数理計画法 第5回

塩浦昭義
情報科学研究科 准教授

shioura@dais.is.tohoku.ac.jp
http://www.dais.is.tohoku.ac.jp/~shioura/teaching

中間試験について

- 日時：12月8日（木）13:00〜14:30（予定）
- 12/1までにレポートを一度も出していない場合，受験不可
- 教科書，ノート等の持ち込みは一切不可
- 座席はこちらで指定
- 試験内容：11/24（第6回目）までの講義で教えたところ
  - 様々な数理計画モデル
  - 線形計画問題の標準形，単体法，各種定理
  - 組合せ計画問題の分枝限定法
- 50点満点，29点以下は追試レポートもしくは単位不可
- 今後の予定
  - 11/24：第6回，12/1：第7回，12/8：中間試験

前回の復習

§2.6 双対問題と弱双対定理

最適値の見積もりの計算(その2)

- 標準形の線形計画問題
  - 最適値を下から見積もりたい(最適値の下界値の計算)

目的関数: $-2x_1 - x_2 - x_3$ →最小化
制約条件: $-x_1 - 2x_2 \qquad\qquad = -12$ ①
$x_1 + 4x_2 + 2x_3 = 20$ ②
$x_1, x_2, x_3 \geq 0$

各制約に変数を
掛けて足し合わせる

①×y_1+②×y_2:

$$(-y_1 + y_2)x_1 + (-2y_1 + 4y_2)x_2 + (2y_2)x_3 = \boxed{-12y_1 + 20y_2}$$

これが下界値になるための条件

x_1の係数に関する条件: $-y_1 + y_2 \leq -2$
x_2の係数に関する条件: $-2y_1 + 4y_2 \leq -1$
x_3の係数に関する条件: $2y_2 \leq -1$

これが下界値
出来るだけ
大きくしたい

最適値の見積もりの計算(その3)

- 標準形の線形計画問題
  - 最適値を下から見積もりたい(最適値の下界値の計算)
  - 一番良い(大きい)下界値を求める問題は次のように定式化される

目的関数: $-12y_1 + 20y_2$ → 最大化
制約条件: $-y_1 + y_2 \leq -2$
$-2y_1 + 4y_2 \leq -1$
$2y_2 \leq -1$
(変数の非負条件は無し)

主問題の各変数と双対問題の各制約条件は1対1対応

主問題の各制約条件と双対問題の各変数は1対1対応

これは線形計画問題
元の問題の双対問題と呼ばれる
元の問題は主問題と呼ばれる

標準形の双対問題

- 標準形の線形計画問題（主問題）

目的関数：$c_1x_1 + c_2x_2 + \cdots + c_nx_n \ \to$ 最小化

制約条件：

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 & \leftrightarrow y_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 & \leftrightarrow y_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m & \leftrightarrow y_m
\end{aligned}
$$

$x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0$

主問題の各制約条件と
双対問題の各変数は
1対1対応

- その双対問題

目的関数：$b_1y_1 + b_2y_2 + \cdots + b_my_m \ \to$ 最大化

制約条件：

$$
\begin{aligned}
a_{11}y_1 + a_{21}y_2 + \cdots + a_{m1} &\leq c_1 & \leftrightarrow x_1 \\
a_{12}y_1 + a_{22}y_2 + \cdots + a_{m2} &\leq c_2 & \leftrightarrow x_2 \\
&\vdots \\
a_{1n}y_1 + a_{2n}y_2 + \cdots + a_{mn} &\leq c_n & \leftrightarrow x_n
\end{aligned}
$$

（変数の非負制約は無し）

主問題の各変数と
双対問題の各制約条件は
1対1対応

弱双対定理とその系(その1)

双対問題の定義より，次の重要な性質が導ける

弱双対定理：主問題と双対問題のそれぞれの任意の実行可能解 $(x_1, x_2, \ldots, x_n)$ および $(y_1, y_2, \ldots, y_m)$ に対して
$\sum_{i=1}^{n} c_i x_i \geq \sum_{j=1}^{m} b_i y_i$ が成立

弱双対定理から，次の性質が導ける

系1：主問題の任意の実行可能解 $(x_1, x_2, \ldots, x_n)$ に対して，
$\sum_{i=1}^{n} c_i x_i \geq$ 双対問題の最大値 が成立.
双対問題の任意の実行可能解 $(y_1, y_2, \ldots, y_m)$ に対して
$\sum_{j=1}^{m} b_i y_i \leq$ 主問題の最小値 が成立

