

13.1.2 ベルマン・フォード法 (Bellman-Ford Method)

ベルマン・フォード法は枝の重みを表す関数 c が負の値をとる時にも適用できる 1 始点最短路アルゴリズムである。

ベルマン・フォード法

ステップ 1: $\bar{S} := V$, $d(i) := \infty$ ($i = 1, 2, \dots, n, t$), $d(s) := 0$ と $\text{prev}(s) := 0$, $k := 0$ とおく。

ステップ 2: $k := n + 2$ ($= |V|$) であれば終了, $d(t)$ が最短距離の長さで, その経路は $t, \text{prev}(t), \text{prev}(\text{prev}(t)), \dots$ を逆にたどっていけば良い。

ステップ 3: $\bar{S}$ から適当な頂点 i を選択し, $\bar{S} := \bar{S} \setminus \{i\}$ とおく。

全ての $(i, j) \in E$ について, もし $d(j) > d(i) + c_{ij}$ ならば $d(j) := d(i) + c_{ij}$, $\text{prev}(j) := i$ とおく。

ステップ 4: $\bar{S} \neq \emptyset$ ならばステップ 3 へ。そうでなければ $k := k + 1$, $\bar{S} := V$ とおき, ステップ 2 へ。

ベルマン・フォード法にて $\bar{S}$ はまだその最短距離が求まっていない頂点の集合を表している。 $d(i)$ は始点 s から頂点 i までの最短距離の上界値を表し, $\text{prev}(j)$ は最短経路において頂点 j に到達する前の頂点の候補を表している。

もし与えられたネットワークに負な閉路が存在した場合, その閉路を通れば通るほど距離が短くなるので, 最短路問題に解が存在しなくなる。

ベルマン・フォード法は枝の重みを表す関数 c が負の値をとる場合においても始点 s から他の全ての頂点への最短路を正しく求め, その計算量は $O(|V||E|)$ であることが分かる。

例 13.3 図 13 のネットワークに対して, 始点 s から終点 t をベルマン・フォード法を用いて解いてみよう。

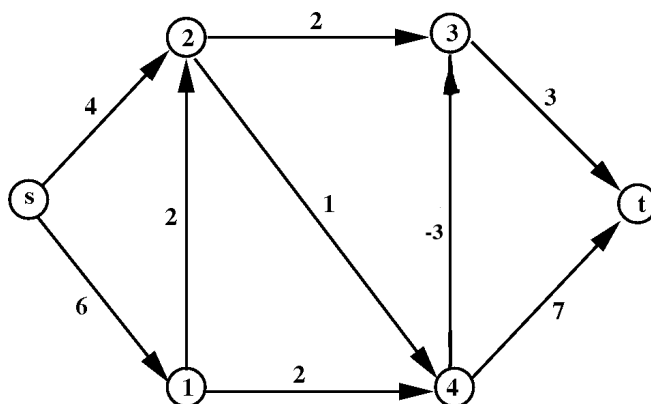


Figure 13: 枝の重みが負な最短路問題。

初期化

$$\bar{S} = \{s, 1, 2, 3, 4, t\}$$

$$\begin{aligned} d(s) &= 0, & d(1) &= \infty, & d(2) &= \infty, & d(3) &= \infty, & d(4) &= \infty, & d(t) &= \infty \\ \text{prev}(s) &= 0, & \text{prev}(1) &=, & \text{prev}(2) &=, & \text{prev}(3) &=, & \text{prev}(4) &=, & \text{prev}(t) &= \\ (k &= 0) \end{aligned}$$

$i = s$ を選択

$$\begin{aligned}\bar{S} &= \{1, 2, 3, 4, t\} \\ d(s) &= 0, \quad d(1) = 6, \quad d(2) = 4, \quad d(3) = \infty, \quad d(4) = \infty, \quad d(t) = \infty \\ prev(s) &= 0, \quad prev(1) = s, \quad prev(2) = s, \quad prev(3) =, \quad prev(4) =, \quad prev(t) = \\ (k &= 0)\end{aligned}$$

$i = 1$ を選択

$$\begin{aligned}\bar{S} &= \{2, 3, 4, t\} \\ d(s) &= 0, \quad d(1) = 6, \quad d(2) = 4, \quad d(3) = \infty, \quad d(4) = 8, \quad d(t) = \infty \\ prev(s) &= 0, \quad prev(1) = s, \quad prev(2) = s, \quad prev(3) =, \quad prev(4) = 1, \quad prev(t) = \\ (k &= 0)\end{aligned}$$

