数理計画法 第7回目資料 — ネットワーク計画

塩浦 昭義*

平成14年11月13日

参考文献 — 福島雅夫著「数理計画入門」, システム制御情報ライブラリー15, 朝倉書店 (1996)

1　ネットワーク計画とは？

★ **グラフ** — **頂点** (節点, 点) が**枝** (辺, 弧, 線) で結ばれたもの (詳しくは講義「情報数学」を復習)
★ **ネットワーク** — グラフに数値データ (距離, 移動時間, コスト, etc.) が付加されたもの

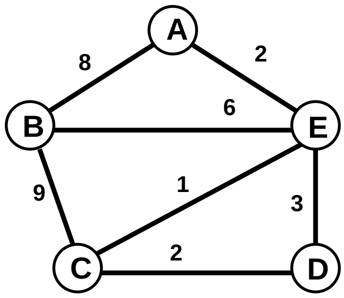


図1: 無向グラフと関連するネットワーク

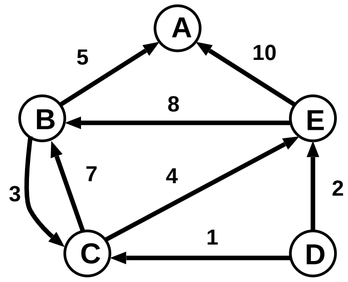


図2: 有向グラフと関連するネットワーク

★ **ネットワーク計画問題** — ネットワークに関する数理計画問題

○ **最小木問題**

例: 大学内での安価なLANの構築 (図1参照)
- 頂点 → コンピュータ
- 枝 → 設置可能なLANケーブル
- 枝の数値 → LANケーブルの費用 (長さ)

最小化	設置するLANケーブルの総費用
条件	LANケーブルの設置後, すべてのコンピュータ間で通信可能

○ **最短路問題**

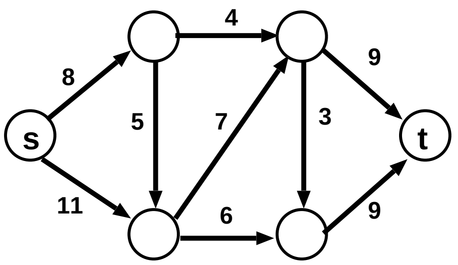
例: 仙台駅から気仙沼駅までのJRでの最短経路は? (図2参照)
- 頂点 D → 仙台駅
- 頂点 A → 気仙沼駅
- その他の頂点 → 乗り換え駅
- 枝 → JRの路線, 数値は駅間距離

最小化	仙台駅から気仙沼までの総距離
条件	枝の集合は仙台駅から気仙沼までのルートに対応

※ 上記の問題は講義「アルゴリズムとデータ構造」で学習済み

*東北大学大学院 情報科学研究科 shioura@dais.is.tohoku.ac.jp

○ **最大フロー問題** — 例: 工場から小売店へ商品をたくさん配送したい



頂点 S → 工場
頂点 T → 小売店
その他の頂点 → 配送センター
枝 → 配送可能な区間
枝の数値 → 配送可能な商品の最大数

図 3: 最大フロー問題の例.

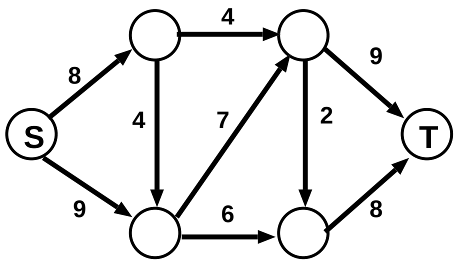


図 4: 配送パターンの例. 総配総量は 17 である.

○ **最小費用フロー問題**— 例: 工場から小売店へ商品をなるべく安く配送したい

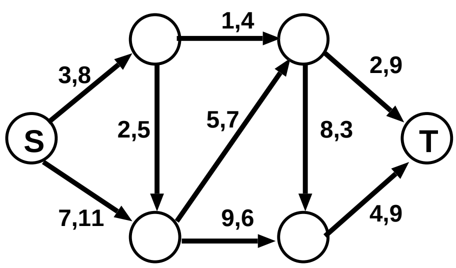


図 5: 最小費用フロー問題の例.
左側の数値は商品一つ運ぶのに必要な費用, 右側は配送可能な商品の最大数.
なお, 小売店での需要量は 13 とする.

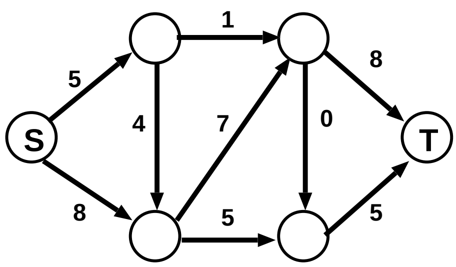


図 6: 配送パターンの例. 総費用は 196 である.

