

2 グラフ (ネットワーク) ・ モデル

[用語集]

- ▷ 節点 (ノード node, 点, 結合点, 端点 vertex)
- ▷ 枝 (アーク arc, 辺, エッジ edge)
- ▷ ネットワーク network, グラフ graph
- ▷ 有向グラフ / 無向グラフ (directed graph, digraph / undirected graph)
- ▷ 路 (パス path, 経路, 道)

2.1 グラフ

グラフ (graph) とは, 幾つかの _____ の集合 V とそれらをつなぐ _____ の集合 E によって定義されるものであり, (V, E) と表す.¹ 枝に向きがある場合, _____ グラフといい, 向きがない場合, _____ グラフと言う. (平行な枝が無いとすると) 枝は節点の対で表すことが出来るため, 節点 i から節点 j への有向枝は (i, j) , 無向枝の場合は $\{i, j\}$ と書く. 例えば, 下図左は $V = \{ \text{_____} \}$, $E = \{ \text{_____} \}$ としたグラフ (V, E) であり, 下図右は $E = \{ \{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{3, 5\}, \{4, 5\} \}$ としたグラフ (V, E) である.² また, グラフに対して「枝の長さ」や「枝上の流れ」などを考えるとき, グラフをしばしばネットワークと呼ぶ. 実際, 通信網 (インターネットなど) や交通網 (道路交通, 鉄道網, 航空路線など) を抽象化したものでもあり, 現実的応用の多いモデリング・ツールである.

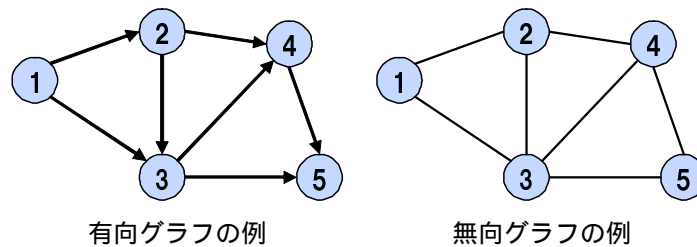


図 1: グラフは絵で描くことが出来る

2.2 グラフ (ネットワーク) によるモデリング Graph structure is ubiquitous!?

問題の構造や分析対象の関係を明確にする上でもグラフは便利である.³

- 電力, 水道, ガス, 電話, インターネット網
- 地理上の接続関係 (道路, 国境を接しているかどうかなど)
- 人間関係 (同盟関係), 食物連鎖, 企業間の取引関係 ([関連] 連鎖倒産とは何か?)
- 共著の有無 ([関連] Erdős number とは何か?), 論文の被引用関係, www のリンクの接続関係
- 時間と状態の記述, マルコフ過程の推移図, オートマトン, パーコレーション
- 神経回路網 ([関連] ニューラル・ネットワークとは何か?)
- 系統樹 (言語学の単語間の関係, 生物進化, 血縁関係)

¹節点はノード, 点, 結合点, 枝は辺, アーク, エッジなどとも呼ばれる.

²以下では, 特に断らない限り, 単に“グラフ”といえは, 有向グラフを指すものとする.

³増田による解説記事 [2] によれば, 社会に見られるネットワーク構造の持つ特性そのものを分析する「複雑ネットワーク」と呼ばれる研究分野が, 1998 年頃から発展してきているという.

warm-up それぞれの例において、何が節点（ノード）で何が枝となるだろうか？

Exercise 2.1 上記の例の中で、[関連]を付した質問について調べなさい。

2.3 グラフ（ネットワーク）上のよく知られた問題例

- [最短経路問題] カーナビ, 最短（最安）経路探索など.
- [最小費用流問題] 輸送問題の一般化.
- [最大流問題] ネットワーク上に最大どれだけのフローを流せるか.
- [PERT] 日程・工程の評価のための枠組み. 大規模建設工事における作業順序の策定など.
- [安定結婚問題（2部グラフのマッチング問題, 割当問題）] 与えられた2種類のグループ（例：男と女）の中から最も相性の良い組み合わせを求める. 研修医と研修先の組み合わせなど.
- [巡回セールスマン問題] 各枝 $e \in E$ に距離 d_e が与えられているグラフ (V, E) において, 全てのノードを丁度1度ずつ経由する巡回路で, アーク上の距離の合計（巡回路の長さ）を最小にするものを求める.
- [配送計画] どのような順番でどこに何を運ぶか, さらに何台どのような種類のトラックが必要か.

Exercise 2.2（割当問題） アイテムが n 個, 人が m 人いるとし, $n \leq m$ とする. i 番目のアイテムを j 番の人に割り当てたときの収入を w_{ij} 円とする. このとき, 総収入が最大となるように, n 個のアイテムを人に割り当てよ. ただし, 各アイテムは丁度1人の人に, 各人は最大1個のアイテムしか持てないとする.

この問題は割当問題として知られる問題である. まず, 問題の状況をグラフを用いて例示し, 割当問題を整数計画問題として定式化しなさい.

3 最短経路問題

グラフ $G = (V, E)$ の各枝 (i, j) が長さ a_{ij} を持つとき, ある節点 $s \in V$ から, 別の節点 $t \in V$ への路の中で, 最も長さの短いものをみつける問題を _____ という.

ここで, 節点 s から t への路（パス）とは, 節点の列 $P = (s, i, j, \dots, k, t)$ で, $(s, i) \in E, (i, j) \in E, \dots, (k, t) \in E$ を満たすものである. また, 特に始点と終点と同じ路を _____ という.

それらの枝の長さの和 $a_{si} + a_{ij} + \dots + a_{kt}$ を路 P の長さと言う. ある始点 s から終点 t を結ぶ路の内, 長さが最小のものを最短経路と言う.

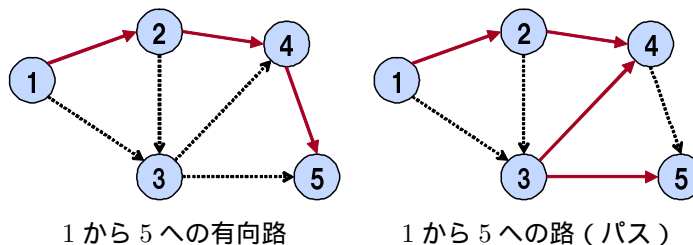


図 2: 路の例（今回は有向路（左図）のみを路として扱う）

3.1 ダイクストラ法

最短路問題を解く有名なアルゴリズムとしてダイクストラ法がある。

ダイクストラ法

[入力] グラフ $G = (V, E)$, 非負の枝の長さ $a: E \rightarrow \mathbb{R}_+$, 出発節点 $s \in V$

[出力] 節点 s から他の全節点 $i \in V \setminus \{s\}$ への最短路 $P_{s,i} = (s, \dots, i)$ とその距離 $d(i)$

Step 0. $S \leftarrow \phi, \bar{S} \leftarrow V, d(s) \leftarrow 0, d(i) \leftarrow \infty$ for $i \in V \setminus \{s\}$ とおく.

