整数計画法
Integer Programming (IP)

8.1　整数計画問題

8.2　定式化（Formulation）のテクニック　省略

8.3　緩和法（Relaxation）

8.4　分枝限定法（Branch and Bound）

8.1　整数計画問題

- ナップサック問題
- 巡回セールスマン問題
- 施設配置問題
- スケジューリング問題
- 配送計画問題
- 輸送問題

ナップサック問題
KP (Knapsack Problems)

積載重量制限 4kgの条件下で，最大収益を上げるには？　表8.2 KP問題事例

商品	A	B	C	D
収入 [万円]	3	4	1	2
重量 [kg]	2	3	1	3

max. $\sum_{i=1}^{n} v_i x_i$

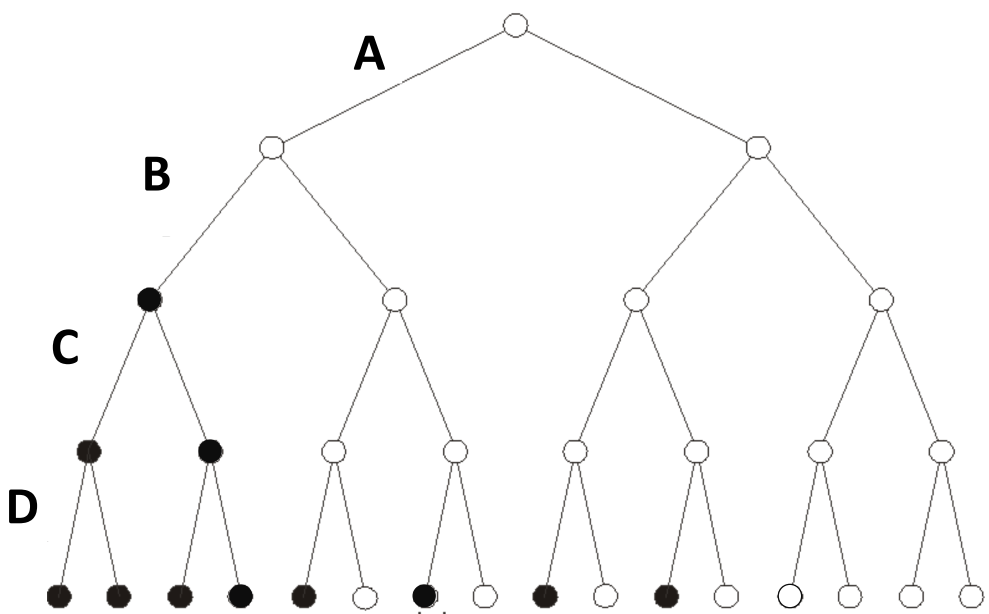
s.t. $\sum_{i=1}^{n} w_i x_i \leq b$

$x_i \in \{0,1\} \quad \forall i \in N$

$x_1, \ldots, x_n = 0 \ or \ 1$

0-1問題

組合せ 2^n 通り　＞　列挙木



<u>厳密解法</u>
完全列挙法：nが増えると時間増
＞分枝限定法

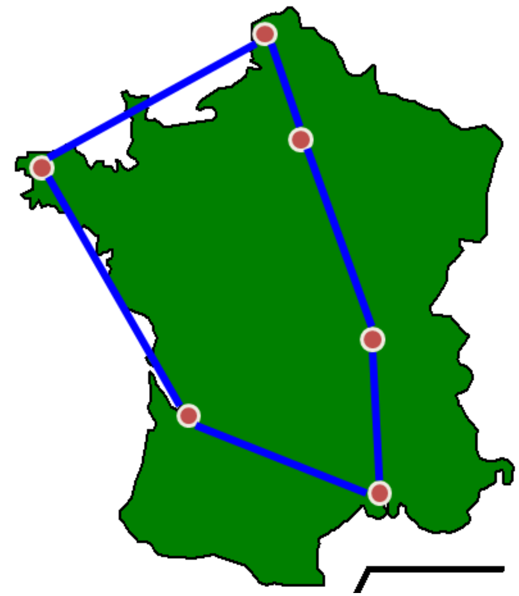
<u>近似解法</u>
＞貪欲法 (Greedy Algorithm)
単位あたり価値の高いものから検索

<u>ヒューリスティック解法（発見的解法）</u>

巡回セールスマン問題
TSP(Traveling Salesman Problems)

セールスマンが全ての都市を1回ずつ通過し，出発地に戻って来る最小移動距離となる訪問経路を決定する.

ハミルトン閉路問題が原型.



6都市のとき，

5! /2= 5 × 4 × 3 × 2 /2 = 60通り.
n都市ならば,（*n* - 1）! /2通り.