◎ **授業の内容**　最大フロー問題, 最小費用フロー問題に対する

- 定式化
- 最適性条件
- アルゴリズム

2　最大フロー問題

2.1　定式化

入力:　有向グラフ $G = (V, E)$, $s \in V$ (**供給点**), $t \in V$ (**需要点**)
　　　各枝 $(i, j) \in E$ の**容量** $u_{ij} (\geq 0)$

$$
\left\|
\begin{array}{lll}
\text{最大化} & f & \\
\text{条件} & \displaystyle\sum_{j:(s,j)\in E} x_{sj} - \sum_{i:(i,s)\in E} x_{is} = f & \cdots \quad (1) \\
& \displaystyle\sum_{j:(t,j)\in E} x_{tj} - \sum_{i:(i,t)\in E} x_{it} = -f & \cdots \quad (2) \\
& \displaystyle\sum_{j:(k,j)\in E} x_{kj} - \sum_{i:(i,k)\in E} x_{ik} = 0 \qquad (k \in V \setminus \{s, t\}) & \cdots \quad (3) \\
& 0 \leq x_{ij} \leq u_{ij} \quad ((i, j) \in E) & \cdots \quad (4)
\end{array}
\right.
$$

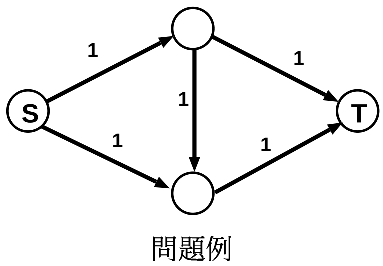
$(x_{ij} \mid (i, j) \in E)$ は**フロー** $\overset{\text{定義}}{\Longleftrightarrow}$ $(x_{ij} \mid (i, j) \in E)$ は上記のＬＰの許容解
f — フロー $(x_{ij} \mid (i, j) \in E)$ の**流量**

★ 解説

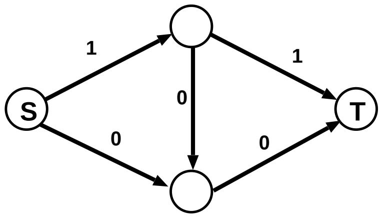
変数 x_{ij}	枝 $(i, j) \in E$ を流れるフロー量
変数 f	供給点 s での供給量 (= 需要点 t での需要量)
制約式 (1)	(s から出る枝のフロー量の総和) − (s に入る枝のフロー量の総和) = (供給点 s での供給量)
制約式 (2)	(t から出る枝のフロー量の総和) − (t に入る枝のフロー量の総和) = − (需要点 t での需要量)
制約式 (3)	(頂点 k から出る枝のフロー量の総和) − (k に入る枝のフロー量の総和) = 0　　(**流量保存則**)
制約式 (4)	各枝を流れるフローは非負かつ容量を越えてはならない (**容量制約**)

※ 最大フロー問題は線形計画問題の特殊ケース $\Longrightarrow$ 単体法で解くことができる!

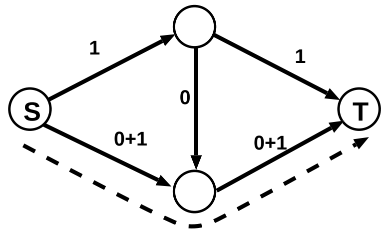
2.2　残余ネットワークとフロー増加法



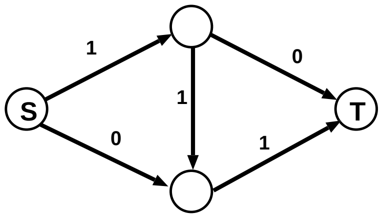
問題例



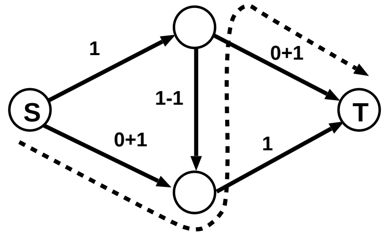
最大フロー?



最大フローではない!



最大フロー?