弱双対定理とその系(その2)

弱双対定理から，次の性質が導ける

系2：主問題と双対問題のそれぞれの任意の実行可能解
$(x_1, x_2, \ldots, x_n)$および$(y_1, y_2, \ldots, y_m)$が
$\sum_{i=1}^{n} c_i x_i = \sum_{j=1}^{m} b_i y_i$を満たす
➔ $(x_1, x_2, \ldots, x_n)$および$(y_1, y_2, \ldots, y_m)$は
主問題と双対問題の最適解

系3：主問題が非有界ならば，双対問題は実行可能解をもたない
双対問題が非有界ならば，主問題は実行可能解をもたない

黒板で証明

主問題と双対問題の関係

性質:双対問題の双対問題は主問題に一致する

証明→黒板で

手順その1:

（1）双対問題を標準形に書き換え

（2）書き換えた問題の双対問題をつくる

（3）得られた双対問題を書き換えて,

元の問題（主問題）に一致することを確かめる.

手順その2:

双対問題の最適値のもっとも小さい上界値を求める問題を作る

➔ 元の問題（主問題）に一致することを確かめる

双対定理

- 主問題の最小値と双対問題の最大値は一致する！

双対定理：主問題と双対問題のどちらか一方が最適解をもつ
➔もう一方も最適解をもち，
「主問題の最小値＝双対問題の最大値」が成立

（証明の概略）

２段階シンプレックス法で主問題の最適基底解を求める

➔そのときの目的関数の係数から

双対問題の最適解が得られ，目的関数値は一致

双対定理の証明の流れ

主問題：最適基底解を求める

目的：$-2x_1 - x_2 - x_3$→最小化
制約：$-x_1 - 2x_2 \qquad = -12$
$x_1 + 4x_2 + 2x_3 = 20$
$x_1, x_2, x_3 \geq 0$

目的：$-28 + 4x_2$→最小化
制約：$x_1 = 12 - 2x_2$
$x_3 = 4 - x_2$
$x_1, x_2, x_3 \geq 0$

主問題の最適基底解(12,0,4)

双対問題

目的：$-12y_1 + 20y_2$→最大化
制約：$-y_1 + y_2 \leq -2$
$-2y_1 + 4y_2 \leq -1$
$2y_2 \leq -1$
（変数の非負条件は無し）

x_1, x_3が基底変数
➔双対問題の1，3番の制約を等式にして解を求める
$-y_1 + y_2 = -2$, $2y_2 = -1$
➔$(y_1, y_2) = \left(\frac{3}{2}, -\frac{1}{2}\right)$
これは双対問題の実行可能解
目的関数値＝主問題の最小値-28
➔弱双対定理より最適解

不等式制約の場合の双対問題(その1)

- 全てが不等式制約の問題の双対問題の形は？

目的関数: $c_1x_1 + c_2x_2 + \cdots + c_nx_n \quad \rightarrow$ 最小化
制約条件: $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq b_1$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq b_m$$
$$x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0$$

- まず標準形に変形

目的関数: $c_1x_1 + c_2x_2 + \cdots + c_nx_n \quad \rightarrow$ 最小化
制約条件: $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n - x_{n+1} = b_1$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n - x_{n+2} = b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n - x_{n+m} = b_m$$
$$x_1 \geq 0, \ldots, x_n \geq 0, x_{n+1} \geq 0, \ldots, x_{n+m} \geq 0$$

不等式制約の場合の双対問題(その2)

- 標準形の双対問題を計算

$$
\begin{array}{ll}
\text{目的関数：} & b_1y_1 + b_2y_2 + \cdots + b_my_m \quad \to \text{最大化} \\
\text{制約条件：} & a_{11}y_1 + a_{21}y_2 + \cdots + a_{m1} \le c_1 \\
 & a_{12}y_1 + a_{22}y_2 + \cdots + a_{m2} \le c_2 \\
 & \vdots \\
 & a_{1n}y_1 + a_{2n}y_2 + \cdots + a_{mn} \le c_n \\
 & -y_1 \le 0 \\
 & -y_2 \le 0 \\
 & \vdots \\
 & -y_m \le 0
\end{array}
$$

