組合せ最適化問題に対する解法

厳密解法 最悪の場合，求解までに時間がかかるかもしれないが厳密に最適解を求める

- 分枝限定法…＿＿＿＿＿＿＿＿ どんな組合せ最適化問題に対しても有効な＿＿＿＿＿＿＿＿＿＿＿＿，＿＿＿＿＿＿＿＿＿＿＿＿，最適解を求める方法である．
- 分枝カット法
- 動的計画法
- …

近似解法 最適解を求めることを諦めて，ある程度最適値に近い値をもつ実行可能解をなるべく早く求める

- 制度保証付き近似解法
- メタヒューリスティック（タブーサーチ，遺伝的アルゴリズム，アニーリング法…）
- 乱択アルゴリズム
- …

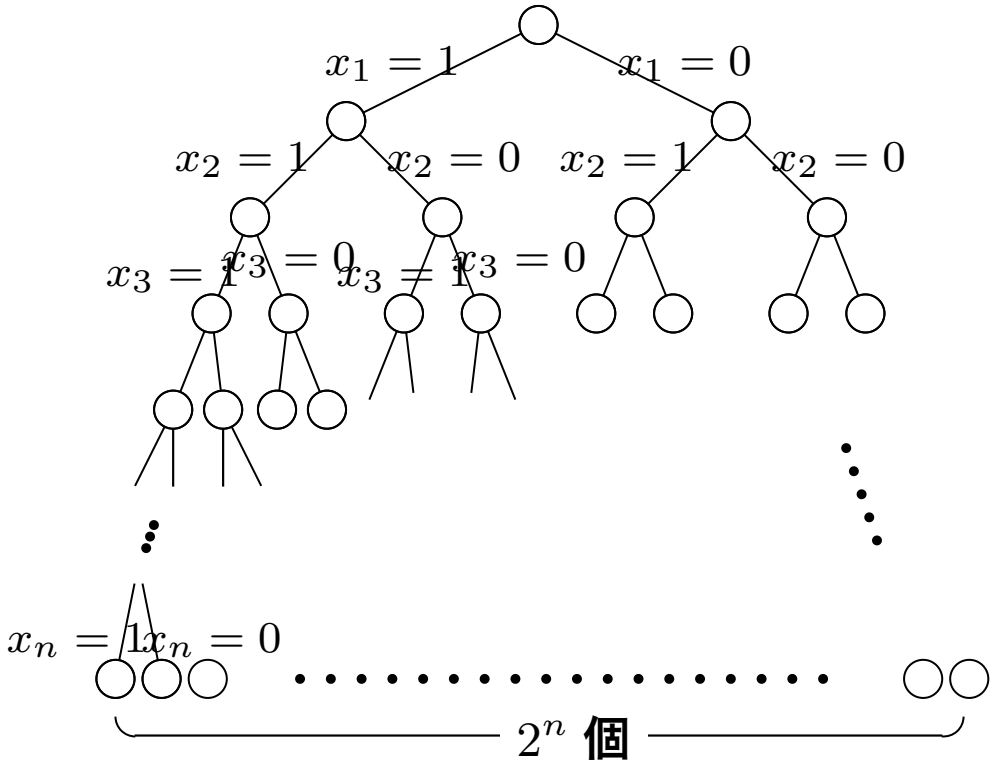
解の列挙

$$
\left|
\begin{array}{lll}
\text{最大化} & c_1x_1 + c_2x_2 + \cdots + c_nx_n & \\
\overset{x}{\text{条}}\text{件} & a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i, & i = 1, 2, \ldots, m \\
 & x_j \in \{0, 1\}, & j = 1, 2, \cdots, n
\end{array}
\right.
$$

のすべての解を列挙する.

- **[　　]の** 0-1 **変数** $x_1, \ldots, x_n$ $\Rightarrow$ **解（実行不能解も）は** 0-1 **の組み合わせから[　]個ある**
- **[　　]— システマティックに列挙**
  1.
  2.
  3.

列挙木



- 列挙木は枝分かれするごとに _____
- _____ いくつかの変数が固定されている状態を表す
- 一番下の頂点では _____

要領よく列挙

- 実行可能解になり得ない解を無視する
  - 固定した変数から実行不可能だと分かれば，それ以降は探索しない
- 最適解になり得ない解を無視する
- 列挙の途中で最適解を得る