最大フローではない!
注意：フローは減らすことも可能

どうやって最大フローであることを判定する? — **残余ネットワーク**の利用

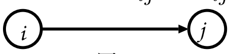
★ 残余ネットワークの構築法

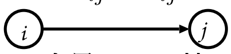
$\boldsymbol{x}$: 現在のフロー $\Longrightarrow$ x に関する残余ネットワーク $G^x = (V, E^x)$, $E^x = F^x \cup R^x$

$F^x = \{(i,j) \mid (i,j) \in E, x_{ij} < u_{ij}\}$, F^x の各枝 (i,j) の容量 $u^x_{ij} = u_{ij} - x_{ij}$

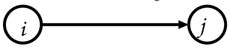
$R^x = \{(j,i) \mid (i,j) \in E, x_{ij} > 0\}$, R^x の各枝 (j,i) の容量 $u^x_{ji} = x_{ij}$

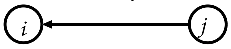
フロー量 $x_{ij} < u_{ij}$ $\Longrightarrow$ 同じ向き, 容量 $u_{ij} - x_{ij}$ の枝を加える



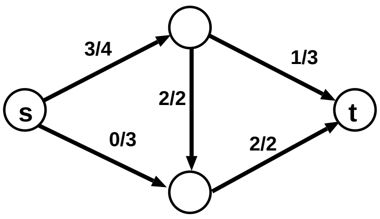


フロー量 $x_{ij} > 0$ $\Longrightarrow$ 逆の向き, 容量 x_{ij} の枝を加える

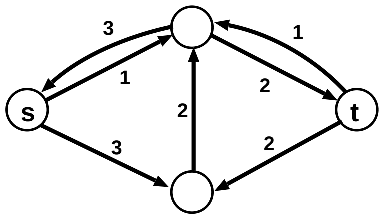




残余ネットワークの例



$\Longrightarrow$



問題例とフロー (各枝の数値: (フロー量)/(容量))

残余ネットワーク (各枝の数値: 容量)

残余ネットワークにおいて s から t への路が存在 $\Longrightarrow$ 現在のフロー $\boldsymbol{x}$ の流量を増加させることが可能

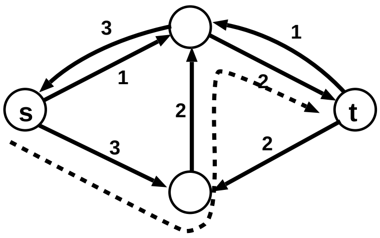
★ フローの更新方法

$\boldsymbol{x}$: 現在のフロー, $P \subseteq E^x$: 残余ネットワーク上の s から t への路

$\alpha = \min\{u^x_{ij} \mid (i,j) \in P\}$ (路 P 上の枝の容量の最小値)

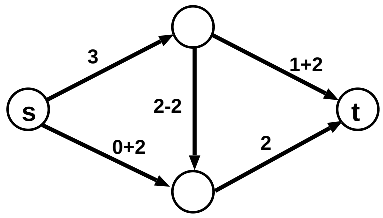
$$\text{元のグラフの各枝 } (i,j) \in E \text{ に対し, } \begin{cases} (i,j) \in P \cap F^x & \Longrightarrow \quad x_{ij} := x_{ij} + \alpha \\ (i,j) \in P \cap R^x & \Longrightarrow \quad x_{ij} := x_{ij} - \alpha \\ \text{それ以外} & \Longrightarrow \quad x_{ij} \text{ は不変} \end{cases}$$

$\Longrightarrow$ フロー $\boldsymbol{x}$ の流量が α 増加



s から t への路が存在, $\alpha = 2$

$\Longrightarrow$



更新後のフロー量. フローの流量は 3 から 5 へ増加

以上のアイデアをもとに最大フローを求めるアルゴリズムを構築 $\Longrightarrow$ フロー増加法

★ フロー増加法

ステップ 0: 各枝 $(i,j) \in E$ に対して $x_{ij} = 0$ とおく.
ステップ 1: $\boldsymbol{x}$ に関する残余ネットワーク $G^x = (V, E^x)$ を構築する.
ステップ 2: G^x における s から t への路 P を求める.
存在しなければ現在のフロー $\boldsymbol{x}$ は最大フローなので終了.
ステップ 3: ステップ 2 で求めた路 P を用いてフロー $\boldsymbol{x}$ を更新.
ステップ 4: ステップ 1 へ戻る.

2.3 フロー増加法の正当性と最大フロー最小カット定理

フロー増加法は本当に最大フローを求める？— カットという概念を用いて証明

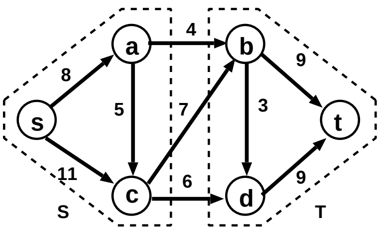
★ **カット** (S,T) $\overset{\text{定義}}{\Longleftrightarrow}$ S と T は頂点集合 V の分割 $(S \cup T = V,\ S \cap T = \emptyset)$, $s \in S$, $t \in T$

