7 最大流問題

グラフ $G=(V,E)$ の各枝 $(i,j)\in E$ に容量 u_{ij} が与えられたネットワーク上で特定の２頂点，入口 s と出口 t の間に流すことのできる総流量 v を最大化する流れ $\mathbf{x}^*=(x^*_{ij})$ が**最大流** (maximum flow) で，これを求める最大流問題は次のように定式化されます:

$$\left|\begin{array}{ll} \text{最小化} & v \\ \\ \text{条　件} & \displaystyle\sum_{\{j|(i,j)\in E\}} x_{ij} - \sum_{\{j|(j,i)\in E\}} x_{ji} = \begin{cases} v, & i=s \\ 0, & \forall i\in V\setminus\{s,t\} \\ -v, & i=t \end{cases} \\ \\ & 0\le x_{ij}\le u_{ij},\quad \forall (i,j)\in E. \end{array}\right. \tag{7.10}$$

最短路問題 (6.4) がネットワーク流の枝の費用のみを考慮したモデル化であるのに対し，最大流問題 (7.10) は枝の費用を無視して容量だけを考慮しており，両者は互いを補完する関係にあります．また，問題自体がそうであるように (7.10) を解くアルゴリズムは最小費用流問題 (1.2) のアルゴリズムの特殊化で，これを逆に一般化すれば (1.2) の解決が可能となります．

問題 (7.10) のアルゴリズムは大きく２つに分類することができる:

増量可能路法 (augmenting-path algorithm).　入口 s と出口 t 以外の各頂点における流量保存条件を常に保ち，s から t への増量可能な有向路に沿って流量を徐々に増加させる．

予備流押し出し法 (preflow-push algorithm).　ネットワークへ一度に枝の容量の総和を流し込み，各頂点の流量保存条件を超過した流れは出口 t か，あるいは入口 s へ押し出す．

ここでは，前節に紹介した最短路問題のアルゴリズムをサブルーチンとして利用できる増量可能路法について説明することにします．その前に，これら２つのアルゴリズムが拠り所とする (7.10) の基本構造を調べておきましょう．

7.1 流れとカット

まず，増量可能路法で中心的な役割を演じる２つの概念を定義しましょう:

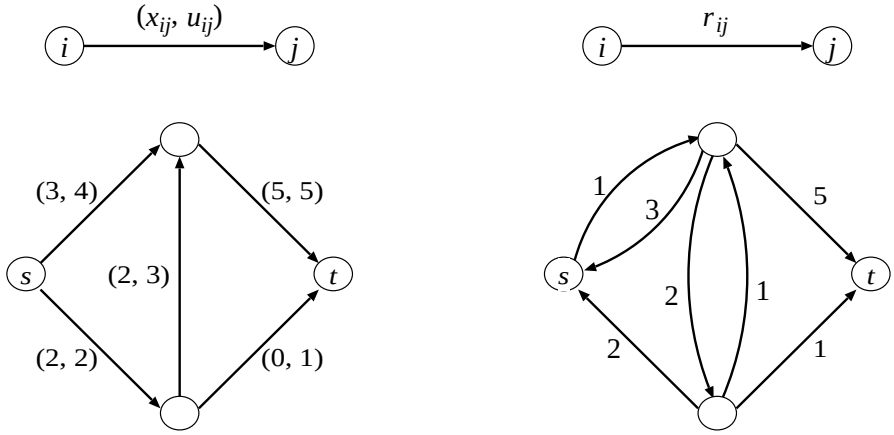

図 7.5: 流れ $\mathbf{x}$ のあるネットワークと残余ネットワーク $N(\mathbf{x})$.