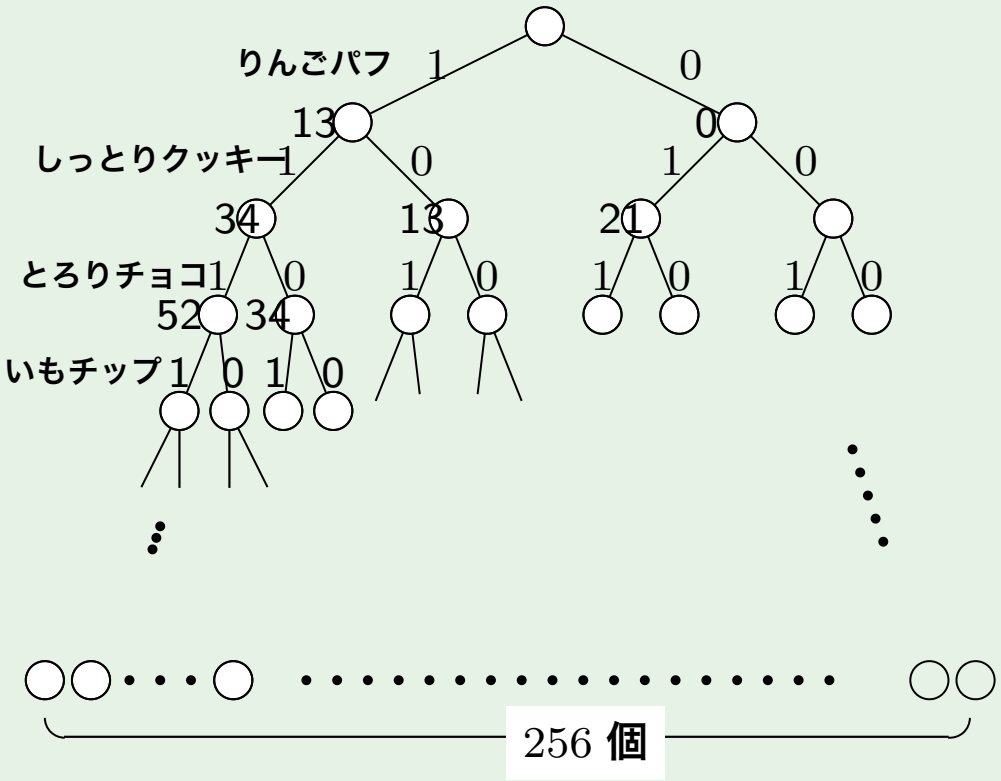
ナップサック問題に対する列挙の打ち切り 1

8 個のお菓子の中から 500 円以内で価値の和が最大となるように選択したい.

アイテム	ガム	あめ	りんごパフ	とろりチョコ	山盛り煎餅	いもチップ	ラムネ	しっとりクッキー
値段 a_j (十円)	5	1	13	20	15	8	7	21
価値 c_j	2	1	8	7	4	6	3	8
効率 c_j/a_j	0.40	1.00	0.62	0.35	0.27	0.75	0.43	0.38

価値の順に列挙木を構成

リンゴパフ，しっとりクッキー，とろりチョコの3つを選択する（対応する変数を 1 に固定する）と，制約の 50（十円）を超える.



ナップサック問題に対する列挙の打ち切り 2

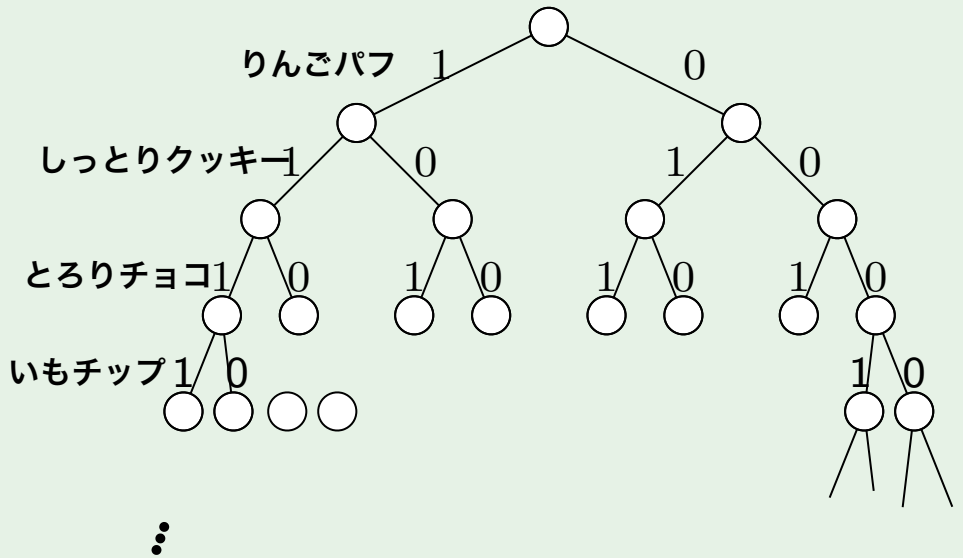
アイテム	ガム	あめ	りんごパフ	とろりチョコ	山盛り煎餅	いもチップ	ラムネ	しっとりクッキー
値段 a_j (十円)	5	1	13	20	15	8	7	21
価値 c_j	2	1	8	7	4	6	3	8
効率 c_j/a_j	0.40	1.00	0.62	0.35	0.27	0.75	0.43	0.38

貪欲アルゴリズムを使って求めた解

アイテム	ガム	あめ	りんごパフ	とろりチョコ	山盛り煎餅	いもチップ	ラムネ	しっとりクッキー
解								

目的関数値は □ ←

リンゴパフ，しっとりクッキー，とろりチョコの３つを選択しないと残りのアイテムを選択しても価値の合計は 16

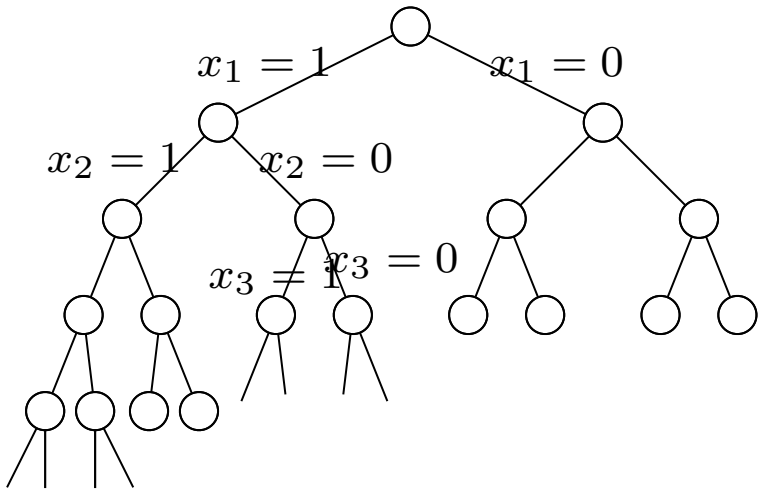


子問題

- 列挙木の中間頂点で，それまでに決められた変数を固定した問題 ..
  変数を固定する前の与えられた問題...
- ある中間頂点において，
  - 1 に固定されている変数の添え字の集合：I
  - 0 に固定されている変数の添え字の集合：O
  - まだ，どちらにも固定されていない変数の添え字の集合：F