Step 1. $S = V$ ならば, 終了. さもなくば, $v \in \arg \min_i \{d(i) \mid i \in \bar{S}\}$ なる節点 v を選ぶ.

Step 2. $S \leftarrow S \cup \{v\}, \bar{S} \leftarrow \bar{S} \setminus \{v\}$ とし, $(v, j) \in E$ かつ $j \in \bar{S}$ であるようなすべての枝 (v, j) に対して

$$d(j) > d(v) + a_{vj} \text{ ならば, } d(j) \leftarrow d(v) + a_{vj}, p(j) \leftarrow v$$

とし, Step 1. に戻る.

3.2 ダイクストラ法の適用例

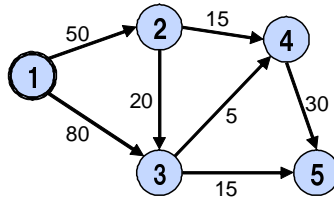


図 3: 重み付きグラフの例

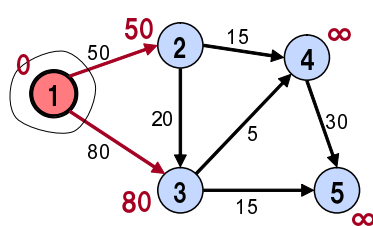
[反復 1] (0) $S = \phi, \bar{S} = \{\text{_____}\}, d(1) = 0, d(2) = d(3) = d(4) = d(5) = \infty$.

(1) $\min\{d(1), d(2), d(3), d(4), d(5)\} = \min\{0, \infty, \infty, \infty, \infty\}$ より, $v = \underline{\quad}$ となる.

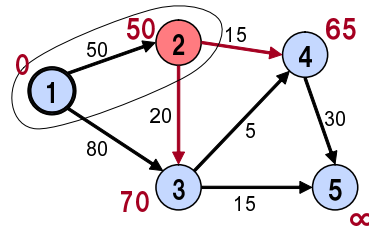
(2) $S = \{1\}, \bar{S} = \{2, 3, 4, 5\}$ であり, さらに $d(2) = \infty > d(1) + a_{12} = 0 + 50$ なので, $d(2) = 50, p(2) = 1$ となる. $d(3) = \infty > d(1) + a_{13} = 0 + 80$ なので, $d(3) = 80, p(3) = 1$ となる ($d(4) = \infty, d(5) = \infty$ は変化せず).

[反復 2] (1) $\min\{d(2), d(3), d(4), d(5)\} = \min\{\text{_____}\}$ より, $v = \underline{\quad}$ となる.

(2) $S = \{1, 2\}, \bar{S} = \{3, 4, 5\}$ なので, さらに, $d(3) = 80 > d(2) + a_{23} = 50 + 20$ であるから, _____ となり, $d(4) = \infty > d(2) + a_{24} = 50 + 15$ であるから, $d(4) = 65, p(4) = 2$ となる ($d(5) = \infty$ は変化せず).



反復 1 終了時



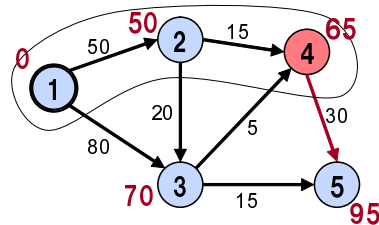
反復 2 終了時

[反復 3] (1) $\min\{d(3), d(4), d(5)\} = \min\{70, 65, \infty\}$ より, $v = 4$ となる.

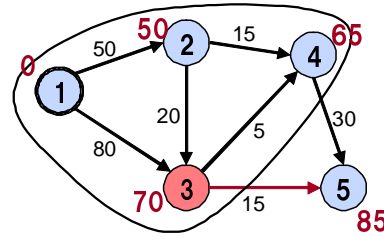
(2) $S = \{1, 2, 4\}$, $\bar{S} = \{3, 5\}$ なので, さらに, $d(5) = \infty > d(4) + a_{45} = 65 + 30$ であるから, $d(5) = 95$, $p(5) = 4$ となる ($d(3) = 70$ は変化せず).

[反復 4] (1) $\min\{d(3), d(5)\} = \min\{70, 95, \infty\}$ より, $v = 3$ となる.

(2) $S = \{1, 2, 3, 4\}$, $\bar{S} = \{5\}$ なので, さらに, $d(5) = 95 > d(3) + a_{35} = 70 + 15$ であるから, $d(5) = 85$, $p(5) = 3$ となる.



反復 3 終了時

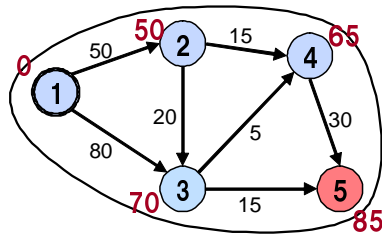


反復 4 終了時

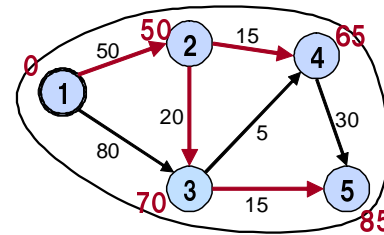
[反復 5] (1) $\bar{S} = \{5\}$ より, 自動的に $v = 5$ となる.

(2) $S = \{1, 2, 3, 4, 5\}$, $\bar{S} = \emptyset$ となる.

[反復 6] (1) $S = V$ であるから計算終了.



反復 5 終了時



反復 6 終了時

計算が終了したとき, $d(1) = 0$, $d(2) = 50$, $d(3) = 70$, $d(4) = 65$, $d(5) = 85$ となっているが, これらの値は節点 1 から各節点への最短路の長さを与えている. 上の計算から分かるように, このアルゴリズムでは出発点である節点 1 に近い節点から順に真の最短路の長さを確定し, 集合 S に含めていく. また, 最後に得られている $p(2) = 1$, $p(3) = 2$, $p(4) = 2$, $p(5) = 3$ によって定まる枝の集合 $\{(p(2), 2), (p(3), 3), (p(4), 4), (p(5), 5)\} = \{(1, 2), (2, 3), (2, 4), (3, 5)\}$ から, 各節点から節点 1 への最短路を見つけ出すことができる.

3.3 ダイクストラ法の性質

ここでは, ダイクストラ法のアルゴリズムにおいて, S が出発点 s からの最短路の長さが分かっている節点の集合となることを (数学的に) 確かめておく. その前に, まず最適性の原理と呼ばれる以下の基本的な原理について理解しておこう.

