4.3 分枝限定法

しらみつぶしに実行可能解について調べるために場合分けを行う際に，不必要な場合分けを省略して計算量を削減する方法を分枝限定法という．ここでは，ナップザック問題を例にして分枝限定法を説明する．

まず，前節の問題：

$$\begin{aligned} \text{目的関数} &: 7x_1 + 8x_2 + x_3 + 2x_4 \rightarrow \text{最大化} \\ \text{制約条件} &: 4x_1 + 5x_2 + x_3 + 3x_4 \leq 6 \\ & x_i \in \{0, 1\} \end{aligned}$$

に対して，

$$\begin{aligned} \text{目的関数} &: 7x_1 + 8x_2 + x_3 + 2x_4 \rightarrow \text{最大化} \\ \text{制約条件} &: 4x_1 + 5x_2 + x_3 + 3x_4 \leq 6 \\ & 0 \leq x_i \leq 1 \ (1 \leq i \leq n) \end{aligned}$$

という問題を考える．この問題は，元の問題の 0-1 条件を，_____ でよいという条件に緩めた問題である．このように作った問題を，_____ という．この問題は，4.2.2 節で説明した欲張り法を用いて最適解を求めることができる．つまり，最適解は $\mathbf{x} = (1, 2/5, 0, 0)$ となる．これを，もとのナップザック問題の _____ という．

実数最適解の性質：

- [0-1 問題の実行可能領域] ___ [連続緩和問題の実行可能領域]
  であるから，
  [0-1 問題の最大値] ___ [連続緩和問題の最大値]
  となっている．つまり，実数最適解における目的関数の値は，0-1 問題における最大値の**上界**となる．

- 実数最適解の変数が全て 0-1 条件を満たしている場合，これは 0-1 問題の最適解である．

- 連続緩和問題が実行可能解を持たないとき，0-1 問題は_____

さて，0-1 問題の最適解は 2^n 個の実行可能解の中に存在する．それらを全て調べ尽くすには，図 4.3 に示すような**分枝図**を使うとよい．

- 図中の一番上の点は元の 0-1 問題を示す．

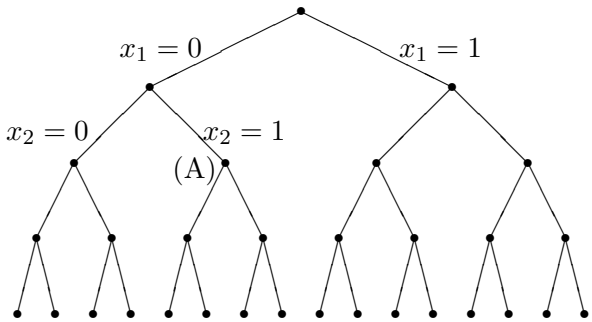

図 4.3: 分枝図

- ２段目の２つの点は x_1 をそれぞれ０か１に固定した場合の問題を示す．同様に３，４段目の点は x_1, x_2 および x_1, x_2, x_3 を固定して残りの変数についての問題を示す．
- 最下段の点は全ての変数を固定した状態を示す．つまり，これらの点の中で実行可能解を示すものについて，目的関数の値を計算して比べることによって，最適解を求めることができる．

2,3,4 段目の点が示す問題を，元の問題の _____ と呼び，0 に固定されている変数の添え字の集合 J_0 と，1 に固定されている変数の添え字の集合 J_1 によって $P(J_0, J_1)$ と書く．たとえば，図中の (A) の点は $J_0 = \{1\}, J_1 = \{2\}$ に対応し，上に示したナップザック問題の例に対する部分問題 $P(\{1\}, \{2\})$ は

$$\begin{aligned} \text{目的関数} &: 8 + x_3 + 2x_4 \rightarrow \text{最大化} \\ \text{制約条件} &: 5 + x_3 + 3x_4 \leq 6 \\ &\quad x_i \in \{0, 1\} \end{aligned}$$