（変数の非負制約は無し）

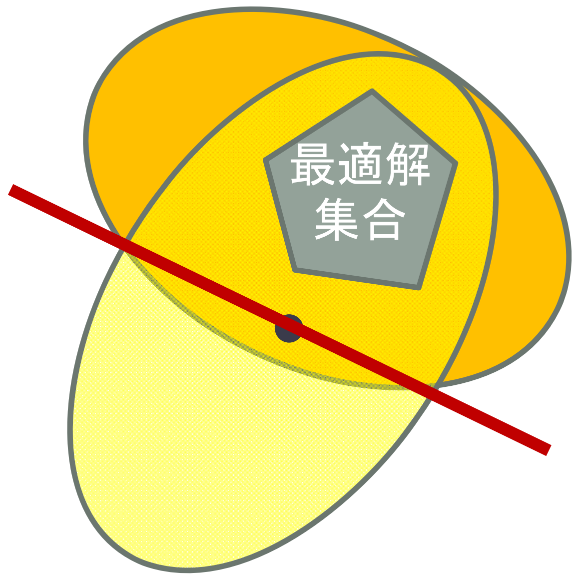
- 次の形に書き換えられる

$$
\begin{array}{ll}
\text{目的関数：} & b_1y_1 + b_2y_2 + \cdots + b_my_m \quad \to \text{最大化} \\
\text{制約条件：} & a_{11}y_1 + a_{21}y_2 + \cdots + a_{m1} \le c_1 \\
 & a_{12}y_1 + a_{22}y_2 + \cdots + a_{m2} \le c_2 \\
 & \vdots \\
 & a_{1n}y_1 + a_{2n}y_2 + \cdots + a_{mn} \le c_n \\
 & y_1 \ge 0, y_2 \ge 0, \ldots, y_m \ge 0
\end{array}
$$

線形計画問題のアルゴリズム(その1)

- シンプレックス法は実用的には高速
  - ただし，理論的には指数時間が必要なケース有り
- 線形計画問題に対する多項式時間アルゴリズムは存在する
- 例1：楕円体法(§2.8)
  - 最初の多項式時間アルゴリズム，ただし実用的には遅い

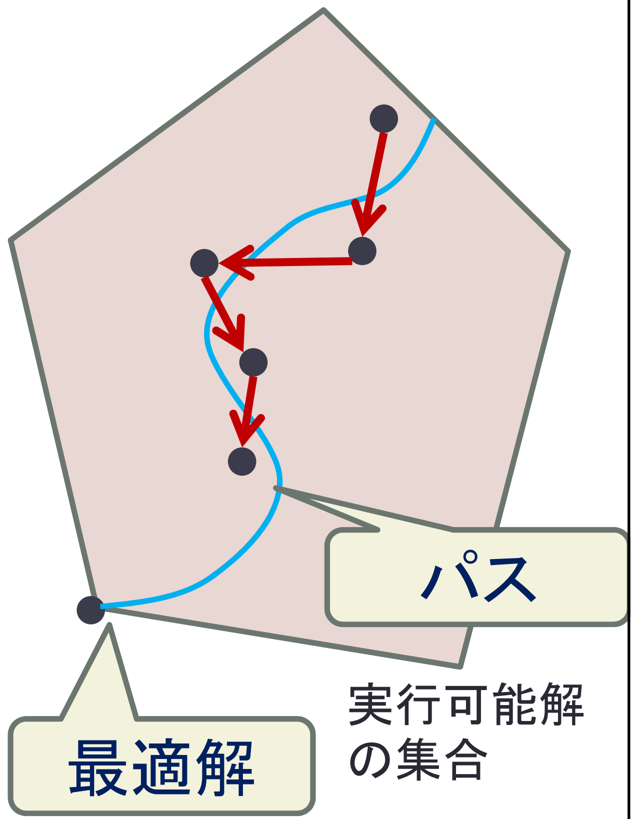
0. 最適解集合を含む楕円体を計算
1. 楕円体の中心が最適解集合に含まれる
   ➔終了
2. 楕円体の中心が最適解集合に含まれない
   ➔最適解集合を切らないように，楕円体を
   半分にする
3. 半分の楕円体を含む新たな楕円体を計算



線形計画問題のアルゴリズム(その2)

- 線形計画問題に対する多項式時間アルゴリズムは存在する
- 例2: 内点法(§2.9)
  - 多項式時間アルゴリズム, 実用的にも高速
  - 現在の主流のアルゴリズム, 多くのソフトウェアで使われている

0. 最適解につながる「パス(path)」を仮想的に考える
1. 適当な初期実行可能解から出発, パスに沿って, 最適解に繰り返し近づけていく.



