数理計画法　第5回

線形計画問題
ネットワーク計画

ネットワーク計画問題とは？
最大フロー問題

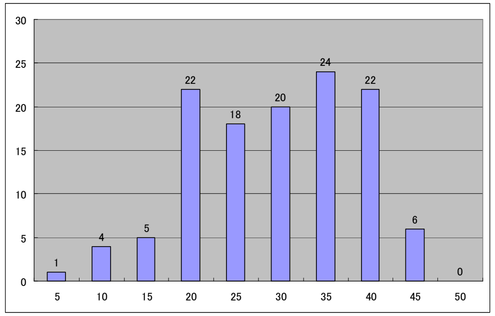
担当：　塩浦昭義
(情報科学研究科　准教授)
shioura@dais.is.tohoku.ac.jp

中間試験の結果について

平均点：27．5点

合格者(25点以上）　76人，不合格者（24点未満）　46人

不合格者は
追試レポートを
提出すること（必須）

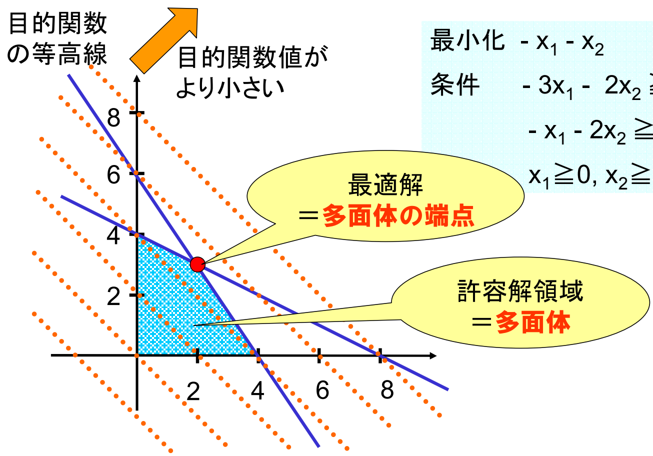


中間試験の追試レポートについて

中間試験の結果が24点以下の学生は
追試レポート を提出してください．

- **レポートの課題：** 中間試験のすべての問題を解くこと
  レポート用紙は授業HPにあるファイルを印刷する
- **締切:１２月６日(木) 13:10まで**
- レポート未提出の場合，またレポートの出来が80％以下の場合には単位は**不可**
- レポートを提出した場合でも，中間試験の得点の変更はしません

LPの幾何的なイメージ



最小化　$-x_1 - x_2$
条件　$-3x_1 - 2x_2 \geqq -12$
$-x_1 - 2x_2 \geqq -8$
$x_1 \geqq 0,\ x_2 \geqq 0$

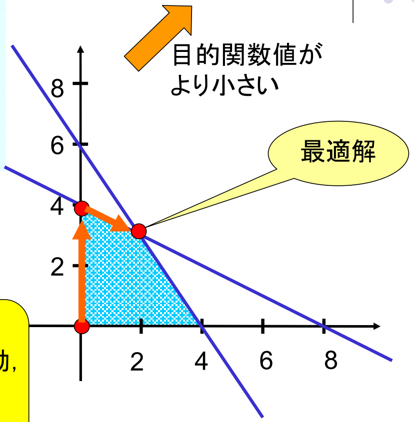
単体法の幾何的なイメージ

最小化 $-x_1 - x_2$

条件 $-3x_1 - 2x_2 \geqq -12$

$-x_1 - 2x_2 \geqq -8$

$x_1 \geqq 0,\ x_2 \geqq 0$



- 原点からスタート
- 隣り合う端点に繰り返し移動，目的関数値を減らす
- 最適解に到達

基底解の幾何的なイメージ

最小化 $-x_1 - x_2$

条件 $-3x_1 - 2x_2 \geqq -12$

$-x_1 - 2x_2 \geqq -8$

$x_1 \geqq 0,\ x_2 \geqq 0$

$z = -4 + (1/3)x_3 - (1/3)x_2$

$x_1 = 4 - (1/3)x_3 - (2/3)x_2$

$x_4 = 4 + (1/3)x_3 - (4/3)x_2$

基底解(**4, 0**, 0, 4)