と書ける．この問題も，ナップザック問題となっていることに注意．

分枝限定法でナップザック問題を解くには，

1. まず根にあたる元の問題 $P(\emptyset, \emptyset)$ の連続緩和問題を解いてみて，実数最適解を求める．実数最適解が 0-1 条件を満たしているならば最適解となっているので終了．そうでなければ，(欲張り法などを用いて）近似最適解を求め，暫定解とする．また，そのときの目的関数の値を暫定値とする．
2. １つの変数を 0 および 1 に固定した部分問題 $P(\{1\}, \emptyset), P(\emptyset, \{1\})$ を作成する．
3. 部分問題の中の１つ $P(J_0, J_1)$ を選び，その問題の連続緩和問題を解く．その結果により，以下の４つの場合が考えられる．

(a) **連続緩和問題に実行可能解が存在しない場合**: 部分問題 $P(J_0, J_1)$ にも実行可能解が存在しない.

(b) **連続緩和問題によって得られた実数最適解が 0-1 条件を満たす場合**: 実数最適解は $P(J_0, J_1)$ の最適解となる．最適解の目的関数の値が暫定値より大きいならば，その解はより良い暫定解とすることができる.

(c) **連続緩和問題によって得られた $P(J_0, J_1)$ の上界値が，現在の暫定値よりも小さい場合**: $P(J_0, J_1)$ の実行可能解の中には元の問題の最適解が存在することはない.

(d) **連続緩和問題によって得られた $P(J_0, J_1)$ の上界値が，現在の暫定値よりも大きく，実数最適解が 0-1 条件を満たさない場合**: $P(J_0, J_1)$ の実行可能解の中に最適解が存在する可能性がある.

(a)(c) では，注目した部分問題の実行可能解の中には元の問題の最適解が存在することはないことがわかるので，これ以上この部分問題を調べて厳密に解くことをやめる．また，(b) では注目した部分問題を厳密に解くことができたのでさらに部分問題を調べる必要がない．このように，さらなる部分問題を調べないでおわることを，部分問題の _____ という．(d) の場合は，注目した部分問題をさらに厳密に解く必要がある．そこで，$P(J_0, J_1)$ の自由変数を 1 つ選んで 0 または 1 に固定することによって 2 つの部分問題を新たに追加する．これを分枝操作という.

4. 全ての部分問題が終端してしまったら終了．そうでなければ 3. に戻る.

ある時点で，終端されていない部分問題を _____ といい，その集合を $\mathcal{A}$ で表す．具体例をあげよう．先に示した問題例を考える.

$$\begin{aligned} \text{目的関数} &: 7x_1 + 8x_2 + x_3 + 2x_4 \rightarrow \text{最大化} \\ \text{制約条件} &: 4x_1 + 5x_2 + x_3 + 3x_4 \leq 6 \\ &\quad x_i \in \{0, 1\} \end{aligned}$$

(0) 近似最適解は欲張り法を用いて _____ とする．つまり，暫定解は _____ で暫定値は ____ ． 2 つの部分問題 _____, _____ を作る．$\mathcal{A} =$_____ である.

(1) 活性部分問題 $P(\{1\}, \emptyset)$ を選ぶ．これは，$x_1 = 0$ と置くことで，

$$\begin{aligned} \text{目的関数} &: \text{\_\_\_\_\_} \rightarrow \text{最大化} \\ \text{制約条件} &: \text{\_\_\_\_\_} \\ &\quad x_i \in \{0, 1\} \end{aligned}$$

となる．この問題の連続緩和問題を解くと（欲張り法），実数最適解は $(x_2, x_3, x_4) =$_____ となり，0-1 条件を満たしている．したがって，上の ____ の場合である．$P(\{1\}, \emptyset)$ における目的関数の最大値は 9 となり，現在の暫定値より大きいので，$(0, 1, 1, 0)$ を暫定解とし，9 を暫定値とする．$P(\{1\}, \emptyset)$ は終端できるので，$\mathcal{A} = \{P(\emptyset, \{1\})\}$ となる．