とすると，0-1 計画問題に対するこの中間頂点における子問題は

$$
\begin{array}{ll}
\text{最大化} & \sum_{j\in F} c_j x_j \quad + \quad \sum_{j\in I} c_j \\
\underset{\boldsymbol{x}}{\text{条件}} & \sum_{j\in F} a_{ij} x_j \quad + \quad \sum_{j\in I} a_{ij} \quad \leq \quad b_i, \quad i = 1, 2, \ldots, m \\
 & x_j \in \{0, 1\}, \quad j \in F
\end{array}
$$



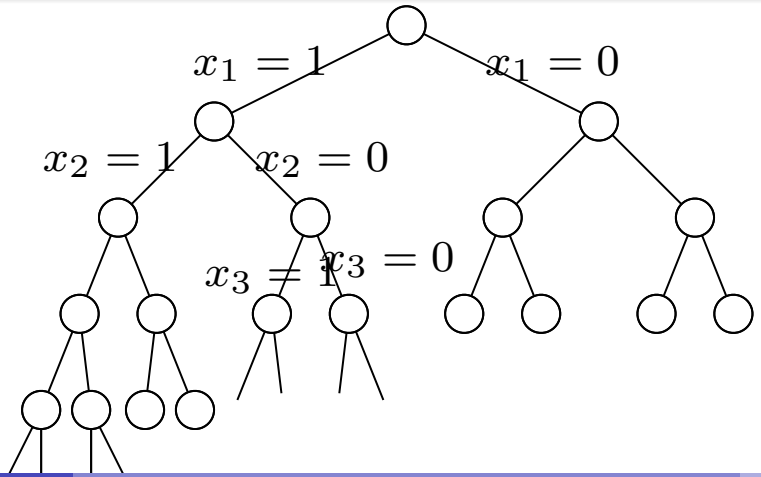
子問題の最適解

- 子問題の最適解は，対応する中間頂点の下に列挙される解の中で最も目的関数値が大きい解

定理

- [　　] $\Rightarrow$ この中間頂点より下に列挙される最適解は存在しない
  (子問題の上界値は[　　]を解けば得られる．)
- [　　]ならば，この解が中間頂点より下に列挙される中で最も良い解

いずれの場合も列挙を打ち切れる



ナップサック問題に対する列挙の打ち切り 3

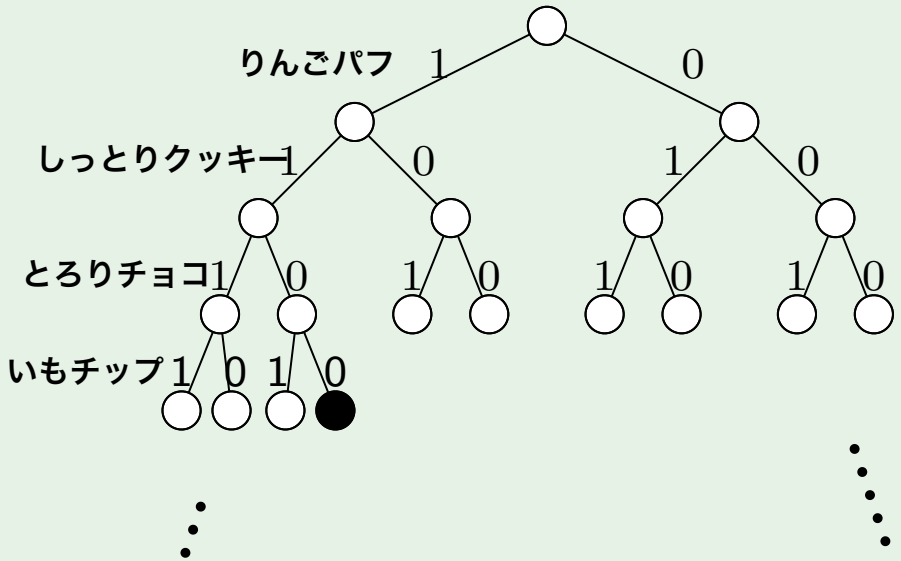
アイテム	ガム	あめ	りんごパフ	とろりチョコ	山盛り煎餅	いもチップ	ラムネ	しっとりクッキー
値段 a_j (十円)	5	1	13	20	15	8	7	21
価値 c_j	2	1	8	7	4	6	3	8
効率 c_j/a_j	0.40	1.00	0.62	0.35	0.27	0.75	0.43	0.38

子問題： $I = \{$ りんごパフ，しっとりクッキー $\}$，$O = \{$ とろりチョコ，いもチップ $\}$

連続緩和問題の解

アイテム	ガム	あめ	りんごパフ	とろりチョコ	山盛り煎餅	いもチップ	ラムネ	しっとりクッキー
解			$\underline{1}$	$\underline{0}$		$\underline{0}$		$\underline{1}$

上界値は $\boxed{}$ $\longleftrightarrow$



分枝限定法概要

Definition

- [ ]... 分枝限定法の過程で生成される列挙木
- [ ]... 分枝限定法の過程で得られたもっとも良い実行可能解
- [ ]... 暫定解の目的関数値

分枝操作 .

限定操作 .