許容解

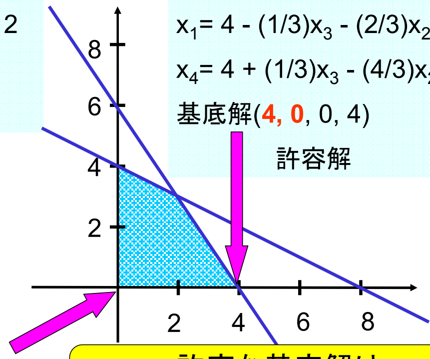
$z = \ \ - x_1 - x_2$

$x_3 = 12 - 3x_1 - 2x_2$

$x_4 = \ 8 - x_1 - 2x_2$

基底解(**0, 0**, 12, 8)

許容解



許容な基底解は
許容解領域の端点に対応する

基底解の幾何的なイメージ

最小化 $-x_1 - x_2$

条件 $-3x_1 - 2x_2 \geqq -12$

$-x_1 - 2x_2 \geqq -8$

$x_1 \geqq 0,\ x_2 \geqq 0$

$z = -6 - (5/2)x_1 - (1/2)x_3$

$x_2 = 6 - (3/2)x_1 - (1/2)x_3$

$x_4 = -4 + 2x_1 + x_3$

基底解(**0, 6**, 0, -4)

許容解でない

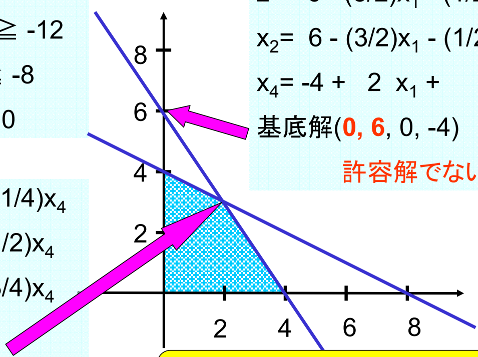
$z = -5 + (3/4)x_3 + (1/4)x_4$

$x_1 = 2 - (1/2)x_3 - (1/2)x_4$

$x_2 = 3 - (1/4)x_3 - (3/4)x_4$

基底解(**2, 3**, 0, 0)

許容解



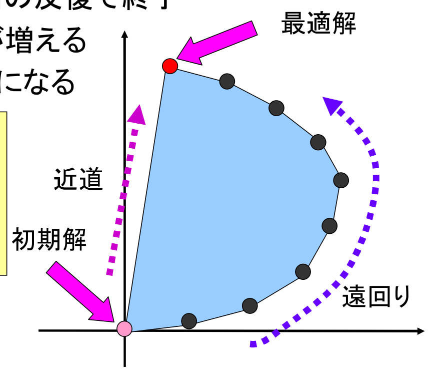
基底解は
制約の「交点」に対応する

単体法の計算時間について

- 基底解の更新の仕方に依存して反復回数が変わる
- 普通の場合，多項式回の反復で終了
- 運が悪いと反復回数が増える
- 最悪O(2^n)回（指数回）になる

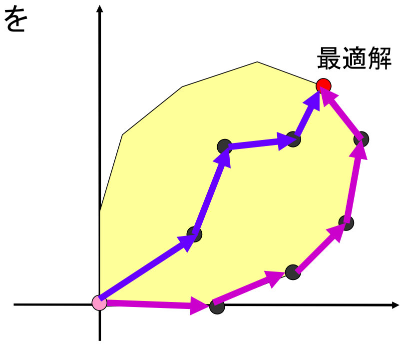
反復回数を小さく
（多項式回）に
するような
ピボット演算の規則は
あるか？

長年の
未解決問題



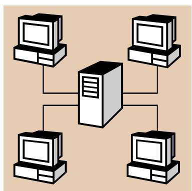
内点法

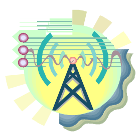
- 単体法は許容解領域の端を通って最適解にたどり着く➔遠回りになる可能性大
- 内点法：許容解領域の中を通って最適解を求める
- 理論的に速いこと：多項式回の反復で終了
- 実用上も速い