$i = 2$ を選択

$$\begin{aligned}\bar{S} &= \{3, 4, t\} \\ d(s) &= 0, \quad d(1) = 6, \quad d(2) = 4, \quad d(3) = 6, \quad d(4) = 5, \quad d(t) = \infty \\ prev(s) &= 0, \quad prev(1) = s, \quad prev(2) = s, \quad prev(3) = 2, \quad prev(4) = 2, \quad prev(t) = \\ (k &= 0)\end{aligned}$$

$i = 3$ を選択

$$\begin{aligned}\bar{S} &= \{4, t\} \\ d(s) &= 0, \quad d(1) = 6, \quad d(2) = 4, \quad d(3) = 6, \quad d(4) = 5, \quad d(t) = 9 \\ prev(s) &= 0, \quad prev(1) = s, \quad prev(2) = s, \quad prev(3) = 2, \quad prev(4) = 2, \quad prev(t) = 3 \\ (k &= 0)\end{aligned}$$

$i = 4$ を選択

$$\begin{aligned}\bar{S} &= \{t\} \\ d(s) &= 0, \quad d(1) = 6, \quad d(2) = 4, \quad d(3) = 2, \quad d(4) = 5, \quad d(t) = 9 \\ prev(s) &= 0, \quad prev(1) = s, \quad prev(2) = s, \quad prev(3) = 4, \quad prev(4) = 2, \quad prev(t) = 3 \\ (k &= 0)\end{aligned}$$

$i = t$ では何もすることがなく終了. $k := 1$, $\bar{S} := \{s, 1, 2, 3, 4, t\}$ とおく.

$i = s$ を選択

$$\begin{aligned}\bar{S} &= \{1, 2, 3, 4, t\} \\ d(s) &= 0, \quad d(1) = 6, \quad d(2) = 4, \quad d(3) = 2, \quad d(4) = 5, \quad d(t) = 9 \\ prev(s) &= 0, \quad prev(1) = s, \quad prev(2) = s, \quad prev(3) = 4, \quad prev(4) = 2, \quad prev(t) = 3 \\ (k &= 1)\end{aligned}$$

$i = 1$ を選択

$$\begin{aligned}\bar{S} &= \{2, 3, 4, t\} \\ d(s) &= 0, \quad d(1) = 6, \quad d(2) = 4, \quad d(3) = 2, \quad d(4) = 5, \quad d(t) = 9 \\ prev(s) &= 0, \quad prev(1) = s, \quad prev(2) = s, \quad prev(3) = 4, \quad prev(4) = 2, \quad prev(t) = 3 \\ (k &= 1)\end{aligned}$$

$i = 2$ を選択

$$\begin{aligned}\bar{S} &= \{3, 4, t\} \\ d(s) &= 0, \quad d(1) = 6, \quad d(2) = 4, \quad d(3) = 2, \quad d(4) = 5, \quad d(t) = 9 \\ prev(s) &= 0, \quad prev(1) = s, \quad prev(2) = s, \quad prev(3) = 4, \quad prev(4) = 2, \quad prev(t) = 3 \\ (k &= 1)\end{aligned}$$

$i = 3$ を選択

$$\bar{S} = \{4, t\}$$

$$\begin{aligned} d(s) &= 0, & d(1) &= 6, & d(2) &= 4, & d(3) &= 2, & d(4) &= 5, & d(t) &= 5 \\ prev(s) &= 0, & prev(1) &= s, & prev(2) &= s, & prev(3) &= 4, & prev(4) &= 2, & prev(t) &= 3 \\ (k &= 1) \end{aligned}$$

$i = 4$ を選択

$$\bar{S} = \{t\}$$

$$\begin{aligned} d(s) &= 0, & d(1) &= 6, & d(2) &= 4, & d(3) &= 2, & d(4) &= 5, & d(t) &= 5 \\ prev(s) &= 0, & prev(1) &= s, & prev(2) &= s, & prev(3) &= 4, & prev(4) &= 2, & prev(t) &= 3 \\ (k &= 1) \end{aligned}$$

$i = t$ では何もすることがなく終了. $k := 2$, $\bar{S} := \{s, 1, 2, 3, 4, t\}$ とおく. しかし, それ以降の反復では終了条件 $k = 6$ になるまで変わらないので, 結局, 最短路の距離は 5 で最短路は $s \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow t$ となる.