＊出発地を固定し，逆回りは同じ経路（対称TSP）.

min. $\sum_{i}^{n}\sum_{j}^{n} w_{ij} x_{ij}$　*i*: 都市，*j*: 訪問順

s.t. $\sum_{i=1}^{n} x_{ij} = 1 \quad \sum_{j=1}^{n} x_{ij} = 1 \quad \sum_{i \in S}^{n} \sum_{j \in V-S}^{n} x_{ij} \geq 1$

$x_i \in \{0,1\} \quad \forall i \in N$

指数爆発

完全列挙法が適用可能なケースはほとんどない．

- コンピューター精度：CPUの性能指標

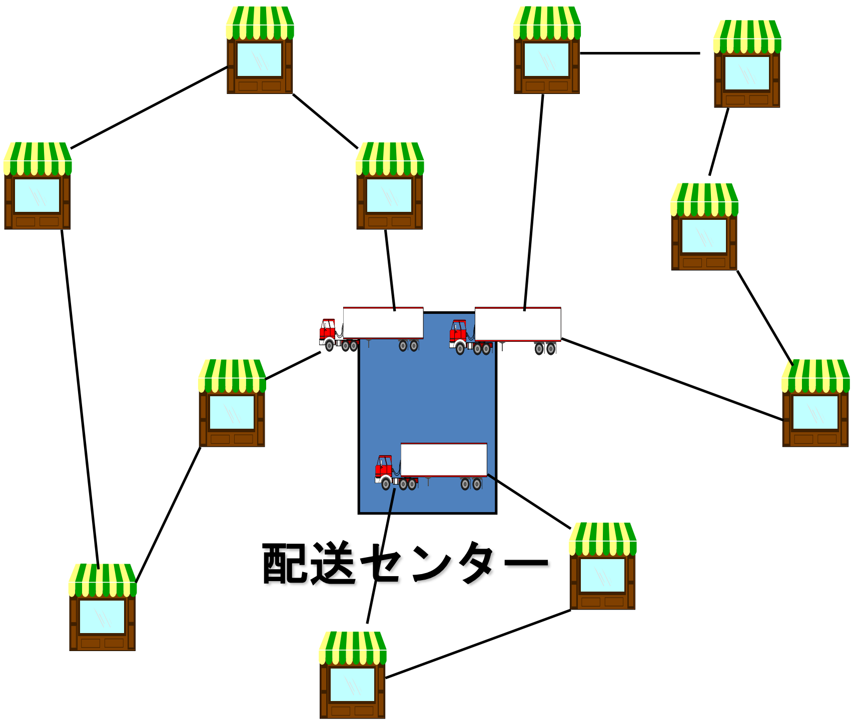
  100MIPS (mega or million instructions per second)

  1秒間に100万回の計算＝1回 に 10^{-6}秒

n	10	100	1,000	10,000
n	10^{-5}秒	10^{-4}秒	10^{-4}秒	0.001秒
n^2	10^{-4}秒	0.01秒	1秒	100秒
n^3	0.001秒	1秒	16.6分	277時間
2^n	0.001秒	10^{14}世紀	10^{284}世紀	
$n!$	0.036秒	10^{141}世紀	10^{2551}世紀	

宇宙の年齢 1.5×10^8 世紀（150億年）

配送計画問題
VRP (Vehicle Routing Problems)



総輸送距離最小化

総車両数最小化

分類

- 0-1整数計画問題 : 0-1 Integer Programming
- 全数整数計画問題 : All (pure) Integer Programming
- 混合整数計画問題 : Mix Integer Programming [MIP]

解法

- 厳密解法 : Exact method　最適解が厳密に保証.
- 近似解法 : Approximation method

  最適解近い解で，精度が保証.
- ヒューリスティック解法 : Heuristic method

  解の精度が保証されていない.