(2) 活性部分問題 $P(\emptyset, \{1\})$ を選ぶ．これは，

$$\begin{aligned} \text{目的関数} \quad &: \quad \_\_\_\_\_ \rightarrow \text{ 最大化} \\ \text{制約条件} \quad &: \quad \_\_\_\_\_ \\ & \quad x_i \in \{0, 1\} \end{aligned}$$

となる．この問題の連続緩和問題を解くと，実数最適解は $(x_2, x_3, x_4) =$ _____ となり，上界は _____ である．したがって，上の ____ の場合である．$P(\emptyset, \{1\})$ の部分問題によって最適解が得られる可能性があるので，部分問題 _____ , _____ を作る．$\mathcal{A} =$ _____ となる．

(3) $P(\{2\}, \{1\})$ を選ぶ．これは

$$\begin{aligned} \text{目的関数} \quad &: \quad \_\_\_\_\_ \rightarrow \text{ 最大化} \\ \text{制約条件} \quad &: \quad \_\_\_\_\_ \\ & \quad x_i \in \{0, 1\} \end{aligned}$$

である．実数最適解は $(x_3, x_4) =$_____ となり，上界は _____ である．したがって，上の ____ の場合であり，最適解が得られる可能性はないので終端．$\mathcal{A} = \{P(\emptyset, \{1, 2\})\}$.

(4) $P(\emptyset, \{1, 2\})$ を選ぶ.

$$\begin{aligned} \text{目的関数} \quad &: \quad 7 + 8 + x_3 + 2x_4 \rightarrow \text{ 最大化} \\ \text{制約条件} \quad &: \quad 4 + 5 + x_3 + 3x_4 \leq 6 \\ & \quad x_i \in \{0, 1\} \end{aligned}$$

となる．この問題は実行可能解を持たない．したがって，上の ____ の場合であり終端.

(5) 活性部分問題がなくなったので，現在の暫定解 $(0, 1, 1, 0)$ を最適解として終了.

分枝限定法では，基本的には全ての解を列挙して目的関数の値を調べていることと等しいので，かならず最適解を求めることができる．

ここで取り上げたナップザック問題の場合，分枝操作によって必ず２つの部分問題が生成されたが，一般には３つ以上の部分問題が生成されることもある．また，ここでは各部分問題の目的関数の上界は実数最適解によって求めたが，計算量を削減するためには与えられた問題ごとにいろいろな工夫が必要.

- 効率的な上界値の計算方法
- 活性部分問題の探索法（複数ある活性部分問題のうち，どれを先に選択するか？）.
  深さ優先探索法 $\cdots$ 元の問題から見て，最も深いものを選ぶ.
  最良優先探索法 $\cdots$ 各活性部分問題における目的関数の上界値の推定値を求め，それが最大のものを選ぶ.

4.4 動的計画法

最適性の原理：ある問題と，それを分解した部分問題を考えると，元の問題の最適解は，部分問題の最適解を含んでいる.

最適性の原理を利用して組合せ最適化問題をとく手法を，**動的計画法**と呼ぶ.（動的計画法も列挙法の１つであり，必ず最適解が求まる.）動的計画法の例として，最短路問題の解法のひとつであるダイクストラ法がある．ダイクストラ法では，最適性の原理として，「全体のネットワークにおける最短路木を T とすると，T から最も遠い点 P を除いた木 T' は，ネットワークから点 P を除いたネットワークにおける最短路木である」という性質を利用している．したがって，全体のネットワークにおける最短路木 T を求めるためには，T' を求めればよく，それを求めるためにはその中で最も遠い点 P' を除いたネットワークにおける最短路木 T'' を求めればよく,... と，再帰的になり，最も近い点から順にネットワークを大きくしていけばよいことが分る.

