cap(s,w) \le nU$である。Goldberg-RaoのアルゴリズムではこのFを各フェーズで半分になるまで更新を繰り返す。$F < 1$となれば、最大フローが求まったことになるのでフェーズの回数は高々$1+\log(nU)$である。各フェーズの最初のFに応じてΔを

$$\Delta = \lceil \frac{F}{\min\{m^{1/2}, n^{2/3}\}} \rceil$$

として定義し、流量Δのフローf'あるいは(そのようなフローがないときには) 流量Δ未満の極大フローf'を求めて、流量fを増加し更新する。各フェーズでは流量Δのフローf'での更新は高々$\min\{m^{1/2}, n^{2/3}\}$回、極大フローf'での更新は高々 $6\min\{m^{1/2}, n^{2/3}\}$回しか起きないことがいえる(後述)。さらに流量$\Delta$のフロー$f'$あるいは流量$\Delta$未満の極大フロー$f'$を求めることは$O(m\log(n^2/m))$の手間でできるので全体の手間は

$O(m\min\{m^{1/2}, n^{2/3}\}\log(n^2/m)\log(nU))$ となる。

実際に流量Δのフローf'あるいは流量Δ未満の極大フローf'を求めるネットワークはこれまでの単なる残余容量ネットワーク$N(f)$とは少し異なる。残余容量ネットワーク$N(f)$の各辺eに対して長さ$\ell(e)$を以下のように定義する。

$$\ell(e) = \begin{cases} 0 & (cap_f(e) \geq 3\Delta) \\ 1 & (cap_f(e) < 3\Delta). \end{cases}$$

さらにℓに基づいて点iからtへの距離$d_\ell(i)$を計算し、

$$d_\ell(i) > d_\ell(j) \text{ または } d_\ell(i) = d_\ell(j) \text{ かつ } \ell(i,j) = 0$$

を満たす辺(i,j)を使用可能辺といい、使用可能辺のみを使ってフローを増加更新していく。なお、使用可能辺は各点からtへの最短パスのいずれかに含まれる辺である。便宜上、使用可能辺の集合を$A(f,\ell,d_\ell)$と表す。使用可能辺からなるグラフ$G_A = (V, A(f,\ell,d_\ell))$は正の長さの閉路をもたないことがいえる。さらに、

$$2\Delta \leq cap_f(i,j) < 3\Delta, d(i) = d(j), cap_f(j,i) \geq 3\Delta$$

を満たすような辺(i,j)を特別な辺と見なして上記の長さ関数ℓを下記の長さ関数$\bar{\ell}$に変更する。

$$\bar{\ell}(e) = \begin{cases} 0 & (e\text{は特別な辺}) \\ \ell(e) & (\text{それ以外}). \end{cases}$$

もちろんこのように変えても、$d_\ell = d_{\bar{\ell}}$は成立する。使用可能辺からなるグラフ$G_A = (V, A(f,\bar{\ell},d_{\bar{\ell}}))$の各強連結成分(長さ0の辺から形成されている)をそれぞれ1点に縮約して得られるグラフの各辺の容量を$N(f)$の対応する辺の残余容量としてできるネットワーク$N_A(f)$上でsからtへの流量Δのフローf'あるいは極大フローf'を求める($O(m\log(n^2/m))$の手間でできる)。こうして、フローを増加更新していく。もちろんフローを更新するときは、縮約した点を元の点に戻して行う。これは各強連結成分の中から1点を代表点に選び、その点を根とする2種類の根付き木を用いて調整できる。更新されたフローf'に対応する長さ関数ℓ'を用いてさらに上記のことを繰り返す。そして、$d_\ell(s)$個の特殊なs,t-カット(S_k,T_k) $(k = 1,2,...,d_\ell(s)$, $s \in S_k = \{v \in V \mid d_\ell(v) \geq k\}$, $t \in T_k = V - S_k)$のうち容量最小のもの(S,T)を求め($O(m)$でできる)、容量$cap(S,T) = \{cap(i,j) \mid i \in S, j \in T\}$が$cap(S,T) \leq F/2$を満たすようになったときそのフェーズを終了し、新しいフェーズに$F := cap(S,T)$として入るものとする。

詳しい証明は省くが、極大フローf'による更新では$d_{\ell'}(s) > d_\ell(s)$が成立し、さらにd_ℓは長さ関数ℓ'に関しても距離ラベルの条件$(\ell_{d_{\ell'}}(i,j) = \ell'(i,j) + d_\ell(j) - d_\ell(i) \geq 0)$を満たしている。さらに、残余容量ネットワーク$N(f)$の各辺$e$の残余容量$cap_f(e)$を幅と見なして長さ$\ell(e)$との積$cap_f(e)\ell(e)$をその辺の体積と考え、全体積$Vol_{f,\ell} = \sum_{e\in N(f)} cap_f(e)\ell(e)$を用いて、最大フローの流量と現在の流量の差の上界$F$を$F \leq Vol_{f,\ell}/d_\ell(s)$と評価できる。$N(f)$において$s$から$t$への任意のパスに沿って流量1だけフローを増加するとd_ℓだけ体積が減るからである(体積は定義より非負)。こうして各段階で辺の残余容量の最大値をMとおくと、$Vol_{f,\ell} \leq mM$、$cap(S,T) \leq mM/d_\ell(s)$を満たす。さらに、$cap(S,T) \leq (\frac{2n}{d_\ell(s)})^2 M$も示せて、この2つから極大フロー$f'$による更新は高々各フェーズで$\min\{6m^{1/2}, 5n^{2/3}\}$回であることが示せる。

定理4. ネットワークNの最大フローは、$O(m\min\{m^{1/2}, n^{2/3}\}\log(n^2/m)\log(nU))$の手間で求めることができる($U$は$N$の辺の最大容量)。

5. おわりに

他の最適化問題に対しても重要な成果が発表されている。M. Thorupによる無向ネットワークの最短パスを$O(m)$の手間で求めるアルゴリズムおよびB. Chazelleによる$O(m\alpha(m,n)\log\alpha(m,n))$の手間で最小スパンニング木を求めるアルゴリズムがそれである($\alpha(m,n)$はアッカーマン関数の逆関数)。これらは、従来の壁を破る画期的な成果であるが、文献[1]と同じプロシーディングに載っている。またネットワークアルゴリズムのプログラムや実用化アルゴリズムが載せてある東大の松井知己先生とA.V. Goldbergのホームページを以下に紹介しておく。極めて有用であるのでご興味のある方は参考にしていただきたい。

http://www.misojiro.t.u-tokyo.ac.jp/ tomomi/opt-code.html

http://www.neci.nj.nec.com/homepages/avg/index.html

参考文献

[1] Goldberg A. V. and Rao S.: " Beyond the Flow-Decomposition Barrier", Proc. 38th IEEE Symposium Foundations of Computer Sceince, Miami Beach, 1997, pp. 2—11.