1. 子問題が実行不可能ならば，そこで打ち切り.
2. 子問題の上界値が暫定値よりも小さければ，そこで打ち切り.（最大化問題が対象）
3. 子問題の連続緩和問題の最適解がもとの問題で実行可能ならば，その子問題に対応する中間頂点の下にある解の中の最適解が得られたことになり，打ち切り.（暫定値よりも目的関数値が良ければ，暫定解の更新）

[ ]が打ち切られたときに終了. [ ]が最適解となる.

ナップサック問題を分枝限定法で解く

アイテム	ガム	あめ	りんごパフ	とろりチョコ	山盛り煎餅	いもチップ	ラムネ	しっとりクッキー
値段 a_j (十円)	5	1	13	20	15	8	7	21
価値 c_j	2	1	8	7	4	6	3	8
効率 c_j/a_j	0.40	1.00	0.62	0.35	0.27	0.75	0.43	0.38

貪欲アルゴリズムを使って求めた解

アイテム	ガム	あめ	りんごパフ	とろりチョコ	山盛り煎餅	いもチップ	ラムネ	しっとりクッキー
解								

目的関数値は □**.** ← **下界値**

分枝操作 1: $I = \{$ **りんごパフ** $\}, O = \emptyset$ **とした子問題** 1 **と,** $I = \emptyset$, $O = \{$ **りんごパフ** $\}$ **とした子問題** 2 **を作成.**

限定操作 1:子問題 1 は実行可能解をもつので連続緩和問題を解く.

アイテム	ガム	あめ	りんご パフ	とろり チョコ	山盛り 煎餅	いも チップ	ラムネ	しっとり クッキー
解			$\underline{1}$					

上界値は □ **で打ち切りはできない**

分枝操作 2: $I = \{$ **りんごパフ, しっとりクッキー** $\}, O = \emptyset$ **とした子問題** 1-1 **と,** $I = \{$ **りんごパフ** $\}, O = \{$ **しっとりクッキー** $\}$ **とした子問題** 1-2 **を作成.**

限定操作 2: 子問題 1-1 は実行可能解をもつので連続緩和問題を解く.

アイテム	ガム	あめ	りんご パフ	とろり チョコ	山盛り 煎餅	いも チップ	ラムネ	しっとり クッキー
解			1					1

[ ]であるので分枝を打ち切る.

目的関数値は[ ]で暫定値よりも良いので, 暫定値と暫定解を更新.

限定操作 3: 子問題 1-2 は実行可能解をもつので連続緩和問題を解く.

アイテム	ガム	あめ	りんご パフ	とろり チョコ	山盛り 煎餅	いも チップ	ラムネ	しっとり クッキー
解			1					0

上界値は[ ]で暫定値よりも小さいので, ここで打ち切る.

限定操作 4: 子問題 2 は実行可能解をもつので連続緩和問題を解く.

アイテム	ガム	あめ	りんご パフ	とろり チョコ	山盛り 煎餅	いも チップ	ラムネ	しっとり クッキー
解			0					

上界値は [ ] で暫定値よりも小さいので，ここで打ち切る.

ここで分枝限定法は終了し，現在の暫定解が最適解となる

一般的な整数計画問題に対する分枝限定法

$$
\begin{aligned}
\underset{\boldsymbol{x}}{\textbf{最大化}} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\textbf{条件} \quad & a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i, \quad i = 1, 2, \ldots, m \\
& 0 \leq x_j \leq k_j, x_j : \textbf{整数}, \quad j = 1, 2, \cdots, n
\end{aligned}
$$

0 より大きく k 未満の適切な整数 k' を用いて，$0 \leq x_j \leq k'$ を制約に追加した子問題と $k'+1 \leq x_j \leq k$ を制約に追加した子問題に分枝する

$x_j = 0$, $x_j = 1$, $\ldots$, $x_j = k$ と変数を固定して分枝する

実行可能領域の分割

分枝操作は実行可能領域を分割して子問題を作成している

元問題

$$\begin{array}{ll} \text{最大化} & f(\boldsymbol{x}) \\ \boldsymbol{x} & \\ \text{条件} & \boldsymbol{x} \in S \end{array}$$

分枝操作：実行可能領域 S を $S_1, S_2, \ldots, S_q$ に分割

各 $i(i = 1, \ldots, q)$ の子問題

$$\begin{array}{ll} \text{最大化} & f(\boldsymbol{x}) \\ \boldsymbol{x} & \\ \text{条件} & \boldsymbol{x} \in S_i \end{array}$$

変数 x_j に対して，

- **制約 $0 \leq x_j \leq k'$, あるいは，$k' + 1 \leq x_j \leq k$ を追加して分枝するときには，実行可能領域を 2 つに分割.**
- **$x_j = 0$, $x_j = 1$, $\cdots$, $x_j = k$ と変数を固定して分枝するときは，実行可能領域を $k + 1$ 個に分割.**

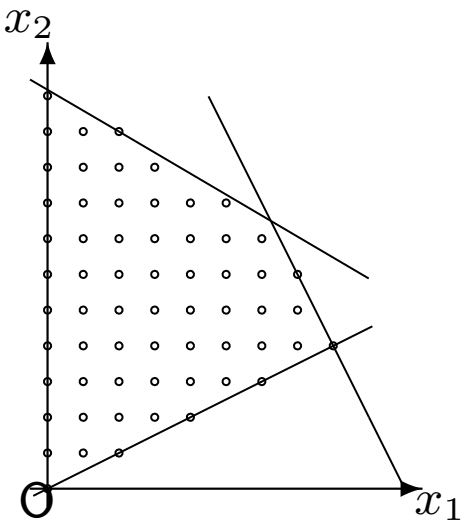
帰納的に分枝と子問題の求解を繰り返す

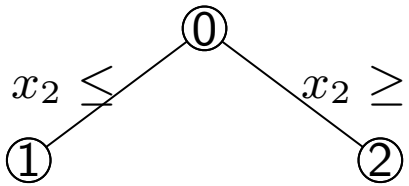
各子問題の最適解を求めることが困難な場合は，さらに実行可能領域を分割.