LPソルバーについて

- 単体法および内点法を用いたLPのソルバー（LPを解くソフトウェア）が入手可能
- フリーのソルバーもあり：lp solve, GLPK,など
- Excel にもLPソルバー（アドイン）がある
- フリーのソルバーに比べ，商用のソルバーは大規模なLPを高速により正確に解くことができる
- C++などのライブラリとしてプログラム中で呼び出して使うことも可能

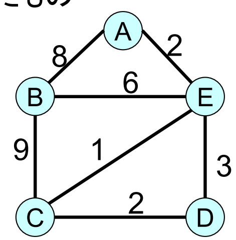




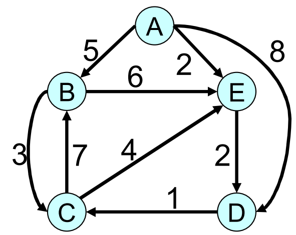
ネットワーク計画問題とは？

グラフとネットワーク

★（無向、有向）グラフ
頂点（接点、点）が枝（辺、弧，線）で結ばれたもの
★ネットワーク
頂点や枝に数値データ（距離、コストなど）が付加されたもの



無向グラフ



有向グラフ

ネットワーク計画問題

★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

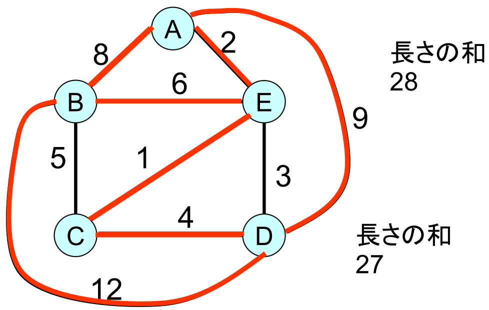
例： 最小木問題 / 最短路問題 } 他の講義で扱う「アルゴリズムとデータ構造」「情報数学」

最大フロー問題 / 最小費用フロー問題 } この授業で扱う

巡回セールスマン問題 } 問題のみ紹介

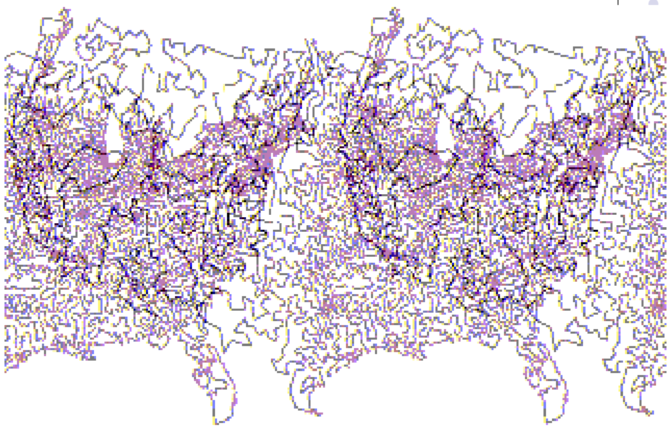
巡回セールスマン問題

枝の長さの和が最小の巡回路を求める



具体例： 運搬経路問題，VLSI設計，基盤配線
ドリルでの穴あけ，など

巡回セールスマン問題



巡回セールスマン問題

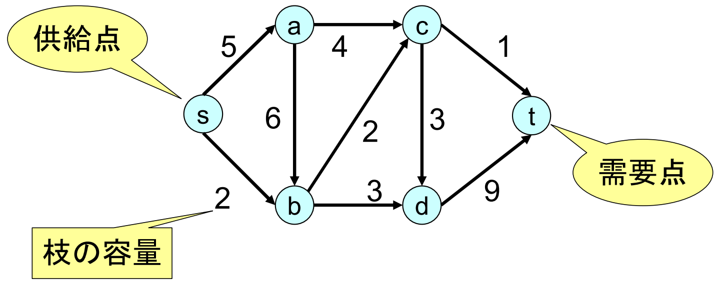
最大フロー問題

最大フロー問題の定義(その1)

入力:有向グラフ G = (V, E)

供給点 $s \in V$, 需要点 $t \in V$

各枝 $(i, j) \in V$ の容量 $u_{ij} \geqq 0$



最大フロー問題の定義(その2)