最適性の原理 いま仮に節点 s から t への最短路 P が得られているものとし, 路 P に含まれる節点を 1 つ任意に選ぶ. その節点を r とすれば, 路 P は節点 s から r までの部分と r から t への 2 つの部分に分割できる. 前半と後半の部分に対応する部分路をそれぞれ P_1 と P_2 とすれば, P_1 は節点 s から r への最短路であり, P_2 は節点 r から t への最短路になっているはずである.

これは何故だろうか? 実際, もし節点 s から r へ P_1 より短い路 P'_1 が別に存在するとすれば, 図 4 に示すように, 路 P'_1 と路 P_2 をつないだ路 $P'_1 \cup P_2$ は明らかに元の路 $P = P_1 \cup P_2$ より短い. これは節 s から t への最短路が P であることに反する. 節点 r から t への部分についても同様であるので, 上に述べた性質は常に成り立つことが分かる. 一般に, 節点 s から t までの最短路 P においては, そのどの部分を取り出しても, それがその両端の節点間を結ぶ最短路になっていることが分かる. このような性質をここでは最適性の原理と呼ぶ.

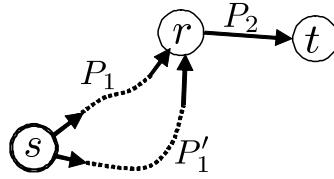


図 4: 最適性の原理

ダイクストラ法で最短路が求まることの証明 以下の主張を帰納法を用いて証明したい:

- 1 $i \in S$ ならば, $d(i)$ は s から i への最短路の長さである.
- 2 $i \in \bar{S}$ ならば, $d(i)$ は s から S に含まれる節点のみを経由して i に至る最短路の長さである.

証明 まず, アルゴリズムの最初の反復が終わった時点で $S = \{s\}$ かつ $d(s) = 0$ となっていることに注意する. 「すべての枝の長さが非負」の仮定より, このときは明らかに1と2が成立している.

次に, ある反復に入った時点で1と2が成り立っているととして, Step 1. で節点 v が選ばれたとし, 以下の2つを示せばよい:

- (a) $d(v)$ が s から v に至る最短路の長さである
- (b) この反復が終わった時点で, v を取り除いた集合 \bar{S} に対して2が成り立つ

まず, (a) を背理法により示す. s から v への最短路を P^* とし, その長さが $d(v)$ と等しくないと仮定する. アルゴリズムの構成法より, 各節点に対して $d(i)$ は s から i へのある路の長さを表していることに注意すれば, この仮定は

$$d(v) > [\text{最短路 } P^* \text{ の長さ}] \tag{1}$$

を意味する. ところで, $v \in \bar{S}$ なので, 帰納法の仮定2と式(1)から, s から v への真の最短路 P^* は途中で \bar{S} の節点を少なくとも1つ経由しているはず. そこで, 最短路 P^* において初めて現れる \bar{S} の節点を u とし, P^* を図5に示すように2つの部分経路 P_1^* と P_2^* に分解する. 最適性の原理より, 路 P_1^* は s から u への最短路であ

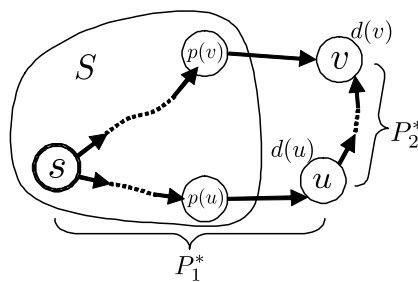


図 5: 背理法の仮定から, $u \in \bar{S}$ 経由の最短路が存在する

ることに注意. さらに, 路 P_1^* は途中で S の節点のみを経由していること, および帰納法の仮定2を考慮すると,

$$d(u) = [\text{最短部分路 } P_1^* \text{ の長さ}] \tag{2}$$

となっていることがわかる. また, 各枝の長さは非負なので

$$(\text{最短路 } P^* \text{ の長さ}) \geq (\text{最短部分路 } P_1^* \text{ の長さ})$$

であり, 仮定2と式(1)を合わせると,

$$d(v) > d(u)$$

である。ところが、 v の定め方 $d(v) = \min\{d(i) \mid i \in \bar{S}\}$ と $u \in \bar{S}$ より、

$$d(v) \leq d(u)$$

であるから、これは矛盾である。したがって、式 (1) は誤りであり、 $d(v)$ は s から v への最短路の長さに等しい。

また、帰納法の仮定[2]より、その路は S の節点のみを経由するため、結局、(a) が成り立つことが言える。

次に、(b) (すなわち、この反復が終了した時点で[2]が保たれていること) を示す。議論を明確にするため、反復の (Step 1.) 開始時点での集合 S に対して、この反復の (Step 2. の) 終了時点での集合を S^+ と書くことにする。すなわち、 $S^+ := S \cup \{v\}$ 。アルゴリズムの Step 2. を実行したとき

(b) $j \in \bar{S}^+$ ならば、 $d(j)$ は s から S に含まれる節点のみを経由して j に至る最短路の長さである

ことを示せば OK である。

s から S^+ の節点のみを経由して j に至る最短路には次の 3 つの場合が考えられる。

- (i) 節点 v を経由しない。すなわち S の節点のみを経由する。
- (ii) 節点 j に到達する直前に節点 v を経由する。
- (iii) 節点 v を経由し、その後 S の節点をいくつか通って j に到達する。

しかし、実際には (i) と (ii) の場合だけを考えれば十分である。なぜなら、もし、(iii) のような路 P が最短路になったとして、その最短路 P における j の直前の節点を k とすれば、最適性の原理より、路 P の k までの部分 P_1 は s から k までの最短路である。また、 k はこれより前の反復において S の要素となっているので、(1) より、図 6 のように S の節点だけを経由して s から k に至る最短路 P'_1 が存在するはずである。明らかに、2 つの最短

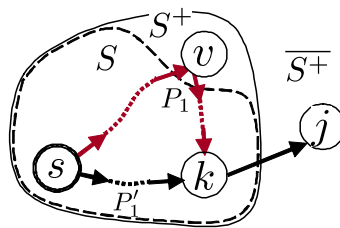


図 6: ケース (iii) は冗長

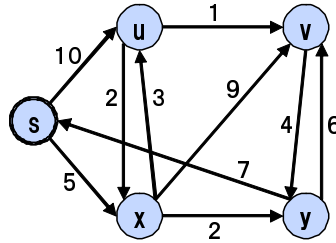
路 P_1 と P'_1 の長さは等しいので、 s から j への最短路として、 P の代わりに P_1 を P'_1 で置き換えた (v を経由しない) 路を考えても構わない。したがって、 s から S^+ の節点のみを経由して j に至る最短路は、(i) と (ii) の 2 つの場合に限定できる。すなわち、その 2 つの場合を比較して、「 $d(j) > d(v) + a_{vj}$ ならば $d(j)$ を $d(v) + a_{vj}$ で置き換え、そうでなければ $d(j)$ の値は変更しない」というアルゴリズムの Step 2. の手続きは (b) を保つ。