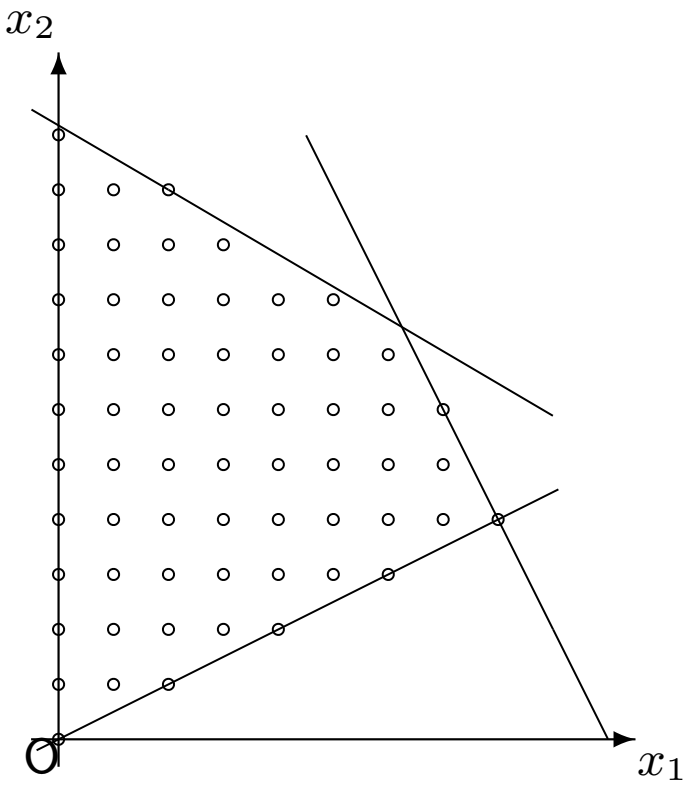
- 分割で作成された子問題のすべての最適解が求められれば，その中で最も良い解が元の問題の最適解.
- 子問題を解かなくても，子問題の実行可能領域内に元の問題の最適解が存在しないと分かれば，ここで分枝を打ち切る

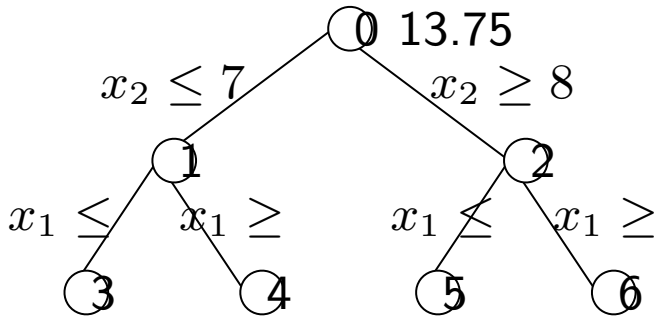
分枝限定法の例

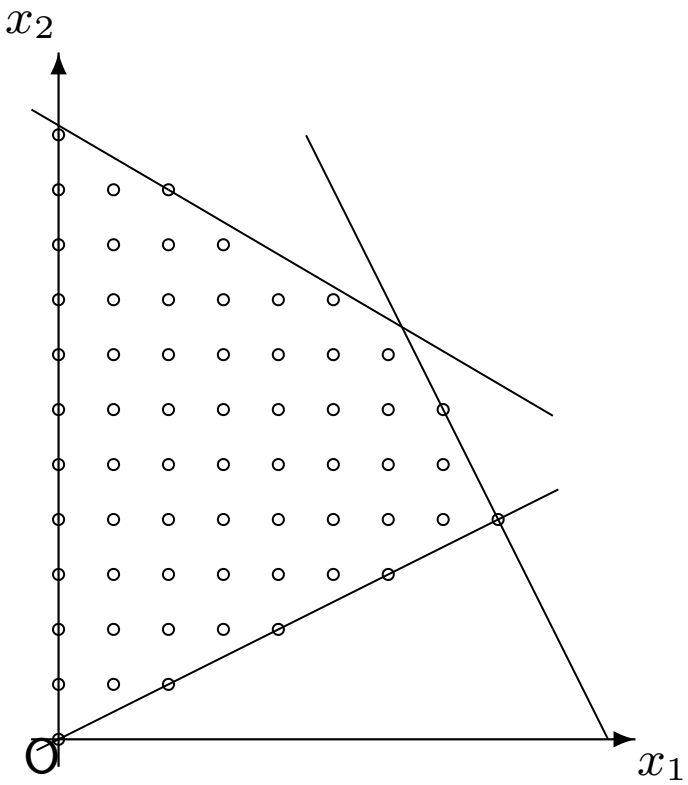
$$
\begin{array}{lrcrcl}
\textbf{最大化} & x_1 & + & x_2 & & \\
(x_1,x_2) & & & & & \\
\textbf{条　件} & 2x_1 & + & x_2 & \leq & 20 \\
 & x_1 & + & 1.7x_2 & \leq & 19 \\
 & x_1 & - & 2x_2 & \leq & 0 \\
 & x_1, x_2 & \geq & 0, & \textbf{整数} &
\end{array}
$$