第5章 組合せ計画

§５．２ 分枝限定法

生産計画問題の定式化

- 目的: $70x_1 + 120x_2 + 30x_3 \rightarrow$ 最大化
- 条件:
$$
\begin{aligned}
5x_1 \qquad\quad + 6x_3 &\le 80 \\
2x_2 + 8x_3 &\le 50 \\
7x_1 \qquad\quad + 15x_3 &\le 100 \\
3x_1 + 11x_2 \qquad\quad &\le 70 \\
x_1 \ge 0, x_2 \ge 0, x_3 &\ge 0
\end{aligned}
$$
x_1, x_2, x_3 は整数

変数(の一部)に
整数条件が付加
➔整数計画問題

見かけは線形計画問題と同じ
でも，最適解の計算は格段に難しくなる

整数計画問題の例2:ナップサック問題

- ハイキングの準備
- n個の品物の中から持って行くものを選択
- ナップサックには b kg まで入れられる
- 品物 $i = 1,2,\ldots,n$ の重さは a_i kg, 利用価値は c_i
- 利用価値の合計を最大にしたい

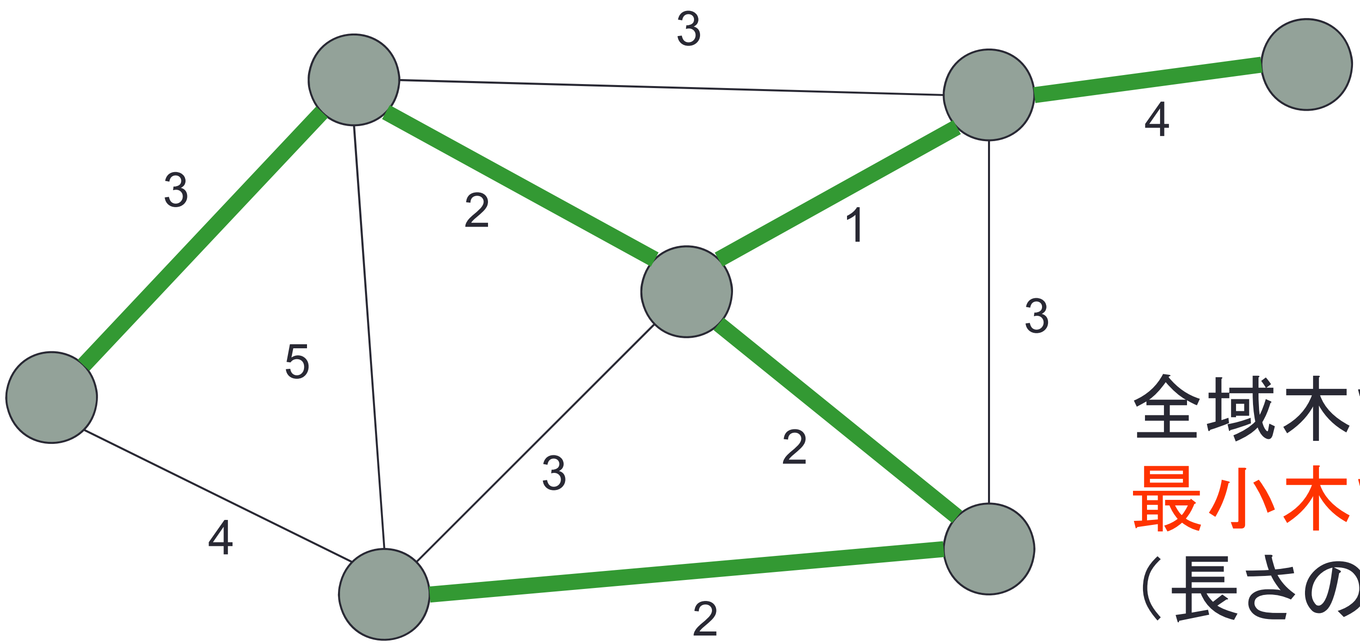
↓

目的関数: $\sum_{i=1}^{n} c_i\, x_i$ → 最大
制約条件: $\sum_{i=1}^{n} a_i\, x_i \leq b$
$x_1, x_2, \ldots, x_n \in \{0,1\}$