以上の議論から、常に[1]と[2]が満たされるので、すべての節点が集合 S に含まれた時点で節点 s から全節点への最短路が得られることが分かる。また、各反復が終了するたびに、どれか 1 つの節点が集合 \bar{S} から取り除かれ、集合 S に入ることになる。□

Exercise 2.3 次のネットワークにおいて節点 s からすべての節点への最短路をダイクストラ法を用いて求めよ。

Exercise 2.4 以下の藤澤先生 (中央大学経営システム工学科) の web サイトではアメリカ合衆国の任意の 2 地点間の最短路をダイクストラ法で求めることができる。下記 URL にアクセスし最短路問題を実行せよ。

<http://opt.indsys.chuo-u.ac.jp/portal/>



4 最小費用流問題

4.1 ネットワーク・フロー, 最小費用フロー

前節では、各枝の長さ（費用）のみが与えられたネットワークを取り扱った。この節ではさらに節点に供給（または需要）が与えられたネットワーク $G = (V, E)$ を考え、そのネットワーク上を流れる流れ（フロー）の量を最適化する。

- ▷ b_i : 節点 i における物資（製品）の供給量（負の値のときは需要量）, $i \in V$
- ▷ c_{ij} : 枝 (i, j) を物資 1 単位流すのにかかる費用（コスト）
- ▷ u_{ij} : 枝 (i, j) に流すことのできる物資の上限数
- ▷ x_{ij} : 枝 (i, j) を流れる（流す）物資の量（フロー）[決定変数]

このとき、各節点の需要と供給を満たしつつ、コストを最小にする問題は次のような線形計画問題（LP）として表すことができる：

$$\begin{array}{l}
 \text{最小化} \quad \sum_{(i,j) \in E} c_{ij} x_{ij} \\
 \text{条件} \quad \sum_{j:(i,j) \in E} x_{ij} - \sum_{j:(j,i) \in E} x_{ji} = b_i \quad i \in V \\
 \quad \quad \quad 0 \leq x_{ij} \leq u_{ij} \quad (i, j) \in E
 \end{array}$$

1 番目の制約条件は各節点 j について、純流出量（= 流出量 - 流入量）が純供給量 b_j に等しいことを要請するもので、これを満たすフロー x を _____ と言う。最後の非負制約はフローの向きを枝の向きに限るための制約で、これを満たす _____ を _____ と言う。

この問題は幾つかの生産地から幾つかの消費地へ最小費用で物資を輸送する問題でもあり、輸送問題とも呼ばれる。

最小費用流問題の一例を図 7 に示す。図 7 のネットワークにおいて、各節点の横にある数字はその節点からネットワークへのフローの供給量を表している。負の供給量はその節点においてフローが吸収されることを意味しているので、その節点におけるフローの需要量を表すと考えることができる。すなわち、図 7 では、節点 1, 2 が供給節点（ソース）で、節点 4 が需要節点（シンク）である。一方、需要・供給量が 0 であるような節点（図 7 では節点 3）は通過節点と呼ばれる。図 7 のネットワークにおいて、各枝の横の 2 つの数字はフロー 1 単位当たりの費用（コスト）および容量の値を示している。例えば、枝 (3, 4) の単位フロー当たりのコストは 5、容量は 15 である。

以下では、ネットワーク全体で需要・供給の過不足が存在しないことを意味する条件

を仮定する。明らかに、この条件が成立しないときには、上の問題の制約条件を満たすフローは存在しない。

最小費用流問題に対しては、これまで数多くの方法が提案されている。特に、最小費用流問題が特別な構造を持った線形計画問題であることから、そのネットワーク構造を巧みに利用することにより、単体法を非常に効率的に実行できる。そのような方法は _____ と呼ばれ、現在のところ実用的にはもっとも有効な方法

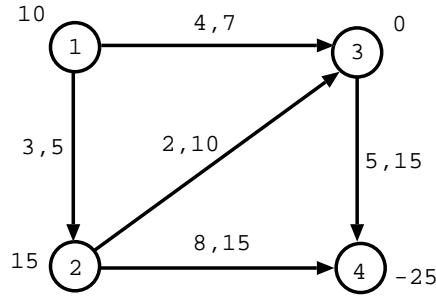


図 7: 最小費用流問題の例 (節点の横の数字は需要・供給量, 枝の横の数字は単位コストと容量)

とされている. これに対し, 単体法とは異なるグラフ論的な考え方に基づいた効率的なアルゴリズムも数多く提案されている. ここでは, 負閉路除去法と呼ばれる方法を紹介する.

4.2 負閉路除去法

ネットワーク $G = (V, E)$ において, あるフロー x が与えられたとき, $G^x = (V, E^x)$ を次のように構成する. まず, ネットワーク $G = (V, E)$ において容量 u_{ij} とコスト c_{ij} をもつ枝 (i, j) を $u_{ij}^x = u_{ij} - x_{ij}$ を持つ枝 (i, j) と, 残余容量 $u_{ji}^x = x_{ij}$ を持つ枝 (j, i) で置き換える. 更に, 前者の枝 (i, j) のコストを $c_{ij}^x = c_{ij}$, 後者の枝 (j, i) のコストを $c_{ji}^x = -c_{ij}$ とする. ただし, $u_{ij}^x = 0$ のときは枝 (j, i) のみを, $u_{ji}^x = 0$ のときは枝 (i, j) のみを考える.

図 7 のネットワークにおいて, 図 8 に示すフローが与えられたとき, そのフローに対する残余ネットワークは図 9 にのようになる.

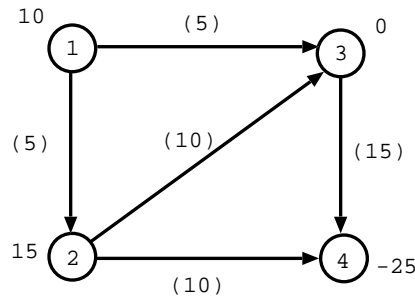


図 8: 図 7 のネットワークにおけるフローの例 (括弧内数字はフロー)

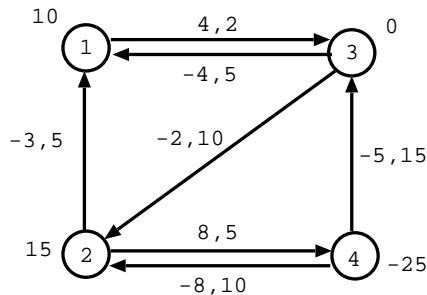


図 9: 図 8 のフローに対する残余ネットワーク (数字は枝のコストと容量)