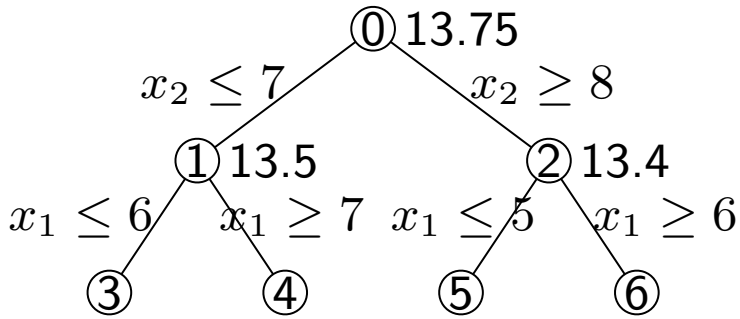


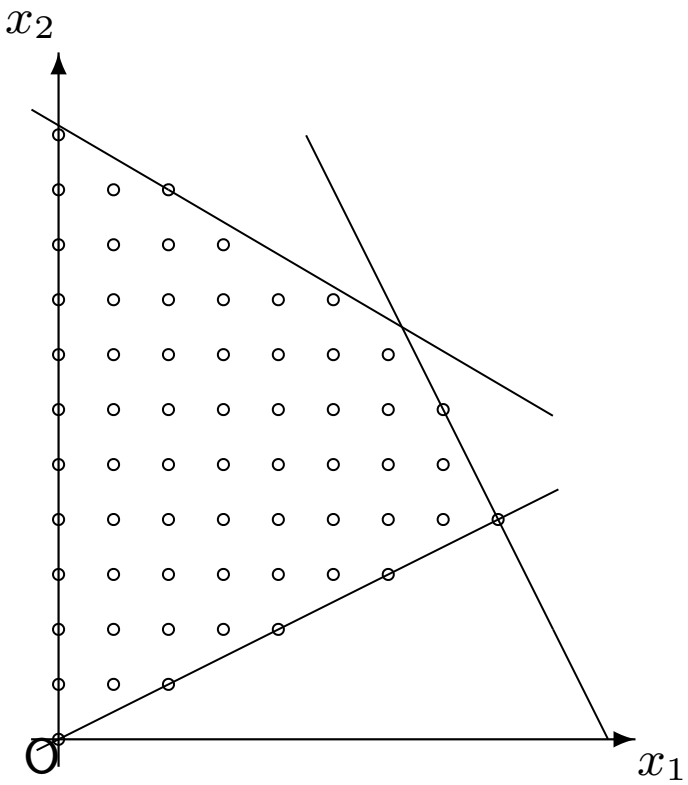


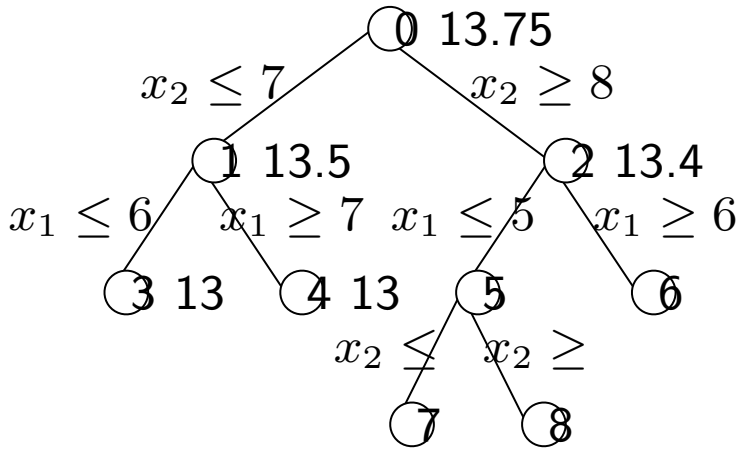

x_2
O
x_1

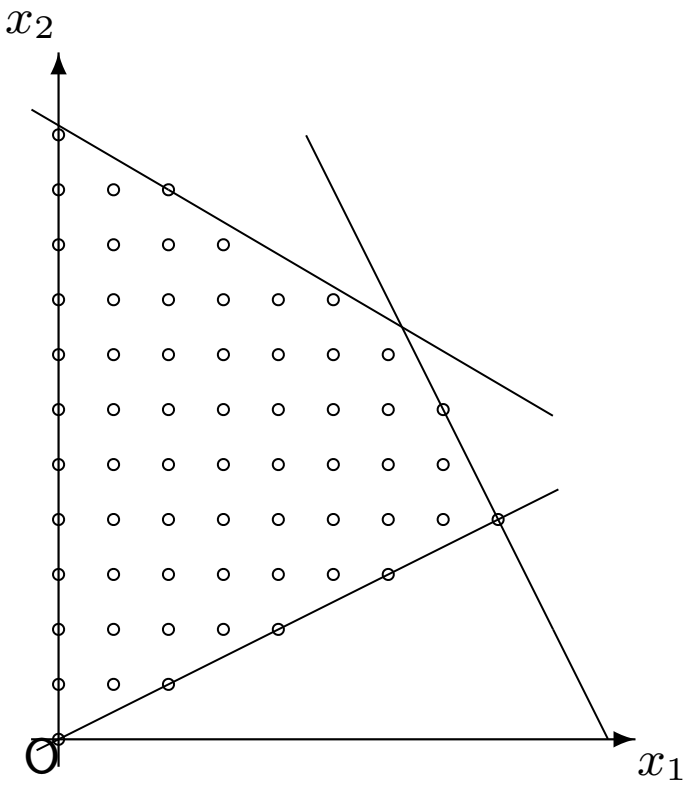

0 13.75
$x_2 \leq 7$
$x_2 \geq 8$
1
2
$x_1 \leq$
$x_1 \geq$
$x_1 \leq$
$x_1 \geq$
3
4
5
6