動的計画法の分りやすい例として，ここでは資源配分問題を挙げる.

[**問題 4.3**] [資源配分問題] 余剰資金を３つの投資対象に投資したい．各投資先に投資したときに見込まれる利益は下表のとおりである．資金が５単位あるとき，どのように投資したときに最も利益を挙げることができるか.

少し一般化して考えよう．投資先の個数を n 個，余剰資金を Z，投資先 i に x_i 単位投資したときの利益を関数 $f_i(x_i)$ で表わすことにすると，この問題は以下のように定式化される.

目的関数：　＿＿＿＿＿＿＿＿＿＿ → 最大化

制約条件：　＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿

表 4.1: 利益

投資額（単位）	1	2	3	4	5
投資先 1	2	4	6	8	10
投資先 2	1	2	5	9	11
投資先 3	3	4	5	6	7

ここで，Z 単位の余剰資金を投資したときに得られる最大利益を $F_n(Z)$ で表わすことにすると，

$$F_n(Z) = \max_{0\le x_n\le Z}(f_n(x_n) + F_{n-1}(Z - x_n)) \quad n > 1 \text{ のとき}$$
$$F_1(Z) = \max_{0\le x_1\le Z} f_1(x_1) \quad n = 1 \text{ のとき}$$

が成り立つ. そして, 与えられた問題は_____ とそれを実現する (x_1, x_2, x_3) _____ を求めることに他ならない．したがって，F_1, F_2, F_3 の順に決定していけば最適解を求めることができる．ここでは，各 F_i の決定には，１つの変数 x_i しか用いられていないことに注意.

実際に，上の問題を解いてみよう.

(1) $0 \le x_1 \le 5$ より，$F_1(0), \ldots, F_1(5)$ を求める．$f_1(x_1)$ は単調増加であるから，

$$F_1(Z) = \max_{0\le x_1\le Z} f_1(x_1) = f_1(Z).$$

したがって，$F_1(0) = 0, F_1(1) = 2, F_1(2) = 4, F_1(3) = 6, F_1(4) = 8, F_1(5) = 10.$

(2) $F_2(Z) = \max_{0\le x_2\le Z}(f_2(x_2) + F_1(Z - x_2))$ を利用して $F_2(0), \ldots, F_2(5)$ を求める.

$$\begin{aligned}
F_2(0) &= f_2(0) + F_1(0) = 0 \\
F_2(1) &= \max(f_2(0) + F_1(1), f_2(1) + F_1(0)) \\
&= \max(\underline{0+2}, 1+0) = 2 \\
F_2(2) &= \max(f_2(0) + F_1(2), f_2(1) + F_1(1), f_2(2) + F_1(0)) \\
&= \max(\underline{0+4}, 1+2, 2+0) = 4 \\
F_2(3) &= \max(f_2(0) + F_1(3), f_2(1) + F_1(2), f_2(2) + F_1(1), f_2(3) + F_1(0)) \\
&= \max(\underline{0+6}, 1+4, 2+2, 5+0) = 6 \\
F_2(4) &= \max(f_2(0) + F_1(4), f_2(1) + F_1(3), f_2(2) + F_1(2), f_2(3) + F_1(1), f_2(4) + F_1(0)) \\
&= \max(_____, _____, _____, _____, _____) = __ \\
F_2(5) &= \max(_____) \\
&= \max(_____) = __
\end{aligned}$$