<u>メタヒューリスティック</u>

理論的背景を持つ解法

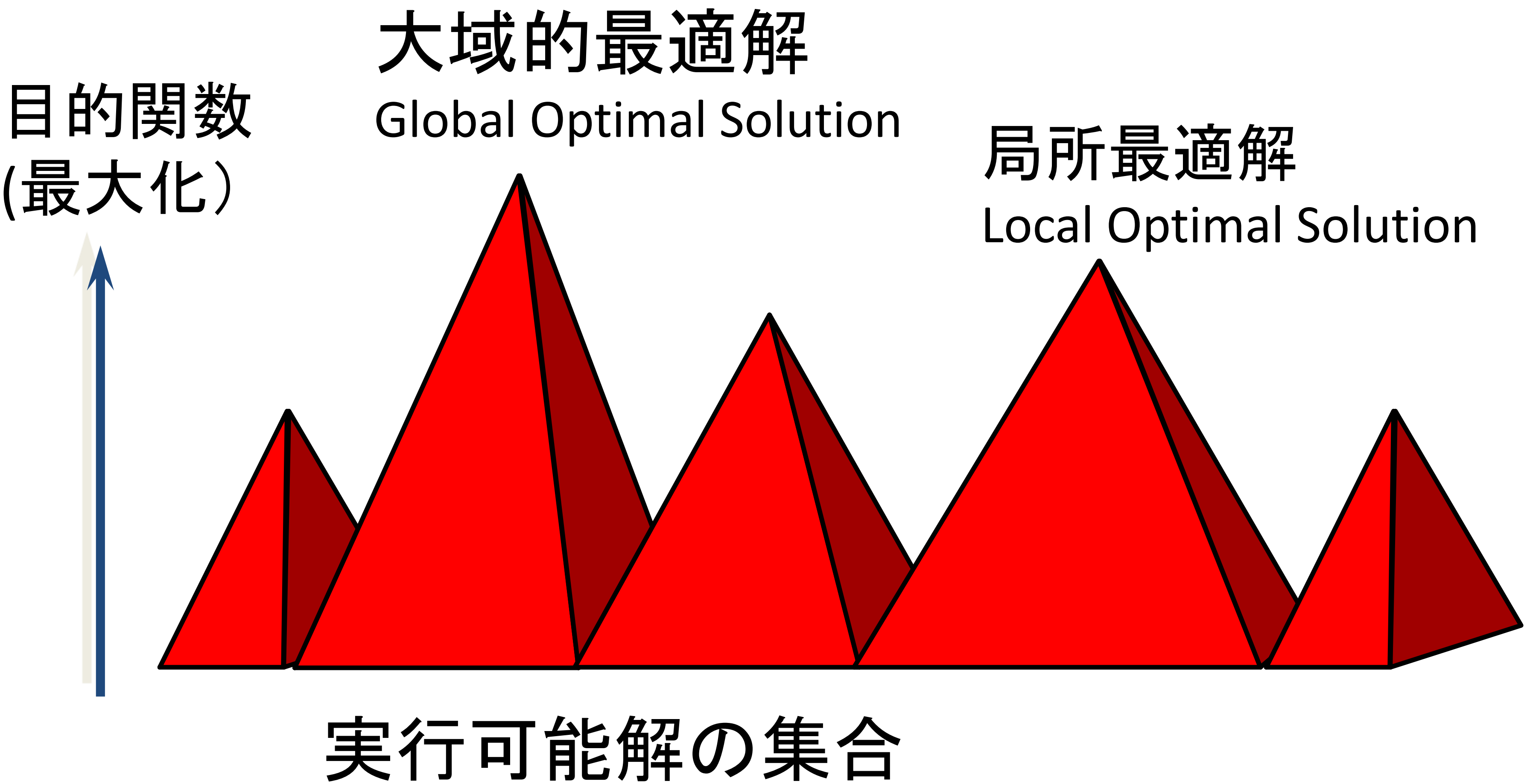
＞アルゴリズム開発

メタヒューリスティクス Metaheuristics

指数爆発の起こる組合せ最適化問題(Combinatorial optimization)について，数理的, 分析的手法に基づく厳密解法に対し, ある暫定解からより良い解を発見的に探索するための方法論. 1) 個々の問題の性質に依拠しないより包括的な枠組, 2) 最適解を求めることではなく, より良い解を現実的な時間で求めることを目的，などの特徴がある.

1. 局所探索法：Local Search
2. 近傍法：Neighborhood Search
3. アニーリング法（焼き鈍し法）：Simulated Annealing
4. タブー探索法：Tabu Search
5. 遺伝的アルゴリズム：Genetic Algorithm
6. 蟻コロニー最適化：Ant Colony Optimization

組合せ最適化問題の概念図



8.3　緩和法の原理

(1)線形緩和

全数整数計画問題

$$\max \quad z = 3x_1 + 2x_2$$

s.t. $2x_1 + x_2 \le 6$

$$x_1, x_2 \in \{1, 0\}$$

整数制約を解除 ⇒

線形緩和問題

$$\max \quad z = 3x_1 + 2x_2$$

s.t. $2x_1 + x_2 \le 6$

$$0 \le x_1, x_2 \le 1$$

線形緩和の最適解　≥　全数整数計画問題の最適解

(2)ラグランジェ緩和

別名　罰金法（ペナルティ法）
制約条件を破る解にペナルティが与えられるように目的関数に制約式を入れ，無制約最適化問題に変換する．ラグランジェ乗数法に発展．

⇒　非線形計画問題に適用するのが一般的

8.4 分枝限定法 (Branch and Bound Method)