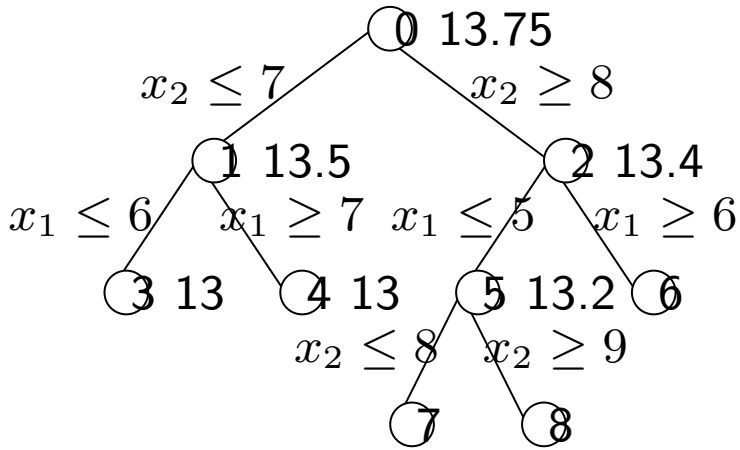

x_2
O
x_1


0 13.75
$x_2 \leq 7$
$x_2 \geq 8$
1 13.5
2 13.4
$x_1 \leq 6$
$x_1 \geq 7$
$x_1 \leq 5$
$x_1 \geq 6$
3
4
5
6


x_2
O
x_1


0 13.75
$x_2 \leq 7$
$x_2 \geq 8$
1 13.5
2 13.4
$x_1 \leq 6$
$x_1 \geq 7$
$x_1 \leq 5$
$x_1 \geq 6$
3 13
4 13
5
6
$x_2 \leq$
$x_2 \geq$
7
8


x_2
O
x_1


0 13.75
$x_2 \leq 7$
$x_2 \geq 8$
1 13.5
2 13.4
$x_1 \leq 6$
$x_1 \geq 7$
$x_1 \leq 5$
$x_1 \geq 6$
3 13
4 13
5 13.2
6
$x_2 \leq 8$
$x_2 \geq 9$
7
8

演習問題

1. 以下の 0-1 ナップサック問題を分枝限定法を用いて解け．ただし，分枝変数（新たに固定する変数）は，連続緩和問題を解いたときに連続値となった変数とせよ．

$$
\begin{array}{ll}
\underset{\boldsymbol{x}}{\text{最大化}} & 7x_1 + 12x_2 + 8x_3 + 5x_4 + 10x_5 \\
\text{条件} & 2x_1 + 3x_2 + 5x_3 + 2x_4 + 8x_5 \leq 11 \\
& x_j \in \{0,1\}, \quad j = 1,2,3,4,5
\end{array}
$$

2. 以下のナップサック問題を分枝限定法を用いて解け．

$$
\begin{array}{ll}
\underset{\boldsymbol{x}}{\text{最大化}} & x_1 + 8x_2 + 7x_3 + 2x_4 \\
\text{条件} & 3x_1 + 8x_2 + 6x_3 + 3x_4 \leq 27 \\
& x_1, x_4 \in \{0,1,2,3,4\} \\
& x_2, x_3 \in \{0,1,2\},
\end{array}
$$

組合せ最適化問題に対する解法

厳密解法 最悪の場合，求解までに時間がかかるかもしれないが厳密に最適解を求める

- 分枝限定法
- **分枝カット法**
  分枝限定法の途中，子問題を解く際にもとの実行可能領域を含み，緩和問題で得られた解を禁止するように新たに制約を追加して，緩和の精度をあげようとするアプローチ
- 動的計画法
- …

近似解法 最適解を求めることを諦めて，ある程度最適値に近い値をもつ実行可能解をなるべく早く求める

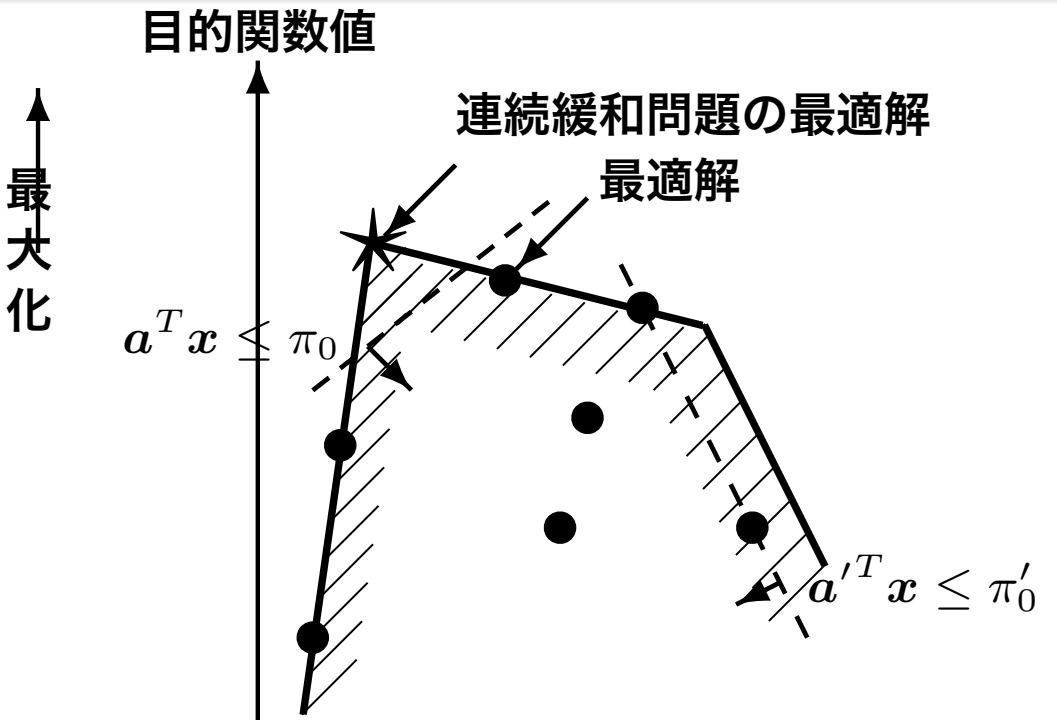
- 制度保証付き近似解法
- メタヒューリスティック（タブーサーチ，遺伝的アルゴリズム，アニーリング法…）
- 乱択アルゴリズム
- …