(3) $F_3(Z) = \max_{0 \le x_3 \le Z}(f_3(x_3) + F_2(Z - x_3))$ を利用して $F_3(0), \ldots, F_3(5)$ を求める．

$$
\begin{aligned}
F_3(0) &= \underline{\qquad\qquad\qquad\qquad} \\
F_3(1) &= \underline{\qquad\qquad\qquad\qquad\qquad\qquad} \\
&= \underline{\qquad\qquad\qquad\qquad\qquad\qquad} \\
F_3(2) &= \max(f_3(0) + F_2(2), f_3(1) + F_2(1), f_3(2) + F_2(0)) \\
&= \max(0 + 4, \underline{3 + 2}, 4 + 0) = 5 \\
F_3(3) &= \max(f_3(0) + F_2(3), f_3(1) + F_2(2), f_3(2) + F_2(1), f_3(3) + F_2(0)) \\
&= \max(0 + 6, \underline{3 + 4}, 4 + 2, 5 + 0) = 7 \\
F_3(4) &= \max(f_3(0) + F_2(4), f_3(1) + F_2(3), f_3(2) + F_2(2), f_3(3) + F_2(1), f_3(4) + F_2(0)) \\
&= \max(\underline{0 + 9}, \underline{3 + 6}, 4 + 4, 5 + 2, 6 + 0) = 9 \\
F_3(5) &= \max(f_3(0) + F_2(5), f_3(1) + F_2(4), f_3(2) + F_2(3), f_3(3) + F_2(2), f_3(4) + F_2(1), \\
&\qquad f_3(5) + F_2(0)) \\
&= \max(0 + 11, \underline{3 + 9}, 4 + 6, 5 + 4, 6 + 2, 7 + 0) = 12
\end{aligned}
$$

(4) F_3 の最大値は $F_3(5) = 12$ で，このとき $x_3 = 1, F_2(4) = 9$．$F_2(4) = 9$ となるのは，$x_2 =$_, _____ のとき．$F_1(0) = 0$ となるのは，$x_1 = 0$ のとき．

したがって，$(x_1, x_2, x_3) = (0, 4, 1)$ が最適解で，最大利益は 12.

4.5 近似解法

組合せ最適化問題の中には，厳密に最適解を求めるためには指数時間かかるものが多い．

どのくらいの時間で問題を解くことができるかによって，問題を大まかに分けることができる．

クラス P: 多項式時間で答え（yes か no か）が求まる問題からなる集合．

クラス NP: 答え（yes か no か）を求めるのに指数時間かかるが，答えが yes である証拠が与えられているときに，yes であることを検証するのは多項式時間でできる問題からなる集合．

クラス NP 困難: クラス NP に含まれるどの問題よりも難しいか，同等に難しい問題からなる集合．（例：巡回セールスマン問題）

NP 困難な問題のように，厳密に最適な解を求めるのに非常に時間がかかる（たとえば，高性能コンピュータ 100 台を 10 億年走らせてやっと答えが出る）問題でも，近似最適解で

あれば現実的な時間で解くことができる場合もある．近似最適解を求める解法は，**ヒューリスティック解法（発見的解法）**と呼ばれている．

欲張り法では一般には最適解を求めることはできないが，近似最適解を求めることはできる．ここでは，巡回セールスマン問題を例に取り，欲張り法の考え方を使ったいくつかのヒューリスティック解法を紹介する．

[**問題 4.4**] [巡回セールスマン問題] あるネットワーク $G = (V, E)$,$V = \{1, 2, \ldots, n\}$ において，全ての節点をちょうど一回づつ通る巡回路のうち，もっとも長さの短いものを求めよ．ただし，各枝 (i, j) には枝の長さ a_{ij} が与えられているとする．（ここでは，無向グラフを扱う．）

例として，図 4.4 を解いてみよう（$n = 4$ である）．

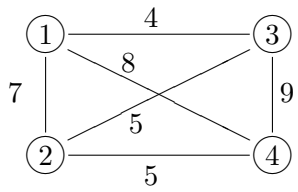

図 4.4: 巡回セールスマン問題の例

最近近傍法：
ある節点から出発して，まだ訪れていない節点のうち現在の節点から最も近い節点へ移動して行き，閉路を求めるやり方．たとえば，図 4.4 で，出発点を節点 1 とすると，$(1 \to 3 \to 2 \to 4 \to 1)$ が得られる．出発点を節点 3 とした場合は，_____ が得られる．