残余ネットワークにおいて, ある節点から出発して枝の向きに沿って, 次々に節点を經由し, 最後にもとの

節点に戻る有向閉路を考えたとき、その閉路上の枝のコスト c_{ij}^x の和をその閉路のコストと呼ぶ。例えば、図 9 の残余ネットワークにおいては、有向閉路 $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$ のコストは $4 + (-2) + (-3) = -1$ で、有向閉路 $2 \rightarrow 4 \rightarrow 3 \rightarrow 2$ のコストは $8 + (-5) + (-2) = 1$ である。残余ネットワークには普通多数の有向閉路が存在するが、それらの中で、コストが負であるようなものを特に _____ と呼ぶ。上の例では有向閉路 $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$ は負閉路である。

さて、いま残余ネットワーク $G^x = (V, E^x)$ において、負閉路が 1 つ見つかったとする。そのとき、その負閉路に沿ってフローを追加すれば、流れ保存則を保ったまま、コストの総和を現在の値から減少させることができる。例えば、図 9 の残余ネットワークにおける負閉路 $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$ に沿って、 e だけフローを追加することを考える。これは、もとのネットワーク G において、枝 $(1, 3), (2, 3), (1, 2)$ のフローをそれぞれ $x_{13} = 5 + e$, $x_{23} = \underline{\hspace{2cm}}$, $x_{12} = \underline{\hspace{2cm}}$ に変更することを意味する。このとき、明らかに節点 1, 2, 3 における流れ保存則は保たれる。さらに、フローの増加量 e が十分小さければ、枝 $(1, 3), (2, 3), (1, 2)$ において容量制約条件が破ぶられることもない。それ以外の節点や枝での流れは不変であるから、 e が十分小さいとき、変更されたフローは実行可能フローで、コストの総和はフロー変更前の

$$3x_{12} + 4x_{13} + 2x_{23} + 8x_{24} + 5x_{34} = 3 \cdot 5 + 4 \cdot 5 + 2 \cdot 10 + 8 \cdot 10 + 5 \cdot 15 = 210$$

から

$$3(5 - e) + 4(5 + e) + 2(10 - e) + 8 \cdot 10 + 5 \cdot 15 = 210 - e$$

に減少する。また、負閉路に沿って追加できるフローの最大値 e_{\max} を知るには、その閉路上の枝の残余容量の最小値を求めればよい。上の例では、

$$e_{\max} = \min\{u_{13}^x, u_{32}^x, u_{21}^x\} = \min\{\underline{\hspace{2cm}}\} = \underline{\hspace{2cm}}.$$

となるので、負閉路に沿って $e_{\max} = \underline{\hspace{2cm}}$ だけフローを追加できる。その結果、 $x = (x_{12}, x_{13}, x_{23}, x_{24}, x_{34})^\top = (3, 7, 8, 10, 15)^\top$ となり、それに対するコストの総和は _____ となる。

このように残余ネットワークに負閉路が存在する限り、その閉路に沿ってフローを追加することにより、コストの総和を減少させることができる。また、残余ネットワークが負閉路を 1 つも含まないならば、そのとき得られているフローは最小費用流問題の最適解になっていることが示せる。上の例において、フロー $x = (x_{12}, x_{13}, x_{23}, x_{24}, x_{34})^\top = (3, 7, 8, 10, 15)^\top$ に対する残余ネットワークを構成すると、容易に確かめられるように、このネットワークには負閉路が存在しない。このことから、現在のフロー $x = (3, 7, 8, 10, 15)^\top$ は最適であることが分かる。

上の例からも分かるように、ある負閉路に沿ってフロー e_{\max} を追加すると、その負閉路が残余ネットワークから消滅する。すなわち、この操作は 1 つの負閉路を残余ネットワークから取り除くことを意味する。このような考え方に基づいて、フローを順次修正していくことにより、最終的に最適フローを求めようとする方法が負閉路除去法である。負閉路除去法の一般的な計算手順は次のように書くことができる。

負閉路除去法

Step 0. 適当な初期フロー x を定める。

Step 1. 残余ネットワーク $G^x = (V, E^x)$ において負閉路を見つける。負閉路が存在しなければ計算終了。

Step 2. 負閉路に沿って可能な限りフローを追加し、新しいフロー x を得る。Step 1. に戻る。

上にも述べたように、負閉路除去法は、各節点での流れ保存則と各枝の容量制約条件を満たしつつ、ネットワーク全体の総費用を減少させる操作を反復する方法である。各枝のコストと容量が _____ ならば、1 回の反復において、総費用は少なくとも 1 単位減少することが保証される。ところで、 $C = \max\{|c_{ij}| \mid (i, j) \in E\}$, $U = \max\{u_{ij} \mid (i, j) \in E\}$ とすれば、負閉路除去法の初期フローに対する総費用は $|E|CU$ を超えることはない。また、そのようにネットワークにおける総費用の最小値は $-|E|CU$ 以下になることはない。したがって、負閉路除去法は最大でも $2|E|CU$ 回以下の反復で最適フローを見出すことが分かる。

Exercise 2.5 図7のネットワークにおける最小費用流を単体法で求めよ.

Exercise 2.6 図10のネットワークにおいてコストと容量が与えられている. 最小費用流を負閉路除去法で求めよ.⁴

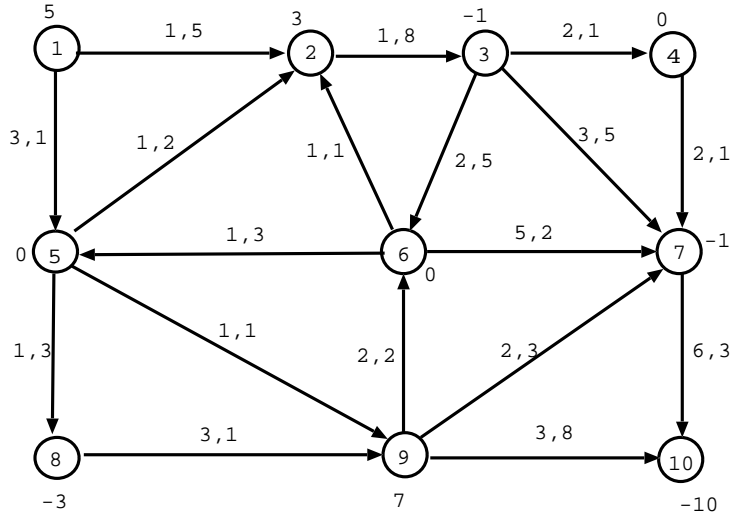


図 10: 節点の横の数字は需要・供給量, 枝の横の数字はコストと容量

5 最大流問題

この節では各枝に, _____ が与えられたネットワーク $G := (V, E)$ において _____ と呼ばれる節点 s から _____ と呼ばれる節点 t に向かってできるだけ多くのものを送る最大流問題を考える.