残余ネットワーク． 問題 (7.10) の流れ $\mathbf{x} = (x_{ij})$ に対し，次の規則にしたがって枝集合 $E(\mathbf{x})$ とその容量 $\mathbf{r} = (r_{ij} \mid (i,j) \in E(\mathbf{x}))$ を定義します:

(a) $u_{ij} - x_{ij} > 0$ ならば $(i,j) \in E(\mathbf{x})$ とし，その容量を $r_{ij} = u_{ij} - x_{ij}$ とする．

(b) $x_{ij} > 0$ ならば $(j,i) \in E(\mathbf{x})$ とし，その容量を $r_{ji} = x_{ij}$ とする．

この結果えらえる有向グラフ $G(\mathbf{x}) = (V, E(\mathbf{x}))$ と容量 $\mathbf{r} = (r_{ij})$ をあわせて**残余ネットワーク** (residual network) とよび，これを $N(\mathbf{x})$ で表します．

図 7.5 の左のネットワークに流れ $\mathbf{x}$ が与えられれば，その残余ネットワーク $N(\mathbf{x})$ は右の図のようになります．残余ネットワークは，つまり現在の流れ $\mathbf{x}$ から増減が可能な方向と量を示します．

***s*-*t* カット．** 頂点集合 $W \subset V$ (ただし，$W \neq \emptyset, W \neq V$) に対し，$W$ とその補集合 $\overline{W} = V \backslash W$ を接続する枝集合を**カット** (cut) とよび，$[W, \overline{W}]$ で表します．特に $s \in W$, $t \in \overline{W}$ のとき，$[W, \overline{W}]$ を *s*-*t* **カット** (*s*-*t* cut) とよびますが，これは**前進枝** (forward arc) の集合

$$(W, \overline{W}) = \{(i,j) \in E \mid i \in W,\ j \in \overline{W}\}$$

と**後退枝** (backward arc) の集合

$$(\overline{W}, W) = \{(i,j) \in E \mid i \in \overline{W},\ j \in W\}$$

の和集合として表されます:

$$[W, \overline{W}] = (W, \overline{W}) \cup (\overline{W}, W).$$

前進枝の容量の総和

$$u[W, \overline{W}] = \sum_{(i,j)\in(W,\overline{W})} u_{ij}$$

を s-t カット $[W, \overline{W}]$ の**容量** (capacity) といいます．また，残余ネットワークにおける s-t カット $[W, \overline{W}]$ の容量

$$r[W, \overline{W}] = \sum_{(i,j)\in(W,\overline{W})} r_{ij}$$

をその**残余容量** (residual capacity) といいます．

さて，$\mathbf{x} = (x_{ij})$ を問題 (7.10) の任意の実行可能流とし，このときの入口 s から出口 t への総流量を v としましょう．この総流量 v を流れ $\mathbf{x}$ の**値** (value) とよびます．任意の s-t カット $[W, \overline{W}]$ を横切る流れ $\mathbf{x}$ は正味

$$x[W, \overline{W}] = \sum_{(i,j)\in(W,\overline{W})} x_{ij} - \sum_{(i,j)\in(\overline{W},W)} x_{ij}$$

です．一方，頂点 $i \in W$ に関する流量保存条件をすべて足しあわせると

$$\begin{aligned} v &= \sum_{i\in W} \left(\sum_{\{j|(i,j)\in E\}} x_{ij} - \sum_{\{j|(j,i)\in E\}} x_{ji} \right) \\ &= \sum_{i\in W} \left(\sum_{\{j\in W|(i,j)\in E\}} x_{ij} + \sum_{\{j\in \overline{W}|(i,j)\in E\}} x_{ij} - \sum_{\{j\in W|(j,i)\in E\}} x_{ji} - \sum_{\{j\in \overline{W}|(j,i)\in E\}} x_{ji} \right) \\ &= \sum_{i\in W} \sum_{\{j\in \overline{W}|(i,j)\in E\}} x_{ij} - \sum_{i\in W} \sum_{\{j\in \overline{W}|(i,j)\in E\}} x_{ji} \end{aligned}$$

となって，

$$v = x[W, \overline{W}] = \sum_{(i,j)\in(W,\overline{W})} x_{ij} - \sum_{(i,j)\in(\overline{W},W)} x_{ij} \tag{7.11}$$

が成り立つことがわかります．ここで，容量条件 $0 \le x_{ij} \le u_{ij}$, $(i,j) \in [W, \overline{W}]$ を使えば，

$$v = x[W, \overline{W}] \le \sum_{(i,j)\in(W,\overline{W})} u_{ij} = u[W, \overline{W}] \tag{7.12}$$

となります．

性質 7.7.　任意の実行可能流 $\mathbf{x}$ の値 v と任意の s-t カット $[W, \overline{W}]$ の容量との間には次の関係が成り立つ:

$$v \leq u[W, \overline{W}].$$

この性質から直ちに最適性の十分条件が得られます.