$$\max \quad z = 8x_1 + 5x_2$$

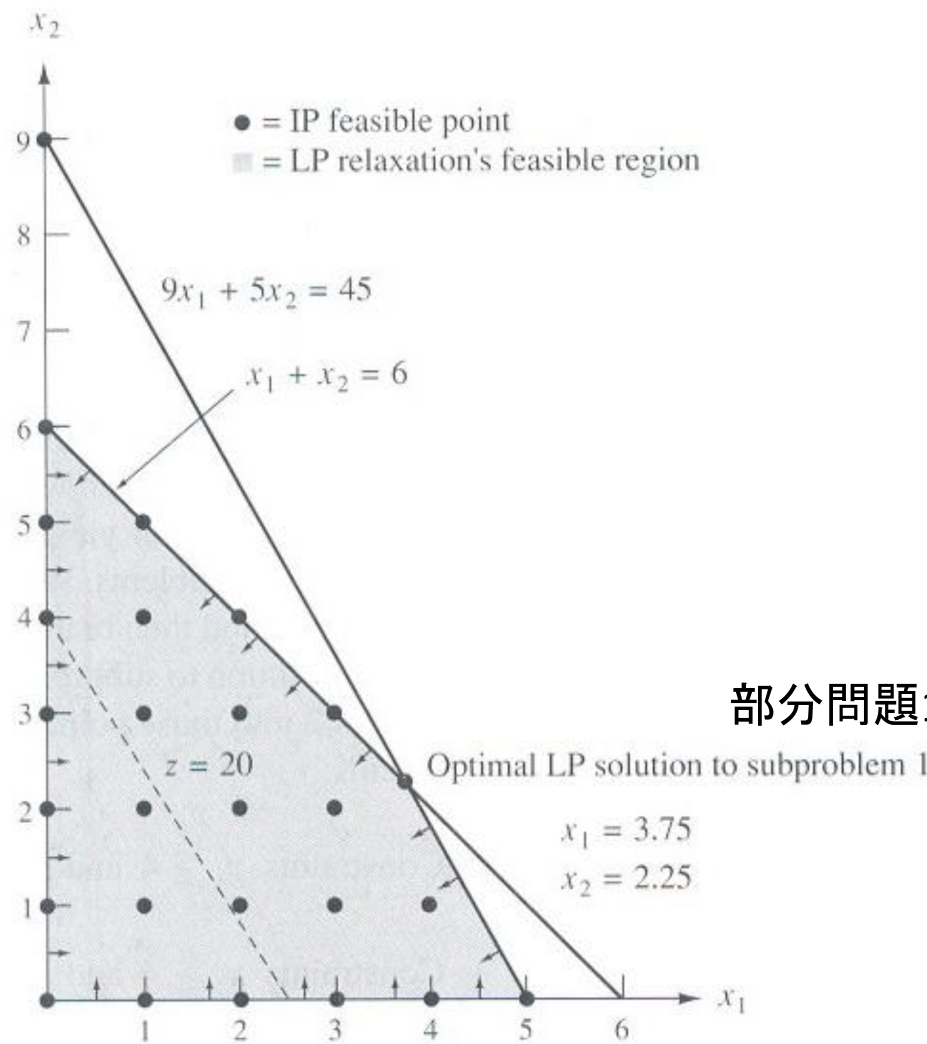
s.t. $x_1 + x_2 \le 6$

$9x_1 + 5x_2 \le 45$

$x_1, x_2 \ge 0, x_1, x_2$ integer

線形緩和最適解 ＝上界値 $z = \dfrac{165}{4} = 41.25,$

$$x_1 = \frac{15}{4}, x_2 = \frac{9}{4}$$



部分問題1

候補解

(0,0), (1,0), (2,0), (3,0), (4,0), (5,0),

(0,1), (1,1), (2,1), (3,1), (4,1), (0,2),

(1,2), (2,2), (3,2), (0,3), (1,3), (2,3),

(3,3), (0,4), (1,4), (2,4), (0,5), (1,5), (0,6)