切除平面

元問題	最大化	$f(\boldsymbol{x})$
	$\boldsymbol{x}$	
	条件	$\boldsymbol{x} \in S$

Definition

任意の $x \in S$ で $\boldsymbol{a}^T \boldsymbol{x} \leq \pi_0$ が成立しているとき，$\boldsymbol{a}^T \boldsymbol{x} \leq \pi_0$ を**妥当不等式**という．
特に，連続緩和問題などで得られた（実行不能な）解 x' を禁止するような制約を**切除平面**という．



切除平面は問題の構造を活かして様々知られている．

切除平面（ゴモリーの小数カット）

$$
\begin{array}{ll}
\text{最大化} & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\underset{\boldsymbol{x}}{\text{条件}} & a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i, \quad i = 1, 2, \ldots, m \\
 & x_j \text{は整数}, \quad j = 1, 2, \cdots, n
\end{array}
$$

の代表的な切除平面

- 連続緩和問題の最適な基底解を $\tilde{x}$．非基底変数の集合：N
  - 整数値でない変数 $\tilde{x}_u$ が存在したとき
  - $x_u = b'_u - \sum_{j \in N} a'_{uj} x_j$ (1) より，b'_u は整数でない
- $\lfloor b'_u \rfloor = \lfloor x_u + \sum_{j \in N} a'_{uj} x_j \rfloor \geq \lfloor x_u \rfloor + \sum_{j \in N} \lfloor a'_{uj} \rfloor \lfloor x_j \rfloor$
- x_u は整数より，$x_u \leq \lfloor b'_u \rfloor - \sum_{j \in N} \lfloor a'_{uj} \rfloor x_j$ (2)
- (1)(2) より

$$
\lfloor b'_u \rfloor - b'_u \geq \sum_{j \in N} (\lfloor a'_{uj} \rfloor - a'_{uj}) x_j
$$

切除平面（離接カット）

$$
\left|
\begin{array}{lll}
\text{最大化} & c_1x_1 + c_2x_2 + \cdots + c_nx_n & \\
\overset{x}{\text{条件}} & a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i, & i = 1, 2, \ldots, m \\
 & x_j \in \{0, 1\} & j = 1, 2, \cdots, n
\end{array}
\right.
$$

- 連続緩和問題の最適解 $\tilde{x}$ で $\tilde{x}_u$ が連続値
- z_0: $x_u = 0$ と固定した子問題の最適値
  z_1: $x_u = 1$ と固定した子問題の最適値
-

$$\sum_{j=1}^{n} c_j x_j - (z_1 - z_0)x_u \leq z_0$$

  は妥当不等式．特に，$\min\{z_0, z_1\} < \sum_{j=1}^{n} c_j \tilde{x}_j$ のとき，$\tilde{x}$ を削除する切除平面．
- 整数値でない $\tilde{x}_u$ に対して $\tilde{x}_u = p$ としたとき，$x_u \geq \lceil p \rceil$ または $x_u \leq \lfloor p \rfloor$ を同時に表す不等式を**離接カット**という

切除平面（被覆不等式）

$$
\begin{array}{|ll}
\text{最大化} & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\overset{x}{\text{条}}\text{件} & a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i, \quad i = 1, 2, \ldots, m \\
 & x_j \in \{0, 1\} \quad j = 1, 2, \cdots, n
\end{array}
$$

- 添字の部分集合 $J \subseteq \{1, 2, \ldots, n\}$ に対して，ある制約式で $\sum_{j \in J} a_{ij} > b_i$ となるとき，J のすべての変数を 1 に出来ない．
- このとき，

$$\sum_{j \in J} x_j \leq |J| - 1$$

- 得られた解 $\tilde{x}$ に対して，

$$\sum_{j \in J} a_{ij} > b_i, \text{ かつ } \sum_{j \in J} \tilde{x}_j > |J| - 1$$

であるような J をみつければ，切除平面となる．

分枝カット法

分枝限定法で子問題の緩和問題を解く際に，切除平面を加えて緩和問題を強化し，早い段階での打ち切りを図る方法が**分枝カット法**.

子問題を解くときに適切な切除平面を加えることが重要であるが，切除平面を見つける手間や追加する妥当不等式の本数と分枝の広がりとのバランスを考慮してアルゴリズムを設計しなくてはいけない.

演習問題

0-1 計画問題

$$
\left|
\begin{array}{lll}
\textbf{最大化} & c_1x_1 + c_2x_2 + \cdots + c_nx_n & \\
\overset{\boldsymbol{x}}{\textbf{条件}} & a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i, & i = 1, 2, \ldots, m \\
 & x_j \in \{0, 1\} & j = 1, 2, \cdots, n
\end{array}
\right.
$$

で

$$
\sum_{j=1}^{n} c_j x_j - (z_1 - z_0)x_u \leq z_0
$$

が妥当不等式であることを示せ．ただし，z_0 は $x_u = 0$ と固定した子問題の最適値，z_1 は $x_u = 1$ と固定した子問題の最適値とする．