ここで, ネットワーク G の枝 (i, j) 上の流れの大きさを x_{ij} , 枝 (i, j) の容量, すなわち流れ x_{ij} の上限値を u_{ij} とし, ソースからシンクへの総流量を f と表せば, 問題は次のような線形計画問題として表すことができる:

最大化	f
条件	$\sum_{j:(s,j) \in E} x_{sj} - \sum_{j:(j,s) \in E} x_{js} = f$

	$0 \leq x_{ij} \leq u_{ij} \quad (i, j) \in E$

2 番目の制約条件は 1 つの節点について, 流入量と流出量が等しいことを要請するもので, _____ 則という. 最後の制約 $0 \leq x_{ij} \leq u_{ij}$ は容量制約とも呼ばれる. _____ 則と容量制約を満たす $x = (x_{ij})$ を (実行可能) フローと言う.

LP なので, 単体法を用いることでも解くことが出来る. ここではネットワークの特性を使ったアプローチを説明する.

図 11 の最大流問題の例ではソースは節点 1, シンクは節点 5 であり, 枝に付して書かれた数字は各枝 (i, j) の容量 u_{ij} を示している.

⁴ [ヒント] たとえば, $x_{1,2} = 5, x_{2,3} = 8, x_{3,6} = 3, x_{3,7} = 4, x_{6,5} = 3, x_{5,8} = 3, x_{7,10} = 3, x_{9,10} = 7$, その他の $x_{ij} = 0$ は実行可能フローである.

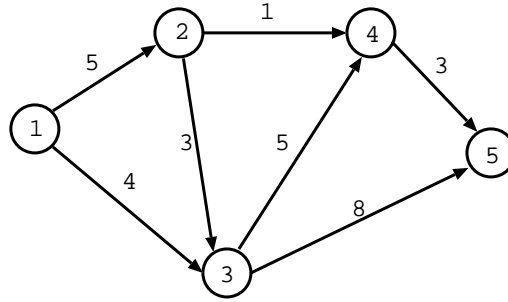


図 11: 最大流問題の例 (数字は枝の容量)

以下に述べるアルゴリズムは、適当な初期フローからはじめて、順次フローを改良していく方法である。ここでは、前節で最小費用流を求める際に導入した残余容量と残余ネットワークの概念を再び導入する。

冗長であるが、ここで再度説明しておこう。以下では簡単のため、もとのネットワーク $G = (V, E)$ においては、任意の 2 節点 $i, j \in V$ の間に枝 (i, j) と (j, i) の両方が存在することはないと仮定する。

あるフロー $x = (x_{ij})$ が与えられているものとしよう。このとき、ネットワーク $G = (V, E)$ の各枝 $(i, j) \in E$ を容量 $u_{ij}^x = u_{ij} - x_{ij}$ をもつ枝と容量 $u_{ji}^x = x_{ij}$ をもつ枝 (j, i) で置き換える (図 12)。ただし、 $u_{ij}^x = 0$ のと

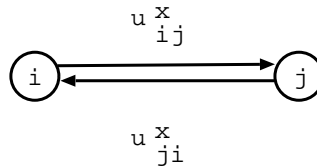


図 12: 残余容量

きは、枝 (j, i) のみを考え、逆に、 $u_{ji}^x = 0$ のときは枝 (i, j) のみを考えるものとする。このように定められた u_{ij}^x 、及び u_{ji}^x をフロー x に対する枝 (i, j) の _____ と言う。さらに、ネットワーク $G = (V, E)$ の各枝 (i, j) を容量 u_{ij}^x を持つ枝 (i, j) と容量 u_{ji}^x を持つ枝 (j, i) で置き換えたネットワークをフロー x に対する残余ネットワークといい、 $G^x := (V, E^x)$ と書くことにする。図 11 のネットワークにおいて図 13 に示すフローが与えられ

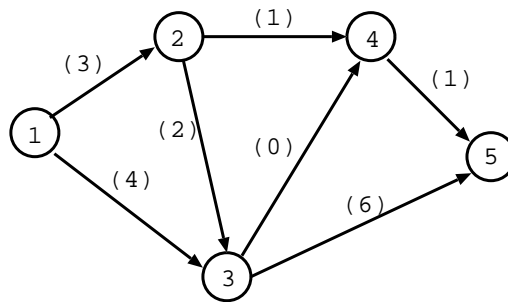


図 13: 図 6 のネットワークにおけるフローの例 (括弧内の数字はフロー)

たとき、そのフローに対する残余ネットワークは図 14 のようになる。たとえば、ネットワーク $G = (V, E)$ の枝 $(1, 2)$ に注目すると、容量が $u_{12} = 5$ のところに $x_{12} = 3$ だけのフローが流れているので、残余ネットワーク $G^x = (V, E^x)$ においては、容量 $u_{12}^x = u_{12} - x_{12} = 2$ の枝 $(1, 2)$ と容量 $u_{21}^x = x_{12} = 3$ の枝 $(2, 1)$ が現れている。これは、現在のフロー x_{12} に対して、節点 1 から 2 に向かって 2 単位のフローを、また節点 2 から節点 1 に向かって 3 単位のフローを増加しても容量制約条件は破れないことを意味している。

さらに、残余ネットワーク $G^x = (V, E^x)$ において、ソース s からシンク t への路が存在すれば、その路に沿っ

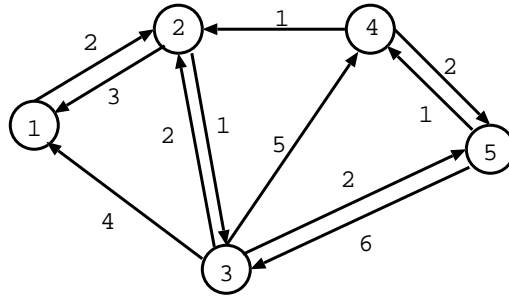


図 14: 図 8 のフローに対する残余ネットワーク (数字は残余容量)

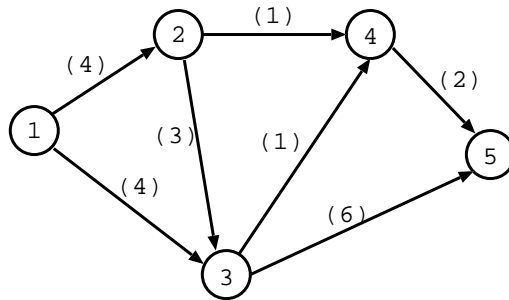


図 15: フロー増加路による流量の増加 (括弧内の数字はフロー)

てフローを追加することによって、もとのネットワーク G 上での流量 f が増加できることが分かる。このことから、残余ネットワークにおけるソースからシンクへの路を (現在のフロー x) に対するフロー増加路と呼ぶ。また、容易に分かるように、フロー増加路に沿って追加できるフローの量は、その路に含まれる枝の残余容量の最小値に等しい。例えば、図 14 の残余ネットワークには、ソース $s = 1$ からシンク $t = 5$ へ $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$ や $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ といった路が存在し、それらのフロー増加路に沿って追加できるフローの量はそれぞれ $\min\{u_{12}^x, u_{23}^x, u_{35}^x\} = \min\{2, 1, 2\} = 1$ および $\min\{\text{_____}\} = \min\{\text{_____}\} = \text{_____}$ となる。これらのフロー増加路のどちらを使っても流量を増加することができるが、たとえば後者の路 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ に沿って 1 単位の流れを実際に図 13 のフローに追加すると、図 15 に示すフローが得られる。