★ 各枝 (i,j) に対し, $(i,j) \in (S,T) \overset{\text{定義}}{\Longleftrightarrow} i \in S,\ j \in T$

$(i,j) \in (T,S) \overset{\text{定義}}{\Longleftrightarrow} j \in S,\ i \in T$

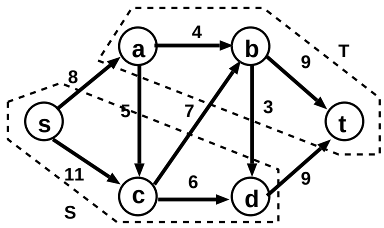
★ カット (S,T) の**容量** $C(S,T) = \displaystyle\sum_{(i,j)\in(S,T)} u_{ij}$

★ **最小カット** $\overset{\text{定義}}{\Longleftrightarrow}$ 容量が最小のカット



カットの例 1: 容量 17

$(S,T) = \{(a,b),(c,b),(c,d)\}$, $(T,S) = \emptyset$



カットの例 2: 容量 24

$(S,T) = \{(s,a),(c,b),(d,t)\}$, $(T,S) = \{(a,c),(b,d)\}$

性質 1: 任意のフローの流量 f と 任意のカット (S,T) に対し, $f \leq C(S,T)$ が成り立つ.

証明: 最大フロー問題の制約 (流量保存則) のうち, $i \in S$ に関するものを足しあわせる. $i,j \in S$ のとき, 変数 x_{ij} の項は頂点 i に関する制約の左辺において係数が $+1$, 頂点 j に関する制約の左辺において係数が -1, 他の制約の左辺において係数が 0 となるので, 和をとると消えてしまう. その結果, 次の式を得る:

$$f = \sum_{k\in S}\left(\sum_{j:(k,j)\in E} x_{kj} - \sum_{i:(i,k)\in E} x_{ik}\right) = \sum_{(i,j)\in(S,T)} x_{ij} - \sum_{(j,i)\in(T,S)} x_{ji}.$$

カットの容量の定義およびフロー量の非負性より, 上記の式の右辺は $C(S,T)$ 以下である. □

※ 性質 1 は LP の弱双対定理に対応する.

★ フロー増加法の正当性の証明:

フロー増加法の終了後, 残余ネットワーク上で s から路により到達できる頂点集合を S とおく

- $s \in S$, アルゴリズムの終了条件より $t \notin S \Longrightarrow (S,T)$ はカット
- S の定義より, 残余ネットワークにおいて S から T に向かう枝は存在しない

$\Longrightarrow$ 元のグラフ G において, S から T に向かう枝 (i,j) のフロー量 $x_{ij} = u_{ij}$

T から S に向かう枝 (j,i) のフロー量 $x_{ji} = 0$

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

$\Longrightarrow$ 性質 1 より f は最大フロー, カット (S,T) は最小カット.

□

上記の証明より, 以下の定理が成り立つ.

定理 2 [最大フロー最小カット定理]: 最大フローの流量とカットの容量の最小値は一致する.

※ 定理 2 は LP の双対定理に対応する.

2.4 フロー増加法の反復回数とアルゴリズムの改良

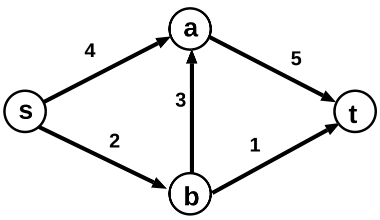
n : 頂点数, m : 枝数, $U = \max\{u_{ij} \mid (i,j) \in E\}$
※ 以下, 枝の容量は整数値と仮定

★フロー増加法の反復回数
- フロー増加法は各反復で流量を少なくとも 1 増加させる
- 最大フローの流量は mU 以下

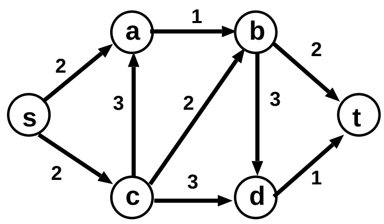
$\Longrightarrow$ 反復回数は高々 mU — 入力のビット長 $n, m, \log U$ に関する多項式ではない!

★ 反復回数を減らすには — ステップ 2 で「特別な」路を求める
- その 1: s から t への (枝数に関する) 最短路 $\Longrightarrow$ 反復回数は高々 $mn/2$
- その 2: s から t への路のうち, 容量 α が最大の路 $\Longrightarrow$ 反復回数は $\mathrm{O}(m \log nU)$
- その他, 様々な多項式時間アルゴリズムが存在

レポート問題 下記のネットワークに対する最大フローおよび最小カットを求めよ.



問題 1



問題 2