変数の全てが
0または1
→0-1整数計画問題

最小木問題

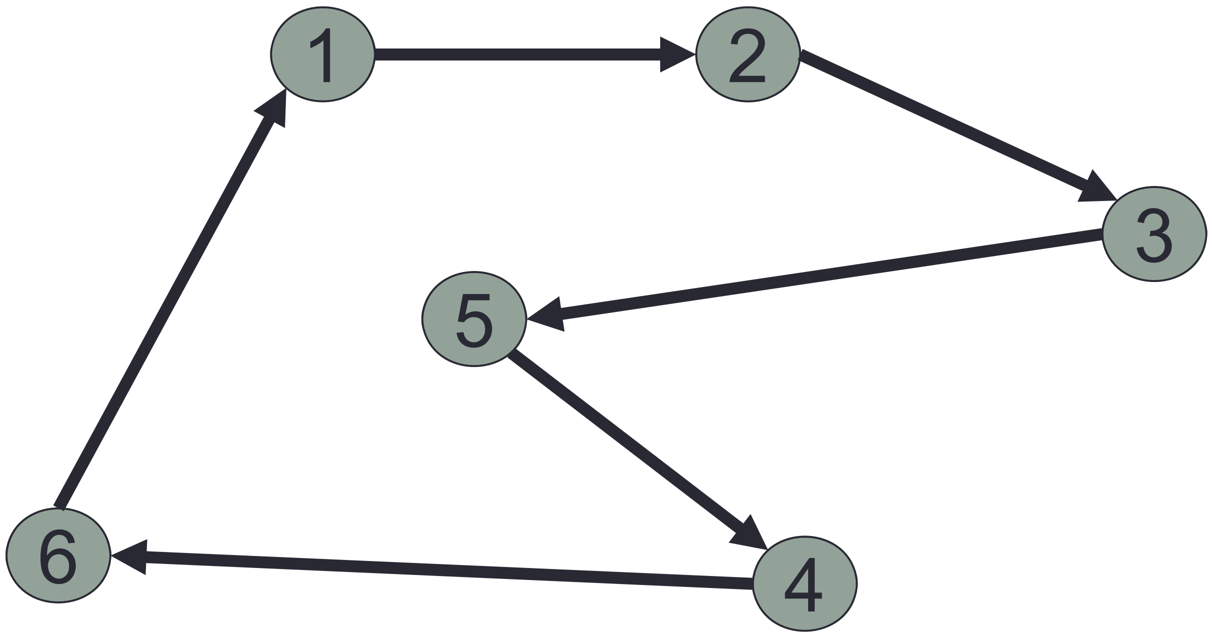
- 入力：無向グラフG=(V,E)，各枝の長さd(e) (e∈E)
- 出力：Gの最小木（Gの全域木で，枝の長さの和が最小のもの）



全域木であり，
最小木である
（長さの和＝14）

巡回セールスマン問題

- セールスマンが n 都市をちょうど一回ずつ巡回する
- 都市 i から j への距離は c_{ij}（平面上の距離で与えられるケースも多い）
- 目的:都市を巡回する際の総距離を最小にする



組合せ計画問題

- 組合せ計画問題とは：
  - 有限個の「もの」の組合せの中から，目的関数を最小または最大にする組合せを見つける問題
  - 例1：整数計画問題全般（整数の組合せ）
  - 例2：グラフの最小木問題，最短路問題，（グラフの枝の組合せ）
  - 例3：巡回セールスマン問題（都市の順列）

- 解きやすい問題と解きにくい問題
  - 解きやすい問題≒多項式時間で解ける問題
  - 解きにくい問題≒NP困難な問題

（多項式時間で解けないと信じられている問題）

※注意：組合せ計画問題の解は有限個→有限時間で必ず解ける！

組合せ計画問題に対するアプローチ

- 組合せ計画問題をどのように解くか？
- 解きやすい問題の場合
  - 多項式時間アルゴリズムを構築（教科書 § 5.1）➔より高速な解法へ
- 解きにくい問題の場合
  - 絶対に最適解が必要な場合➔厳密解法
    - 分枝限定法（ § 5.2, 授業で説明）←現在の主流
    - 動的計画法（ § 5.3,「アルゴリズムとデータ構造」）
  - ある程度良い解であれば十分という場合
    - 精度保証付き近似アルゴリズム