このように、フロー増加路を順次見つけながら、その路に沿ってフローを追加していけば、最終的に最大流が得られることが期待される。このフロー増加法の手順は次のようにまとめられる。

フロー増加法

[入力] グラフ $G = (V, E)$, 非負の枝の容量 $u : E \rightarrow \mathbb{R}_+$, ソース $s \in V$, シンク $t \in V$

[出力] ソース s からシンク t への最大フロー x (および最大フロー量 f)

Step 0. 適当な初期フロー x を定める。(たとえば、すべての $(i, j) \in E$ に対して $x_{ij} = 0$ とする.)

Step 1. 残余ネットワーク $G^x = (V, E^x)$ において、ソース s からシンク t への路 (フロー増加路) を見つける。フロー増加路が存在しなければ、計算を終了する。

Step 2. フロー増加路に沿って可能な限りフローを追加し、新しいフロー x を得る。Step 1. に戻る。

このアルゴリズムの各反復では、残余ネットワークにおけるソース s からシンク t への路と、その路に対するフロー増加可能量を求める必要がある。これは、ソースから到達可能な節点を順次ラベルと呼ばれる印を付けていくことにより、シンクへの路を構成するラベリング法の手続きを用いて実行できる。この方法では、ソースから到達可能であることが判明している節点 (ラベル付けられた節点) の集合 L と、その部分集合で、その節

点からさらに先まで到達可能かどうかを調べ終わった節点 (走査済の節点と言う) の集合 S を常に保持しながら計算を進めていく。その結果, 集合 L がシンク t を含んだ時点で, ソースからシンクへのフロー増加路が得られたことになる。また, 集合 L が t を含む前に, L の要素がすでに走査済となり, S の要素になってしまえば, s から t へのフロー増加法は存在しないと言える。

ラベリング法

Step 0. $L := \{s\}, S := \emptyset$ とする。すべての節点 $i \in V$ に対して $p(i) := 0$ とする。

Step 1. $t \in L$ または $L = S$ ならば終了。そうでなければ, 節点 $i \in L \setminus S$ を一つ選び, $S := S \cup \{i\}$ とする。

Step 2. 残余ネットワークにおける節点 i を始点とする枝 (i, j) (すなわち $u_{ij}^x > 0$ であるような枝) すべてに対して,

$$j \notin L \text{ ならば } L := L \cup \{j\}, p(j) := i$$

とし, Step 1. に戻る。

この手続きはフロー増加法の Step 1. を実行するために用いられる。終了条件 $t \in L$ が満たされたとき, ソース s からシンク t へのフロー増加路が得られるので, これをもってフロー増加法の反復に戻る。一方, 終了条件 $L = S$ が満たされたときは, ソース s からシンク t へのフロー増加路が存在しないので, フロー増加法の計算も終了する。また, アルゴリズムに現れる $p(j)$ は, 最短路問題のダイクストラ法における $p(j)$ と同様, シンク t からソース s へ逆向きにフロー増加路を辿るために用いられる。

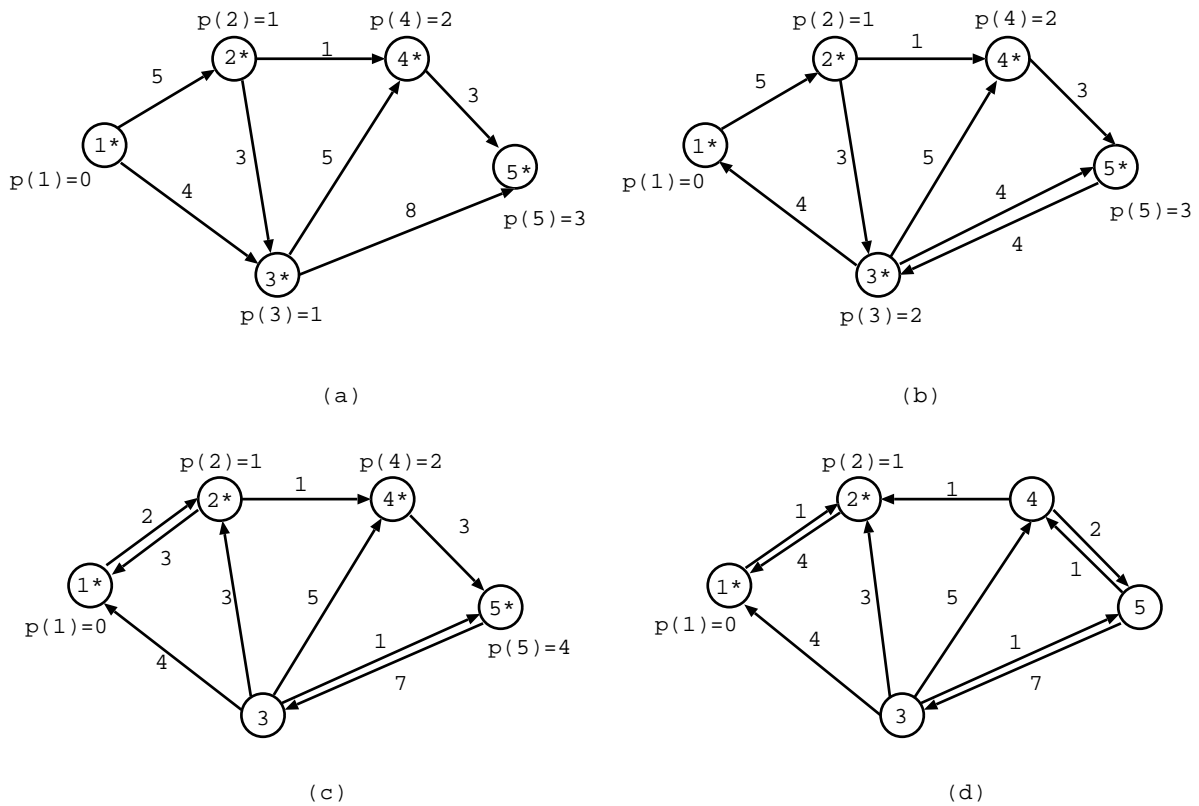


図 16: ラベリング法によるフロー増加法の計算例 (*はラベル付けられた節点)

ラベリング法を用いたフロー増加法を図 11 に適用した結果を示す。図 16(a) には初期のフローを $x = 0$ としたときの残余ネットワーク (もとのネットワークに一致) とそれに対するラベリング法の結果を示している。