目的:供給点から需要点に、

枝と頂点を経由して「もの」をたくさん流したい

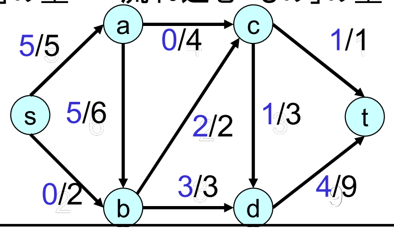
条件1(容量条件):

0 ≦ 各枝を流れる「もの」の量 ≦ 枝の容量

条件2(流用保存条件):

頂点から流れ出す「もの」の量＝ 流れ込む「もの」の量

与えられたネットワークと解の一例



最大フロー問題の定式化(その1)

変数 x_{ij}: フロー＝枝 (i, j) を流れる「もの」の量

変数 f: フロー量＝需要点に流れ込む「もの」の量

(＝ 供給点から流れ出す「もの」の量)

目的:供給点から需要点に「もの」をたくさん流したい

⇒ 最大化 f

容量条件:0 ≦ 各枝を流れる「もの」の量 ≦ 枝の容量

⇒ $0 \leqq x_{ij} \leqq u_{ij}$ $((i,j) \in E)$

最大フロー問題の定式化(その2)

流用保存条件:

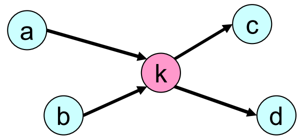
(頂点から流れ出す「もの」の量)

一 (流れ込む「もの」の量) = 0

⇒ $\Sigma\{x_{kj}$ | 枝 (k,j) は 頂点 k から出る$\}$

一 $\Sigma\{x_{ik}$ | 枝 (i,k) は 頂点 k に入る$\}$ = 0

$(k \in V - \{s, t\})$



$(x_{kc} + x_{kd}) - (x_{ak} + x_{bk}) = 0$

供給点と需要点に関する条件:

$\Sigma\{x_{sj}$ | (s,j) は s から出る$\}$ - $\Sigma\{x_{is}$ | (i,s) は s に入る$\}$ = f

$\Sigma\{x_{tj}$ | (t,j) は t から出る$\}$ - $\Sigma\{x_{it}$ | (i,t) は t に入る$\}$ = - f

最大フロー問題の定式化(その3)

定式化をまとめると

最大化 f

条件 $0 \leqq x_{ij} \leqq u_{ij}$ $((i,j) \in E)$

$\Sigma\{x_{kj}$ | (k,j) は k から出る$\}$

- $\Sigma\{x_{ik}$ | (i,k) は k に入る$\}$ = 0 $(k \in V - \{s, t\})$

$\Sigma\{x_{sj}$ | (s,j) は s から出る$\}$

- $\Sigma\{x_{is}$ | (i,s) は s に入る$\}$ = f

$\Sigma\{x_{tj}$ | (t,j) は t から出る$\}$

- $\Sigma\{x_{it}$ | (i,t) は t に入る$\}$ = - f

この問題の許容解 x_{ij} 一 フロー

フローの目的関数値 f 一 フロー値

最大フロー問題の解法

最大フロー問題は線形計画問題の特殊ケース

⇒ 単体法で解くことが可能！

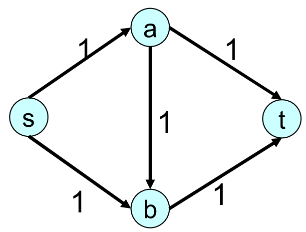
最大フロー問題は良い離散構造をもつ

⇒ この問題専用の解法(フロー増加法など)