      （解の良さに対する理論保証あり）
    - ヒューリスティックス（解の良さは実験的に証明）（ § 5.4, 5.5）

組合せ計画問題に対する厳密解法

- 組合せ計画問題は解を全列挙すれば解ける
- しかし，計算時間が膨大で現実には不可能

  ➔ 解の全列挙における無駄を出来るだけ省く

  - 動的計画法：同一の部分問題を繰り返し解かない
  - 分枝限定法：ある部分問題から最適解が得られないことがわかったら，その部分問題は無視する

分枝限定法の考え方

- 組合せ計画問題を，場合分けによって部分問題に分解（分枝操作）
  - 0–1ナップサック問題：各変数について 0 の場合と 1 の場合に分ける
  - 巡回セールスマン問題：次に訪問する都市によって場合分け
- 分枝の進行の様子は探索木により表現可能

0-1ナップサック問題の探索木

目的関数 $10x_1 + 7x_2 + 25x_3 \rightarrow$ 最大

制約条件 $2x_1 + x_2 + 6x_3 \leq 7$

$x_j \in \{0,1\}\ (j = 1,2,3)$

x_1を0に固定

目的関数 $7x_2 + 25x_3 \rightarrow$ 最大

制約条件 $x_2 + 6x_3 \leq 7$

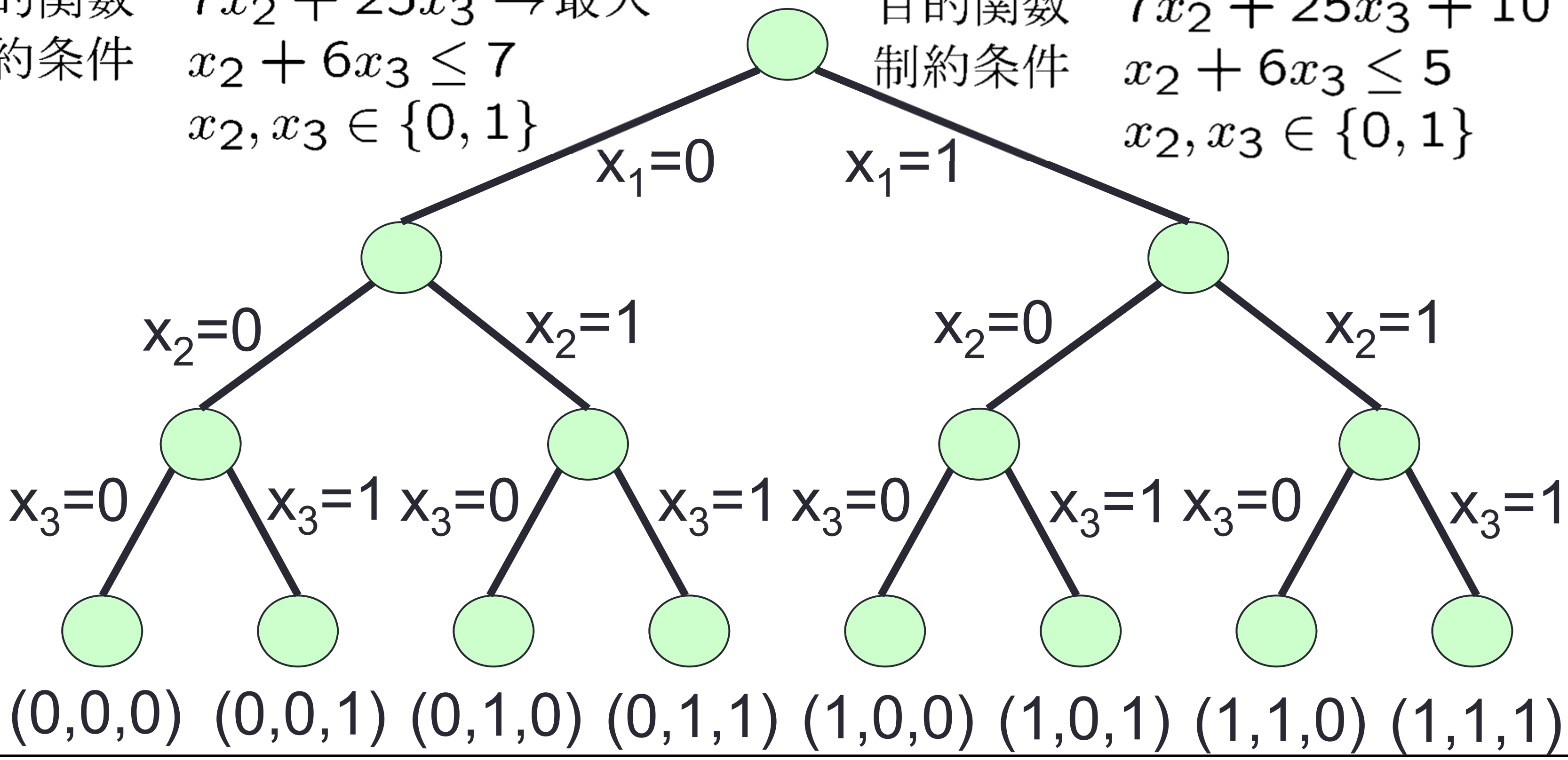
$x_2, x_3 \in \{0,1\}$

x_1を1に固定

目的関数 $7x_2 + 25x_3 + 10 \rightarrow$ 最大

制約条件 $x_2 + 6x_3 \leq 5$

$x_2, x_3 \in \{0,1\}$



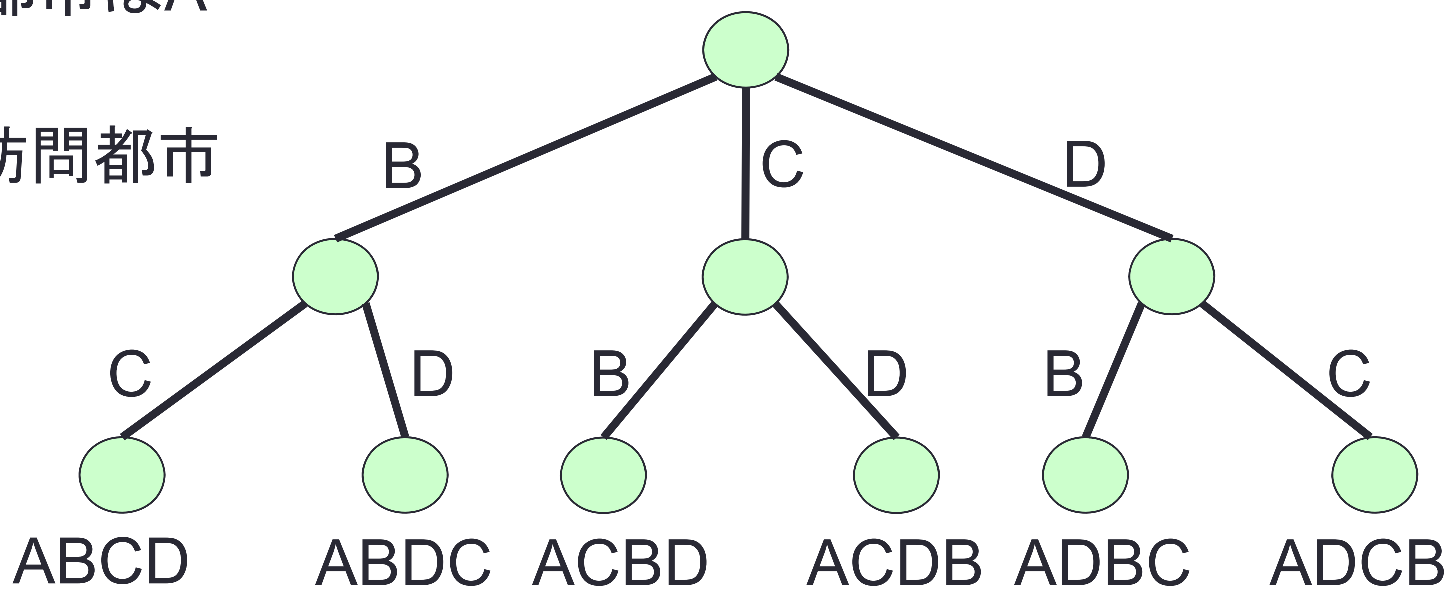
巡回セールスマン問題の探索木

- 4都市{A,B,C,D}の対称巡回セールスマン問題の場合
- 最初の訪問都市はA

2番目の訪問都市

3番目の訪問都市

最終的な訪問の順番



分枝限定法の考え方

- 分枝操作により，たくさんの部分問題が生成される