図 16(b) は得られたフロー増加路 $1 \rightarrow 3 \rightarrow 5$ に沿って 4 単位のフローを流した後の残余ネットワークに対してラベリング法を適用した結果を表している. その結果, フロー増加路 $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$ に沿って ____ 単位のフローが追加できるので, 総流量は 7 単位となる. また, 得られた残余ネットワークに対して, ラベリング法を適用すると図 16(c) のようになり, 新たなフロー増加路 $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ を得る. この路に対するフロー増加量は 1 単位なので, 総流量は 8 単位となり, 図 16(d) の残余ネットワークが得られる. さらにラベリングを行うと, 図 16(d) のように節点 1 と 2 だけにラベルが付いた段階で, それらがすべて走査済となるので, フロー増加路が存在しないことが分かり, 計算が終了する. 最後に得られたフローは $x_{12} = 4, x_{13} = 4, x_{23} = 3, x_{24} = 1, x_{34} = 0, x_{35} = 7, x_{45} = 1$ となり, その流量は ____ 単位である.

5.1 ラベリング法の正当性と最大流最小カット定理

枝の容量 u_{ij} がすべて整数ならば, 初期フローを整数値 (例えば, すべての $(i, j) \in E$ に対して $x_{ij} = 0$) としてフロー増加法を用いたとき, ソース s からシンク t への流量は 1 回の反復で少なくとも 1 単位は増加するので, その反復回数は明らかに ____ である. さらに, 以下では, ラベリング法を用いたフロー増加法の計算が終了した時点で得られているフローが実際に最大流になっていることを示す.

節点集合 V をソース s を含む集合 S とシンク t を含む集合 T に分割したとき, $\{(i, j) \in E : i \in S, j \in T\}$ なる枝の集合をカットと呼び, (S, T) と表す. また, 任意のカット (S, T) に対して, S の節点を始点とする枝を $(i, j) \in (S, T)$, 逆に T の節点を始点し, S の節点を終点とする枝を $(j, i) \in (T, S)$ と書く. さらに, (S, T) に含まれるすべての枝の容量の合計をカット (S, T) の _____ と呼び, $C(S, T)$ で表す. すなわち,

$$C(S, T) := \sum_{(i,j) \in (S,T)} u_{ij}. \tag{3}$$

Exercise 2.7 図 11 のネットワーク G において, $S = \{1, 2, 4\}, T = \{3, 5\}$ のときのカット容量 $C(S, T)$ はいくらか?

いま, x を任意のフロー, f をその流量とし, (S, T) を任意のカットとする. このとき, $s \in S$ かつ $t \in T$ より, 明らかに

$$f = \sum_{(i,j) \in (S,T)} x_{ij} - \sum_{(j,i) \in (T,S)} x_{ji} \tag{4}$$

が成り立つが, すべての枝 $(i, j) \in E$ に対して $0 \leq x_{ij} \leq u_{ij}$ であるから, カット容量の定義式 (3) より, 不等式

$$f \leq C(S, T) \tag{5}$$

を得る.

さて, ラベリング法を用いるフローの増加法の計算が終了した時点で得られているフローを x^* , その流量を f^* とし, ラベル付けされかつ走査済となっている節点の集合を S^* , その補集合 $V \setminus S^*$ を T^* と書くことにしよう. ラベリング法の終了条件より明らかに, $s \in S^*$ かつ $t \in T^*$ なので, (S^*, T^*) はカットである. また, ラベリング法の構成法と残余ネットワークの定義より,

$$(i, j) \in (S^*, T^*) \implies x_{ij}^* = u_{ij} \tag{6}$$

$$(j, i) \in (T^*, S^*) \implies x_{ji}^* = 0 \tag{7}$$

が成り立っているはずである. なぜなら, もしも $x_{ij}^* < u_{ij}$ であるような枝 $(i, j) \in (S^*, T^*)$ が存在するか, あるいは $x_{ji}^* > 0$ であるような枝 $(j, i) \in (T^*, S^*)$ が存在すれば, 残余ネットワーク $G^{x^*} = (V, E^{x^*})$ において $i \in S^*$ かつ $j \in T^*$ であるような枝 $(i, j) \in E^{x^*}$ が存在するので, ラベリング法が終了するという仮定に矛盾するからである. 従って, (4) 式と (6) 式, (7) 式により,

$$f^* = \sum_{(i,j) \in (S^*, T^*)} x_{ij}^* - \sum_{(j,i) \in (T^*, S^*)} x_{ji}^* = \sum_{(i,j) \in (S^*, T^*)} u_{ij} = C(S^*, T^*)$$

が成り立つ. 式 (5) が常に成り立つことを考慮すると, これは x^* が実行可能なすべてのフローの中で最大流量を与えるものであり, 同時に (S^*, T^*) がすべてのカットの中で最小の容量をもつものであることを示している. これより, フロー増加法が終了したときに得られているフローは最大流になっていることが確かめられた. このフローの最大流量とカット容量の最小値が等しいという事実は最大流・最小カット定理と呼ばれている.

最大流・最小カット定理 全ての最大流問題は以下の状態のいずれか 1 つである:

- (i) フロー量を任意に大きくでき, しかも任意のカットが無限に大きなカット容量を持つ.
- (ii) 最大フローが存在し, その最大フロー量が最小なカット容量と一致する.

Exercise 2.8 ([5]) 以下の表はある時点における各都市間を結ぶ飛行機 (路線) の残り座席数を示している. このとき, サンフランシスコ (SF) からニューヨーク (NYC) まで最大で何名まで (追加で) 運ぶことができるか?

出発地	行き先	残り座席数
サンフランシスコ (SF)	デンバー (D)	5
サンフランシスコ (SF)	ヒューストン (H)	6
デンバー (D)	アトランタ (A)	4
デンバー (D)	シカゴ (C)	2
ヒューストン (H)	アトランタ (A)	5
アトランタ (A)	ニューヨーク (NYC)	7
シカゴ (C)	ニューヨーク (NYC)	4

- (1) 都市を節点, 各路線を枝として, ネットワークを図で表しなさい.
- (2) この問題を最大流問題として捉え, 最大フローをフロー増加法+ラベリング法により解きなさい.

参考文献

- [1] 久保・松井 (1999) 『組合せ最適化 [短編集]』 (朝倉書店)
- [2] 増田 (2008) 複雑ネットワークの研究動向について 『オペレーションズ・リサーチ - 経営の科学 -』 vol.53 no.9 (日本オペレーションズ・リサーチ学会) pp.511-516.
- [3] 森・松井 (2004) 『オペレーションズ・リサーチ』 (朝倉書店)
- [4] 東京工業大学情報科学科 「計画数学第一」 講義資料.
- [5] V. Chvátal (1983), *Linear programming*, H.W.Freeman and Company, New York.