13.2 演習問題

演習問題 13.1 例 13.1 に登場したネットワークに対してダイクストラ法を施せ.

演習問題 13.2 図 14 のネットワークに対して s から t までの最短距離とその長さを求めよ.

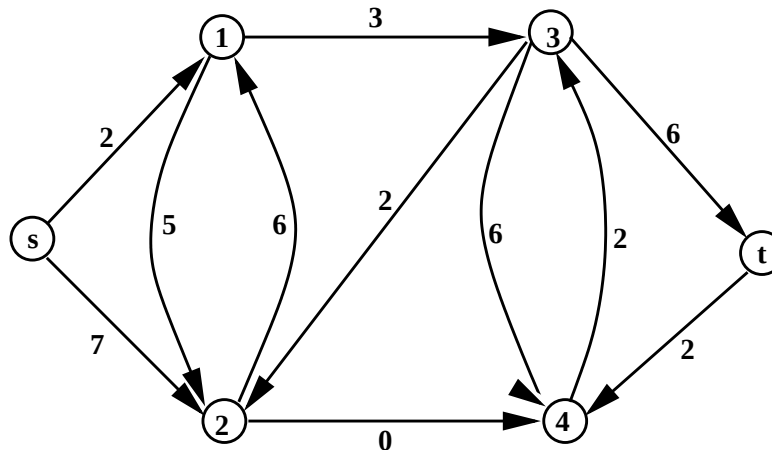


Figure 14: 最短路問題.

演習問題 13.3 演習問題 13.2 のネットワークにおいて, 重さ 5 の枝の重みを -5 に変更して s から t までの最短距離とその長さを求めよ.

演習問題 13.4 あるバス会社で勤務するバス乗務員の 勤務時間帯あたり の経費が表 41 で与えられているとする.

- 9時から17時までに最低でも1人のバス乗務員が勤務し, 経費を最小におさえるスケジューリング問題を最短路問題として定式化せよ. つまり, 問題の解はどのようなネットワークに対する最短路問題を解けばよいのか明記せよ.
- どのようなスケジューリングをすればよいか (解を) 求めよ.

Table 41: バス乗務員の時間帯あたりの経費.

乗務員	A	B	C	D	E	F	G
勤務時間帯	9 時-13 時	9 時-11 時	12 時-15 時	12 時-17 時	13 時-16 時	13 時-15 時	16 時-17 時
経費	31	18	21	33	23	20	11

14 最大流問題 (Maximum Flow Problem)

最大流問題とは与えられたネットワークの各枝に流れるフロー量に制限がある時、その量を最大にしようとする問題である。道路の交通量や鉄道輸送量などの上限値を求めるような状況が丁度当てはまる。

14.1 最大流問題と諸性質

ネットワークを形成する頂点の集合を V 、それらを結ぶ有向な枝の集合を E 、そして枝に流り得るフローの最大量を $u: E \rightarrow \mathbb{R}$ とする。以降、有向ネットワーク $N = (V, E, u)$ について考える。

定義 14.1 有向ネットワーク $N = (V, E, u)$ 、入口 (source) $s \in V$ 、出口 (sink) $t \in V$ が与えられているとする。各枝に流れるフロー (量) を関数 $\varphi: E \rightarrow \mathbb{R}$ で定義した時、入口 $s \in V$ から出る流量が最大となり、以下を満たすフロー (関数) を求める問題を**最大流問題**と呼ぶ。

容量制約 $0 \leq \varphi(i, j) \leq u(i, j), \quad \forall (i, j) \in E;$

流量保存則 $\sum_{(j,k) \in \delta_j^+} \varphi(j, k) - \sum_{(i,j) \in \delta_j^-} \varphi(i, j) = 0, \quad \forall j \in V \setminus \{s, t\}$

よって、容量制約と流量保存則を満たすフロー $\varphi(\cdot)$ を**実行可能流 (feasible flow)**と呼ぶ。特に、入口において最大の流量を実現する実行可能流を**最大流 (maximum flow)** $\varphi^*(\cdot)$ と呼ぶ。