- 解いても無駄な部分問題が検出されたら，さらなる分枝操作をストップ（限定操作）
- 解いても無駄な部分問題の例
  - 最適解がすでに得られた
  - 現在の暫定解より良い許容解を得られる可能性がなくなった
    - 緩和問題を利用
  - 許容解が存在しない（実行不可能）
- 暫定解：分枝限定法のそれまでの計算により得られている最良の許容解

緩和問題

- 緩和問題：元の数理計画問題の制約条件の一部を緩和して得られる問題

0-1ナップサック問題

目的関数：$\sum_{i=1}^{n} c_i\, x_i$ → 最大
制約条件：$\sum_{i=1}^{n} a_i\, x_i \leq b$
$x_1, x_2, \ldots, x_n \in \{0,1\}$

$x_j \in \{0,1\}$を$0 \leq x_j \leq 1$に緩和

連続ナップサック問題

目的関数：$\sum_{i=1}^{n} c_i\, x_i$ → 最大
制約条件：$\sum_{i=1}^{n} a_i\, x_i \leq b$
$0 \leq x_j \leq 1 \ (j = 1,2, \ldots, n)$

連続ナップサック問題は線形計画問題➔多項式時間で解ける
O(n)時間アルゴリズムが存在（「アルゴリズムとデータ構造」）

緩和問題の性質

- 緩和問題は元の問題より解きやすい（簡単に解ける）ことが多い
  - 分枝限定法では，簡単に解ける問題を緩和問題として選ぶ
- 緩和問題は元の問題の条件を緩和した問題

  ➔緩和問題の実行可能解集合は，元の問題の

  実行可能解集合を含む

  ➔緩和問題の最大値≧元の問題の最大値

  ∴緩和問題の最適値（最大値）は，元の問題の最適値の上界
- 緩和問題の最適解を修正することにより，元の問題の実行可能解を作ることが可能なケースが多い
  - 緩和問題の最適解が元の問題の実行可能解

    ➔元の問題の最適解になっている！

0-1ナップサック問題の緩和問題：例

緩和問題の最適解は，c_j/a_j の大きい方から順に変数を増やしていけば得られる