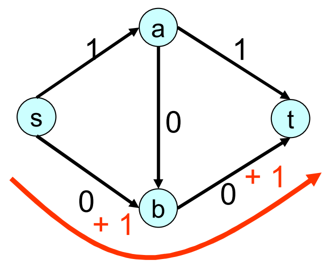
により、より簡単に, より高速に解くことが可能

最大フローの判定

問題の例



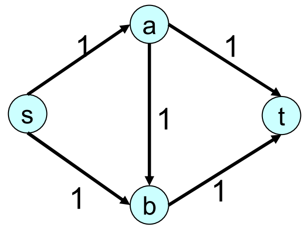
フローの例1:最大？



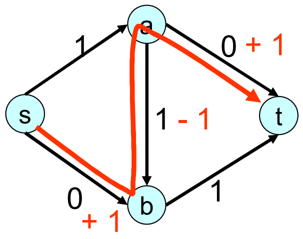
最大フローではない

最大フローの判定

問題の例



フローの例2：最大？



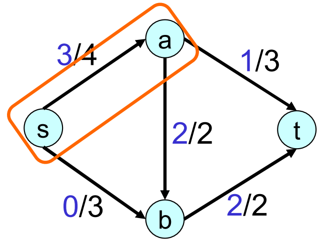
最大フローではない

最大フローであることの判定を効率よく行うには？

⇒ 残余ネットワークを利用

残余ネットワークの定義

残余ネットワークの作り方



問題例とフロー

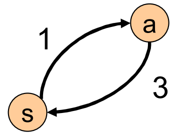
各枝のデータは

（フロー量/容量）

枝(s,a)において

☆さらに4－3＝1だけフローを流せる

⇒ 残余ネットワークに容量1の枝(s,a)を加える

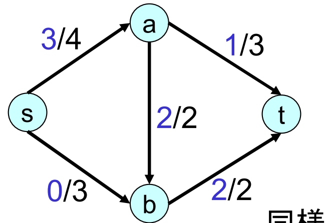


☆現在のフロー3を逆流させて0にすることが出来る

⇒ 容量3の枝(a,s)を加える

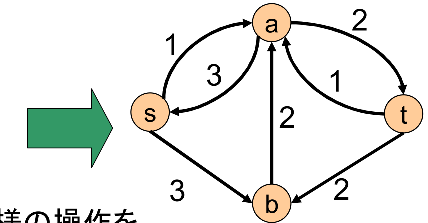
残余ネットワークの定義

残余ネットワークの作り方



問題例とフロー

同様の操作を各枝に行う



残余ネットワークの完成

残余ネットワークの定義（まとめ）

$x = (x_{ij} \mid (i,j) \in E)$: 現在のフロー

フロー x に関する残余ネットワーク $G^x = (V, E^x)$

$E^x = F^x \cup R^x$

順向きの枝集合

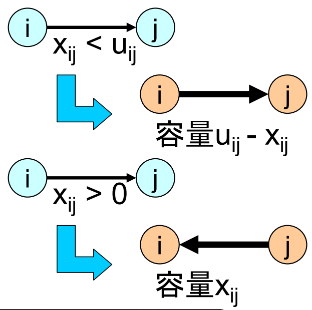
$F^x = \{ (i, j) \mid (i, j) \in E, x_{ij} < u_{ij} \}$

各枝の容量 $u^x_{ij} = u_{ij} - x_{ij}$

逆向きの枝集合

$R^x = \{ (j, i) \mid (i, j) \in E, x_{ij} > 0 \}$

各枝の容量 $u^x_{ji} = x_{ij}$



注意！：現在のフローが変わると残余ネットワークも変わる

残余ネットワークに関する定理

定理１：残余ネットワークに s-t パスが存在する
➔ 現在のフローは増加可能

定理２：残余ネットワークに s-t パスが存在しない
➔ 現在のフローは最大フロー

証明は次回

フロー増加法

最大フローを求めるためのアルゴリズム

ステップ0：初期フローとして、全ての枝のフロー量を
0とする
ステップ1：現在のフローに関する残余ネットワークを作る
ステップ2：残余ネットワークに s-t パスが存在しない
⇒ 終了
ステップ3：残余ネットワークの s-t パスをひとつ求め、
それを用いて現在のフローを更新する
ステップ4：ステップ1へ戻る

レポート問題

問1：

(a) 下記のLPの許容解領域を図示しなさい.

(b) 下記のLPを単体法で解きなさい.
また，各反復で現れる基底解の動きを(a)の図を使って説明しなさい.

最小化 $-x_1 - x_2$

条件 $-2x_1 + x_2 \geqq -2$

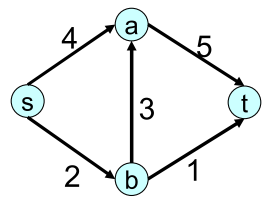
$x_1 - x_2 \geqq -1$

$x_1 \geqq 0,\ x_2 \geqq 0$

レポート問題

問2：次の2つの最大フロー問題に対して、
(a) 定式化せよ

問題(1)



問題(2)