性質 7.8.　実行可能流 $\mathbf{x}$ の値 v に対して

$$v = u[W, \overline{W}] \tag{7.13}$$

を満たす s-t カット $[W, \overline{W}]$ が存在すれば，$\mathbf{x}$ は最大流である.

この２つの性質 7.7, 7.8 を，残余ネットワークを使って言い換えてみましょう．ある $\Delta v \geq 0$ に対し，値 $v + \Delta v$ の実行可能流 $\mathbf{x}'$ が存在するものとします．上の議論における実行可能流 $\mathbf{x}$ の任意性により，(7.12) から

$$v + \Delta v = x'[W, \overline{W}] \leq \sum_{(i,j) \in (W, \overline{W})} u_{ij}.$$

これより (7.11) を辺々引いて

$$\Delta v \leq \sum_{(i,j) \in (W, \overline{W})} (u_{ij} - x_{ij}) + \sum_{(i,j) \in (\overline{W}, W)} x_{ij}$$

が得られます．したがって，流れ $\mathbf{x}$ に対する残余ネットワーク $N(\mathbf{x})$ の定義 (a), (b) から，

$$\Delta v \leq \sum_{(i,j) \in (W, \overline{W})} r_{ij} = r[W, \overline{W}]$$

が導かれます.

性質 7.9.　任意の実行可能流 $\mathbf{x}$ に対する任意の s-t カットの残余容量と入口 s から出口 t へ向かって増量可能な値 Δv との間には次の関係が成り立つ:

$$\Delta v \leq r[W, \overline{W}].$$

性質 7.10.　実行可能流 $\mathbf{x}$ に対し，

$$r[W, \overline{W}] = 0 \tag{7.14}$$

を満たす s-t カット $[W, \overline{W}]$ が存在すれば，$\mathbf{x}$ は最大流である.

7.2 増量可能路法と最大流最小カット定理

与えられた実行可能流 $\mathbf{x}$ に対する残余ネットワーク $N(\mathbf{x})$ に入口 s から出口 t への有向路 $P \subset E(\mathbf{x})$ が存在すれば，それに沿って総流量を増加させることができます．実際，

$$\delta = \min\{r_{ij} \mid (i,j) \in P\} > 0 \tag{7.15}$$

を用い，もとのネットワークの各枝 $(i,j) \in E$ に対して

$$x'_{ij} = \begin{cases} x_{ij} + \delta, & (i,j) \in P \text{ [定義 (a) による枝 } (i,j) \in E(\mathbf{x})\text{] の場合} \\ x_{ij} - \delta, & (j,i) \in P \text{ [定義 (b) による枝 } (j,i) \in E(\mathbf{x})\text{] の場合} \\ x_{ij}, & \text{それ以外の場合,} \end{cases} \tag{7.16}$$

の修正を加えれば，$\mathbf{x}'$ は再び (7.10) の実行可能流となり，ネットワークの総流量が δ だけ増加します．残余ネットワーク $N(\mathbf{x})$ 上の，このような有向路 P を**増量可能路** (augmenting paht) とよびます．増量可能路法は，残余ネットワークに増量可能路が存在しなくなるまで実行可能流と残余ネットワークの更新を繰り返し，値 v をその最大値にまで増加させる方法です．

algorithm AUGMENTING_PATH

$\mathbf{x} := \mathbf{O}$;
while $N(\mathbf{x})$ に増量可能路 P が存在 do begin
  $\delta := \min\{r_{ij} \mid (i,j) \in P\}$;
  P に沿って流れを δ 増加させ，$\mathbf{x}$ と $N(\mathbf{x})$ を更新する
end
end;

入口 s からの有向路が $N(\mathbf{x})$ 上に存在する頂点の集合を W で表せば，出口 t が

$$t \in \overline{W} = V \setminus W$$

となった時点でアルゴリズムは終了します．このとき，$[W, \overline{W}]$ はもとのグラフ G の s-t カットを構成しますが，その残余容量 $r[W, \overline{W}]$ はゼロで，性質 7.10 から $\mathbf{x}$ の最適性が保証されま