	1	2	3	4	5	6	7	8
c_j	15	100	90	60	40	15	10	1
a_j	2	20	20	30	40	30	60	10

b=102

a_jの合計72≦b

(1,1,1,1,(102-72)/40,0,0,0)は最適解
目的関数値＝295
≧0-1ナップサック問題の最適値

$x_5 = 0$に置き換えた解(1,1,1,1,0,0,0,0)は元問題の実行可能解
目的関数値＝265
$x_4 = 0$にして$x_5 = 1$とした解(1,1,1,0,1,0,0,0)も元問題の実行可能解
目的関数値＝245

レポート問題(〆切:次回授業13:05まで)

問1:次の線形計画問題について考える

(1)この問題の双対問題を書け.

(2)この問題の最適基底解は(2,3,0,0)である.スライド「双対定理の証明の流れ」の要領で,双対問題の最適解を計算せよ.

目的関数: $-x_1 - x_2 \quad \rightarrow$ 最小化

制約条件: $3x_1 + 2x_2 + x_3 = 12$

$x_1 + 2x_2 + x_4 = 8$

$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$

問2:(1)次の0-1ナップサック問題[a]の最適解を力ずくで計算せよ.

(2) 0-1ナップサック問題[a],[b],[c]の緩和問題の最適解を計算せよ.

a 目的関数 $10x_1 + 7x_2 + 25x_3 \rightarrow$ 最大

制約条件 $2x_1 + x_2 + 6x_3 \leq 7$

$x_j \in \{0,1\} \ (j = 1,2,3)$

b 目的関数 $7x_2 + 25x_3 \rightarrow$ 最大

制約条件 $x_2 + 6x_3 \leq 7$

$x_2, x_3 \in \{0,1\}$

c 目的関数 $7x_2 + 25x_3 + 10 \rightarrow$ 最大

制約条件 $x_2 + 6x_3 \leq 5$

$x_2, x_3 \in \{0,1\}$