定義 14.2 頂点の部分集合 $X \subseteq V$ で s を含み ($s \in X$)、 t を含まない ($t \notin X$) とき、

$$\delta(X) := \{(i, j) \in E \mid i \notin X, j \in X\} \cup \{(j, k) \in E \mid j \in X, k \notin X\}$$

を $s-t$ カット ($s-t$ cut) と呼ぶ。また、このカットの容量 $\kappa_u(X)$ を

$$\kappa_u(X) := \sum_{\{(j,k) \in E \mid j \in X, k \notin X\}} u(j, k)$$

と定義する。

容量最小の $s-t$ カットを**最小カット (minimum cut)**という。

定義 14.3 有向ネットワーク $N = (V, E, u)$ とこのネットワークに対する実行可能流 $\varphi(\cdot)$ が与えられているとする。この実行可能流に対する**補助ネットワーク (auxiliary network)** もしくは**残余ネットワーク (residual network)** は以下のように定義された有向ネットワーク $N_\varphi = (V, E_\varphi, u_\varphi)$ である。

$E_\varphi^F := \{(i, j) \in E \mid \varphi(i, j) < u(i, j)\}$, $E_\varphi^B := \{(j, i) \in E \mid \varphi(i, j) > 0\}$ とした時、 $E_\varphi := E_\varphi^F \cup E_\varphi^B$ 。そして $u_\varphi: E_\varphi \rightarrow \mathbb{R}$ は次のように定義された関数である。

$$u_\varphi(i, j) = \begin{cases} u(i, j) - \varphi(i, j), & (i, j) \in E_\varphi^F \\ \varphi(j, i), & (i, j) \in E_\varphi^B \end{cases}$$

補助ネットワーク上の s から t への有向道を**増加道 (augmenting path)**という。

図 15 に最大流問題の一例とある実行可能流に対する補助ネットワークを示す。

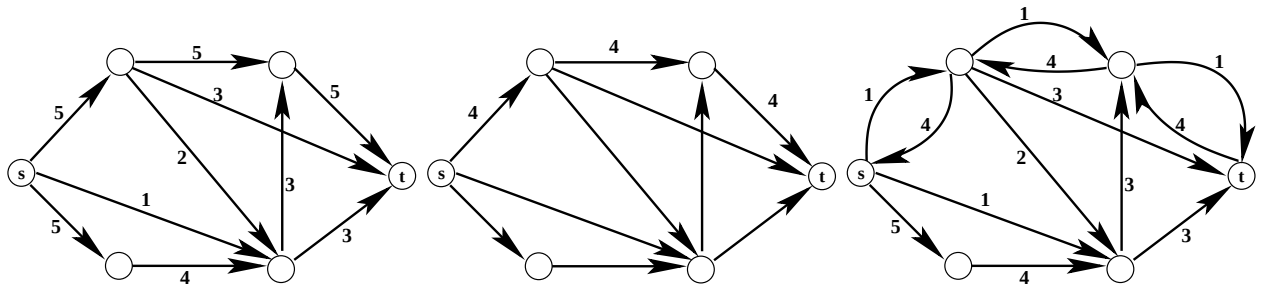


Figure 15: 左：最大流問題，中央：実行可能流，右：補助ネットワーク。

定理 14.1 (増加道定理) 実行可能流 $\varphi(\cdot)$ が最大流であるための必要十分条件は，補助ネットワーク N_φ に増加道が存在しないことである。

証明：（必要性）背理法を用いる．補助ネットワーク N_φ に増加道 $P \subseteq E$ が存在すると仮定する．すると $\varepsilon > 0$ に対して，フロー $\varphi'(\cdot)$ を

$$\varphi'(i, j) = \begin{cases} \varphi(i, j) + \varepsilon, & (i, j) \in P \\ \varphi(i, j) - \varepsilon, & (j, i) \in P \\ \varphi(i, j), & (i, j) \notin P, \quad (j, i) \notin P \end{cases}$$

と定義すると， $\varphi'(\cdot)$ は流量保存則を満たし，十分小さな正の値 $\varepsilon > 0$ に対して容量制約も満たす．さらに $\sum_{(s,k) \in \delta_s^+} \varphi'(s, k) = \sum_{(s,k) \in \delta_s^+} \varphi(s, k) + \varepsilon$ であるので， $\varphi(\cdot)$ が最大流であることに反する．