す．ところが，性質 7.8（および 7.10）は最適性の十分性を与えているにすぎず,「最大流 $\mathbf{x}$ のある場合は必ず (7.13)（および (7.14)）を満たす s-t カット $[W, \overline{W}]$ が存在する」ことを保証しなければ，アルゴリズム AUGMENTING_PATH の正当性も認められません．

定理 7.11 [**最大流最小カット定理**]. 入口頂点 s から出口頂点 t への流れの最大値は，すべての s-t カットの容量の最小値に等しい．

証明: 最大流を $\mathbf{x}^* = (x^*_{ij})$，その値を v^* としよう（性質 7.7 より，容量が有限のカットが存在すれば，最大流は保証される）．このときの残余ネットワーク $N(\mathbf{x}^*)$ には頂点 s から t への有向路は存在しない．なぜなら，そのような有向路 $P \subset E(\mathbf{x}^*)$ が存在したとすれば，(7.15), (7.16) にしたがって v^* を $v^* + \delta$ に改善でき，$\mathbf{x}^*$ が最大流であることに矛盾する．そこで，$N(\mathbf{x}^*)$ において頂点 s からの有向路が存在する頂点の集合を W^* で表すことにすれば，

$$s \in W^*, \quad t \in \overline{W}^* = V \setminus W^*$$

であり，$[W^*, \overline{W}^*]$ は s-t カットとなる．さらに，残余ネットワークの定義 (a), (b) より

$$x^*_{ij} = \begin{cases} u_{ij}, & (i,j) \in (W^*, \overline{W}^*) \text{ の場合} \\ 0, & (i,j) \in (\overline{W}^*, W^*) \text{ の場合}, \end{cases}$$

であることもわかる．これより，

$$\begin{aligned} v^* = x[W^*, \overline{W}^*] &= \sum_{(i,j) \in (W^*, \overline{W}^*)} x^*_{ij} - \sum_{(i,j) \in (\overline{W}^*, W^*)} x^*_{ij} \\ &= \sum_{(i,j) \in (W^*, \overline{W}^*)} u_{ij} \\ &= u[W^*, \overline{W}^*] \end{aligned}$$

が成り立ち，性質 7.7 とともに定理は示された． ■

さらに，アルゴリズム AUGMENTING_PATH を使えば，次の結果も示すことができます:

定理 7.12. 枝 $(i,j) \in E$ の容量 u_{ij} がすべて整数ならば，整数の最大流が存在する．

証明: まず，AUGMENTING_PATH の任意の反復における実行可能流 $\mathbf{x}$ が整数ベクトルであることを帰納的に示そう．流れ $\mathbf{x}$ が整数ベクトルであることを仮定すれば，残余ネットワークの定義 (a), (b) より，$N(\mathbf{x})$ の各枝 $(i,j) \in E(\mathbf{x})$ の容量 r_{ij} もすべて整数となる．したがって，(7.15) から定まる δ も整数，(7.16) によって定まる次の反復の実行可能流 $\mathbf{x}'$ も整数ベクトルとなる．

さて，δ は整数なので，流れの値 v は 1 回の反復で少なくとも 1 単位増加する．ところが，どの s-t カット $[W, \overline{W}]$ の容量 $u[W, \overline{W}]$ も整数で有限の値をとることから，性質 7.7 により AUGMENTING_PATH は有限回の反復ののちに終了することがわかる．このときの流れ，つまり最大流も整数ベクトルである． ■

定理 7.11 および 7.12 の証明から，最大流が存在すれば AUGMENTING_PATH によって有限時間のうちに最大流 $\mathbf{x}^*$ と容量最小の s-t カット $[W^*, \overline{W}^*]$ の得られることがわかります．また，すべての枝の容量 u_{ij} が整数の場合，その有限時間収束性も次のようにして示されます: 容量 u_{ij} の上限を U とすれば，カット $[\{s\}, V \setminus \{s\}]$ の容量は高々nU です; アルゴリズムの各反復では，増量可能路が見つかれば，それに沿って少なくとも 1 単位は値が増加すます: したがって，アルゴリズムの反復回数は最大でも nU 回にすぎません．増量可能路の探索に，例えばダイクストラ法を用いれば，最悪の場合でも n^3U に比例する基本演算で終了します．

演習問題

7.1 次のグラフで与えられるネットワークで，入口 1 から出口 7 への最大流を求めなさい．