分枝操作

部分問題の作成

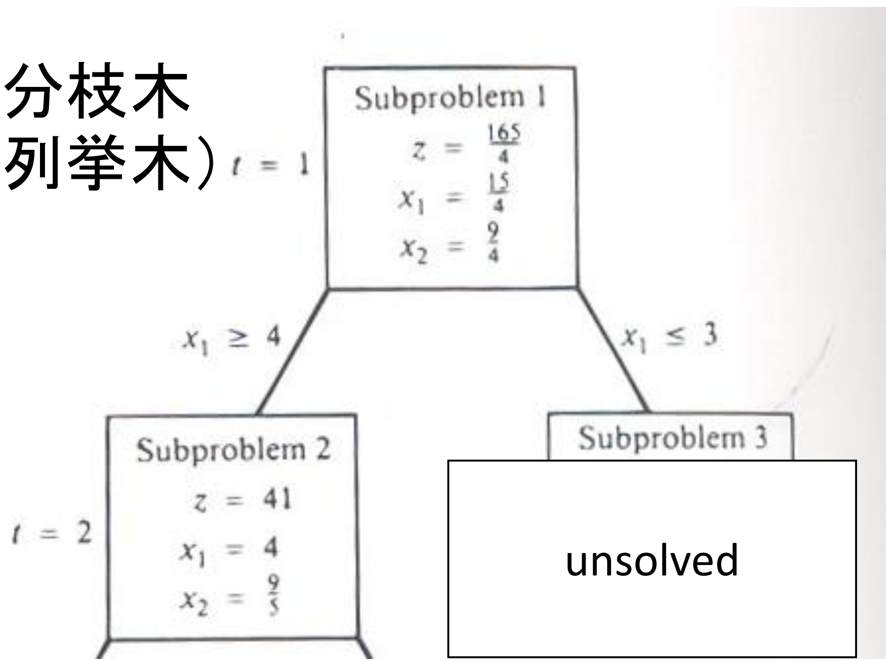
部分問題2

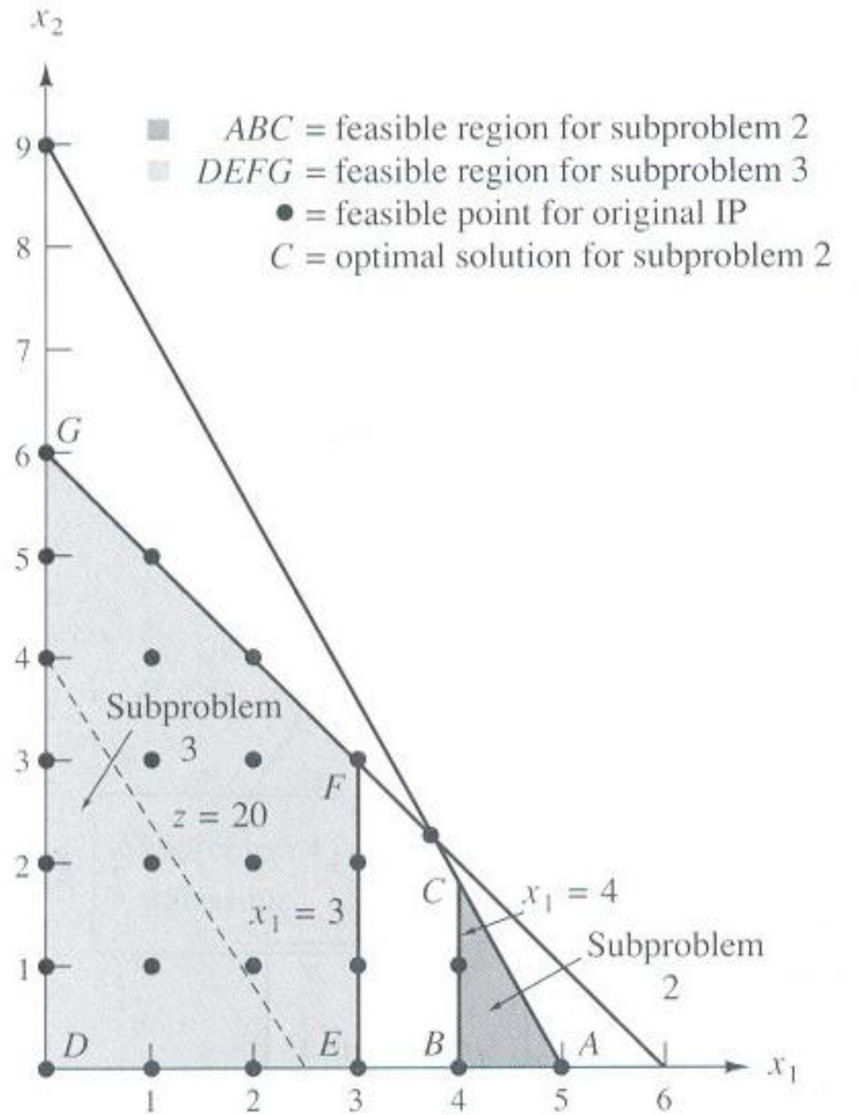
部分問題 1 + 制約条件 $x_1 \geq 4$

部分問題 3