（十分性）補助ネットワーク N_φ に増加道が存在しないと仮定する．そして N_φ において $s \in V$ から到達可能な全ての頂点を $X \subseteq V$ と呼ぶ．明らかに $s \in X$ であるが，仮定より $t \notin X$ であるので， X は $s-t$ カットである．ここで $s \in V$ における正味流出量を考える．

$$\begin{aligned} \sum_{(s,k) \in \delta_s^+} \varphi(s, k) &= \sum_{(s,k) \in \delta_s^+} \varphi(s, k) + \sum_{j \in X \setminus \{s\}} \left[\sum_{(j,k) \in \delta_j^+} \varphi(j, k) - \sum_{(i,j) \in \delta_j^-} \varphi(i, j) \right] \\ &= \sum_{\{(j,k) \in E \mid j \in X, k \notin X\}} \varphi(j, k) - \sum_{\{(i,j) \in E \mid i \notin X, j \in X\}} \varphi(i, j) \\ &= \sum_{\{(j,k) \in E \mid j \in X, k \notin X\}} u(j, k) \\ &= \kappa_u(X) \end{aligned}$$

ただし，最初の等式は流量保存則より導かれ，2番目の等式は $i, j \in X$ であれば，その枝に流れるフローは打ち消されることによる．さらに，もし $(j, k) \in E$ で $j \in X, k \notin X$ であれば， $\varphi(j, k) = u(j, k)$ である．そして $(i, j) \in E$ で $i \notin X, j \in X$ であれば $\varphi(i, j) = 0$ であるので，3番目の等式が成り立つ．一般的には $\varphi(j, k) \leq u(j, k)$ ， $\varphi(i, j) \geq 0$ であるので， $\sum_{(s,k) \in \delta_s^+} \varphi(s, k) \leq \kappa_u(X)$ であるが，上記では等式が成り立つので， $\varphi(\cdot)$ は最大流である． ■

系 14.1 (最大流最小カット定理) 最大流量 $\sum_{(s,k) \in \delta_s^+} \varphi^*(s, k)$ は最小カット $\delta(X^*)$ のカット容量 $\kappa_u(X^*)$ に等しい．

14.2 フォード・フケルソン法 (Ford-Fulkerson Method)

フォード・フケルソン法

ステップ 1 : $\varphi(i, j) := 0, (i, j) \in E$ とする.

ステップ 2 : 補助ネットワーク $N_\varphi = (V, E_\varphi, u_\varphi)$ を作成し, (s から t までの) 増加道 P を見つける. 増加道が存在しなければ終了.

ステップ 3 : $\varepsilon = \min\{u_\varphi(i, j) \mid (i, j) \in P\}$ を求め, $\varphi(i, j) := \begin{cases} \varphi(i, j) + \varepsilon, & (i, j) \in P \\ \varphi(i, j) - \varepsilon, & (j, i) \in P \end{cases}$ と更新し, ステップ 1 へ.

各枝の容量 $u(\cdot)$ が整数値をとるときには, フォード・フケルソン法が有限回数で終了することがいえる. しかし, 実数値をとる場合は有限回数で終了しなかったり, 実行可能流が最大流に収束しなかったりもする.

14.3 演習問題

演習問題 14.1 図 16 で与えられたネットワークに対して最大流量と最大流を求めよ. ただし, 各枝に表示されている数字はその容量とする.

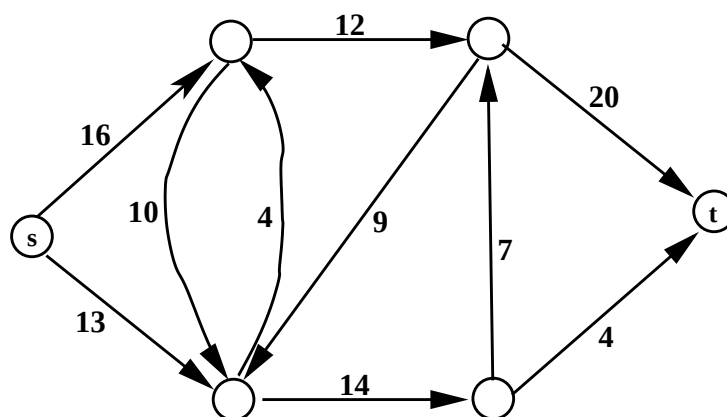


Figure 16: 最大流問題.