部分問題 1 + 制約条件 $x_1 \leq 3$

分枝木
（列挙木）





部分問題 4

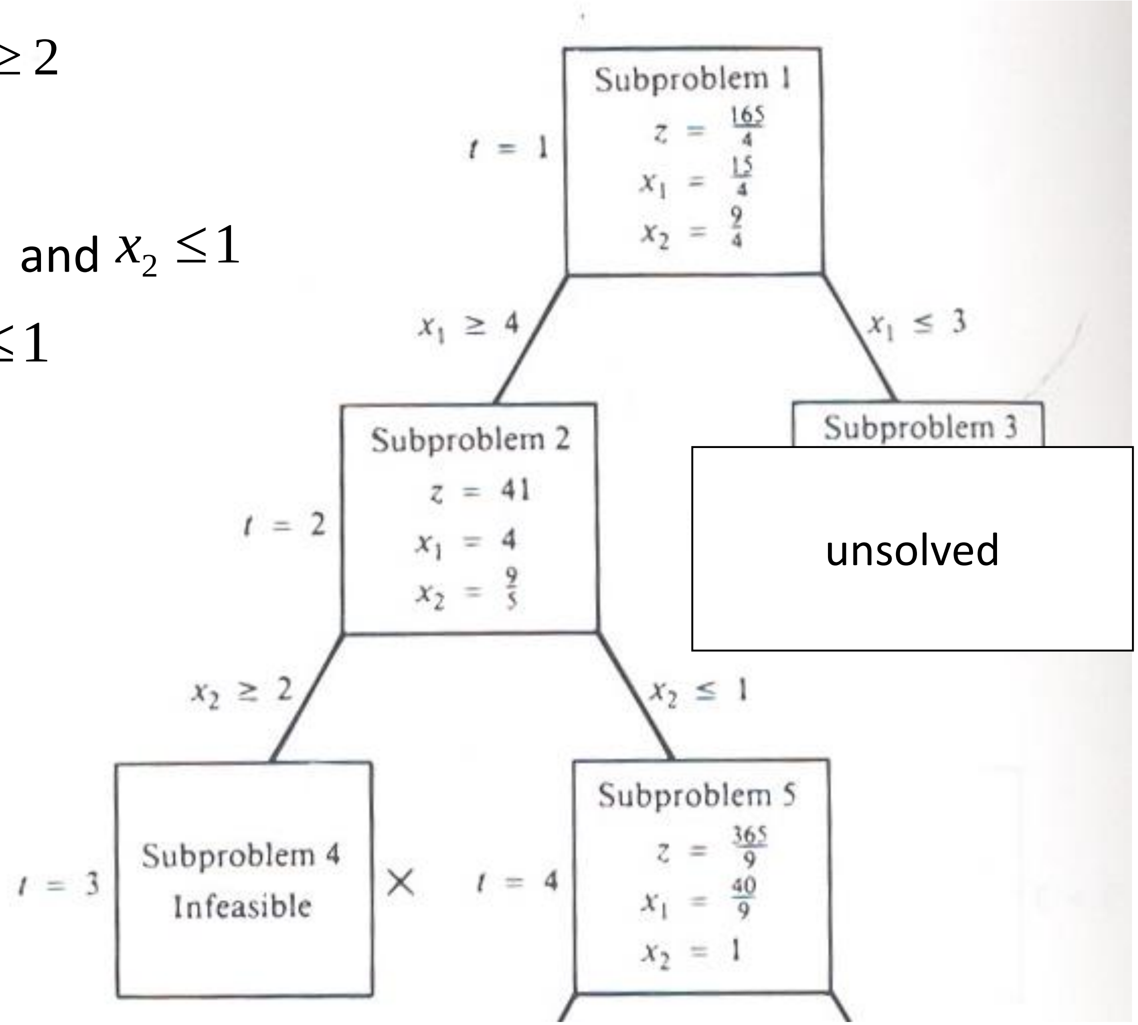
部分問題 1 + 制約条件 $x_1 \ge 4$ and $x_2 \ge 2$

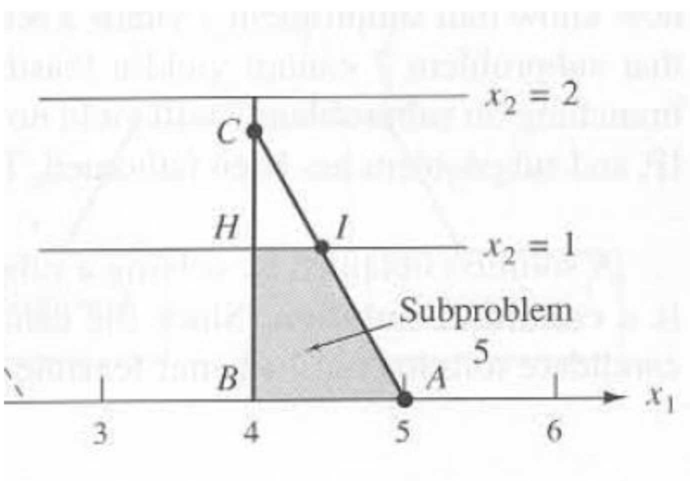
= 部分問題 2 + 制約条件 $x_2 \ge 2$

部分問題5

部分問題 1 + 制約条件 $x_1 \ge 4$ and $x_2 \le 1$

= 部分問題 2 + 制約条件 $x_2 \le 1$





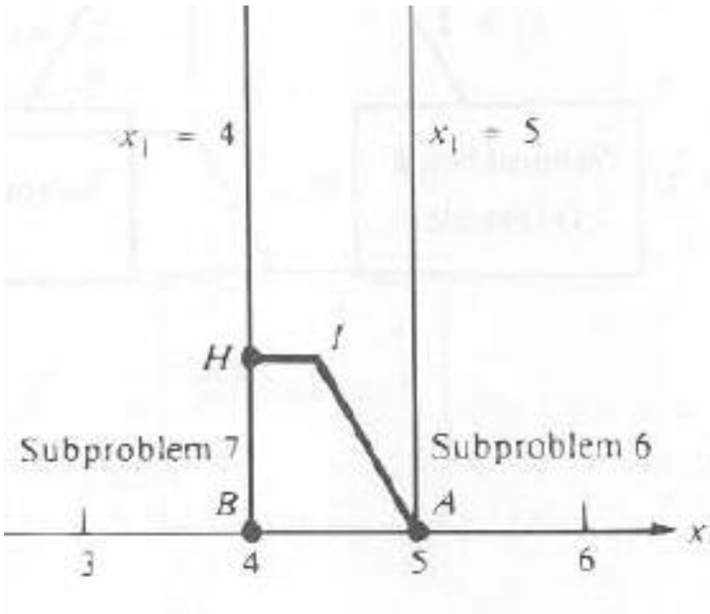
限定操作

部分問題 6　部分問題 5 + 制約条件 $x_1 \geq 5$

部分問題 7　部分問題 5 + 制約条件 $x_1 \leq 4$

候補解　H(4,1), B (4,0), A (5,0)のみ

＞分枝操作終了

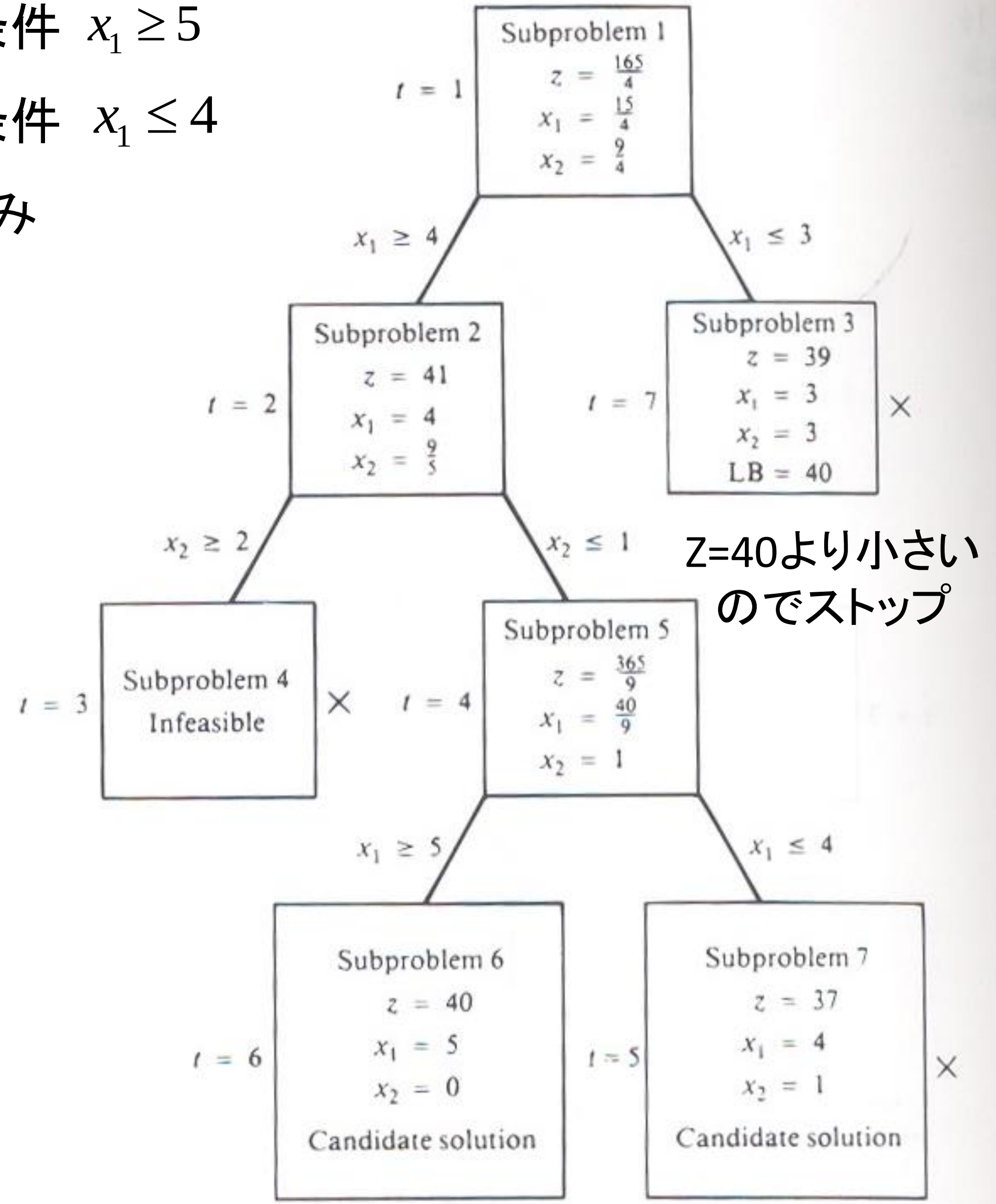


LIFO (last-in-first-out)順に解く

部分問題7　z = 37

<u>部分問題 6　z = 40</u>　最適解

部分問題 3　z = 39



Z=40より小さいのでストップ

下界値（更新される）

分枝限定法のプロセス

1) 部分問題の作成（分枝操作）

線形緩和問題の最適解で非整数となった変数を対象

2) 非許容解を排除（限定操作）

3) 許容下界値を見つける

線形緩和問題の最適解がすべて整数解

4) LIFO順に解く

5) （最大化の場合）下界値を超える解がなければ終了

混合整数計画問題における分枝限定法

整数制約条件のある変数のみ分枝操作する.

$$\max \quad z = 2x_1 + x_2$$

$$\text{s.t.} \quad 5x_1 + 2x_2 \le 8$$

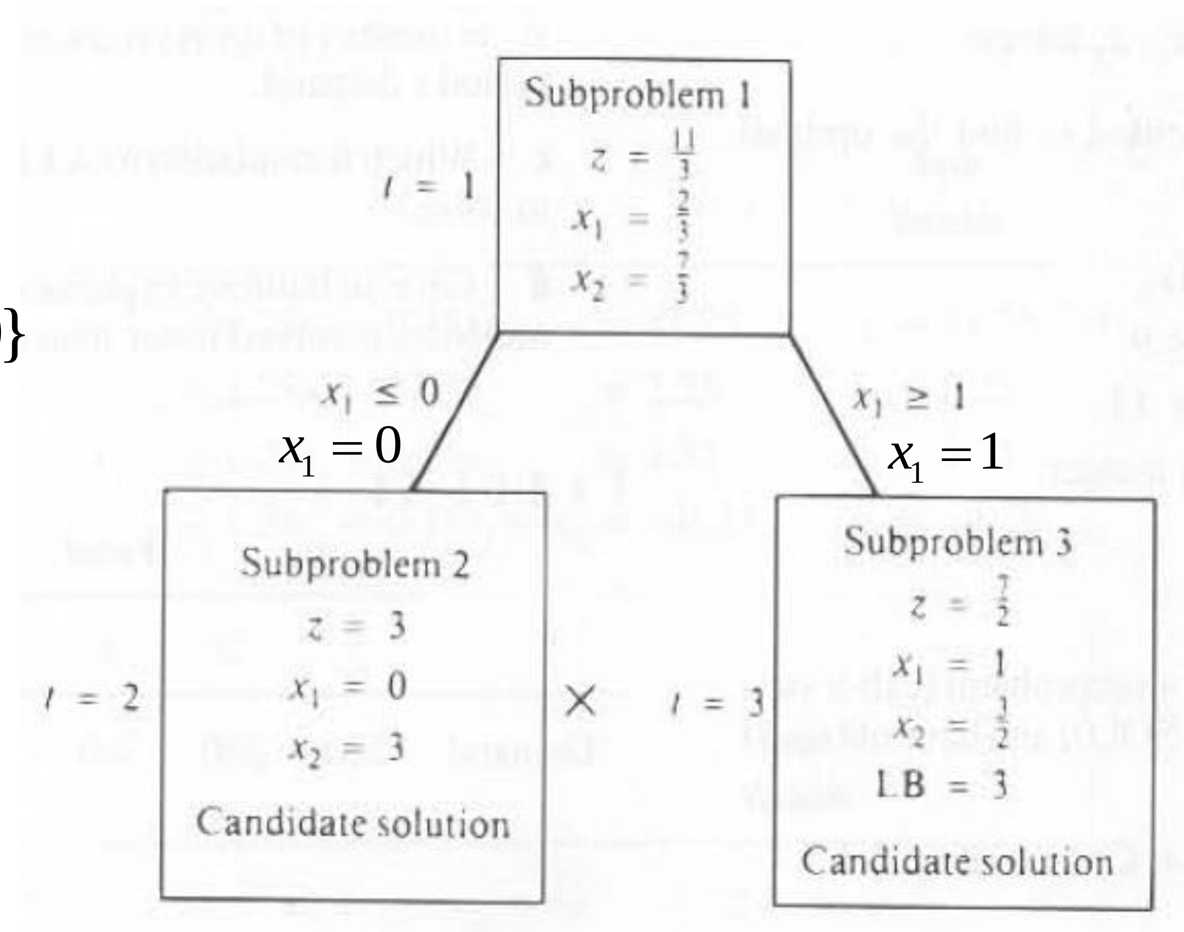
$$x_1 + x_2 \le 3$$

$$x_1, x_2 \ge 0 : x_1 \in \{1, 0\}$$

線形緩和最適解

$$z = \frac{11}{3}, x_1 = \frac{2}{3}, x_2 = \frac{7}{3}$$

x_1のみ分枝



分枝限定法：ナップサック問題

重量制限 4kgの条件下で，最大収益を上げるには？

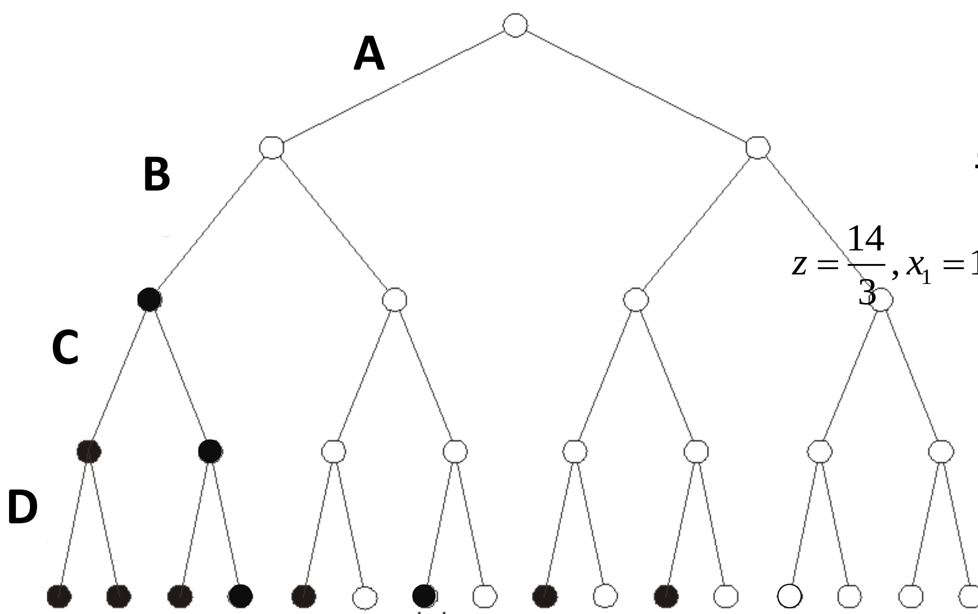
商品	A	B	C	D
収入 [万円]	3	4	1	2
重量 [kg]	2	3	1	3

$$\text{max.}\quad z = 3x_1 + 4x_2 + x_3 + 2x_4$$

$$\text{s.t.}\quad 2x_1 + 3x_2 + x_3 + 3x_4 \le 4$$

$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

組合せ 2^n 通り ＞ 分枝木（列挙木）



線形緩和最適解 $0 \le x_i \le 1$

$$z = \frac{17}{3}, x_1 = 1, x_2 = \frac{2}{3}, x_3 = 0, x_4 = 0$$

部分問題1 $x_2 = 0$

$$z = \frac{14}{3}, x_1 = 1, x_2 = 0, x_3 = 1, x_4 = \frac{1}{3}$$

部分問題2 $x_2 = 1$

$$z = \frac{11}{2}, x_1 = \frac{1}{2}, x_2 = 1, x_3 = 0, x_4 = 0$$

部分問題3 $x_4 = 0$

$z = 4$

部分問題4 $x_4 = 1$

$z = 7/2$

部分問題5 $x_1 = 0$

$z = 5$

部分問題6 $x_1 = 1$

非許容解