挿入法：
小さな巡回路を考え，巡回路に含まれない節点を順々に巡回路に組み入れて，全節点をとおる巡回路を求めるやり方．組み込む節点の選び方によって，**最近挿入法**，**最遠挿入法**，**最廉挿入法**がある．

たとえば，最近挿入法の具体的手順は以下のとおり．ただし，巡回路 R と節点 i の距離を

$$d(R, i) = \min_{j \in R} a_{ij}$$

で定義する．（a_{ij} は枝 e_{ij} の長さ．）

1. 任意に節点 i_0 を選び，$R = \{i_0\}$ とおく．
2. $d(R, i)$ が最小となる $i \notin R$ を選ぶ．

3. $a_{ij} = d(R, i)$ となる $j \in R$ に対して，節点 j の直前に i を経由する巡回路 $R \leftarrow R \cup \{i\}$ を作る．

4. R が全ての節点を含んでいれば終了．そうでなければ 2. にもどる．

ただし，R 内の節点数が 3 未満のうちは，R を特に巡回路とは見なさずに作業をする．前回と同じネットワークの例では，

(1) $i_0 = 1$ とする．

(2) R に最も近い節点として $i = 3$ を選び，$R = \{1, 3\}$ とする．

(3) R に最も近い節点として $i = 2$ を選ぶ．$R = \{1, 3, 2\}$ とする．

(4) $i = 4$ を選ぶ．最も近いのは節点 2 なので，2 の前に挿入し $R = \{1, 3, 4, 2\}$ とする．

となる．

最遠挿入法では，R に追加する節点として _____ 節点を選択する．また，最廉挿入法では，_____ の長さが最小である節点を選ぶ．

4.6 メタヒューリスティックス

NP 困難な問題であっても，ヒューリスティック解法によって近似最適解を効率的に求めることができる．また，近似最適解に対して修正を加えることによって，よりよい近似最適解を得ることができる問題もある．このように，より良い近似最適解を求める戦略を，**メタヒューリスティックス**と呼ぶ．

ここでは，いくつかの代表的な方法の概要のみを説明する．

局所探索法： 与えられた近似最適解の近くにある実行可能解の中で，目的関数が現在より小さくなる解を選び，次の最適解とする．近くにある解の中で，現在の解が最もよければ終了．

現在の解の「近くにある解」を _____ という．局所探索法では，
・近傍の定義のしかた
・複数ある近傍の中で，次の解をどのように選ぶか
等によって，計算量や近似最適解の善し悪しが変わってくる．

欠点は _____

局所探索法の欠点を補うためには，初期解を変化させて何度も局所探索法を繰り返す等の手段が取られるが，別の考え方として，多少の 改悪 も認めて近似をよくして行く方法もある．

焼きなまし法： 各繰り返しにおいて，近傍からランダムに解を選ぶ．改良になるのなら即，その解を次の近似最適解とし，改悪になる場合は,「改悪」の大きさに応じて確率的にそれを次の近似最適解にするかどうかを決める．

しかし，このままだといつまでも収束しないので，繰り返しの初期段階では，大きな改悪も高い確率で起こるようにし，徐々に改悪の確率を下げていく．改悪となる解を選択する確率を**温度**と考え，徐々に温度を下げていく．

タブー探索法： 局所探索法において，局所的最適解に落ち込んでしまうのを避けるために，近傍の中で現在より改良となる解が見つからない場合は，最も改悪の量が少ない解に移動する．

しかしこうすると，幾つかの解が繰り返し現れてエンドレスになる可能性があるので，すでに近似最適解として選ばれた解のリストを保存しておき，未だ選んでいない解の中で最もよい解へと移動する．

すでに選ばれていてもう選ぶことのできない解のリストを _____ と呼ぶ．このリストは無限に大きくなりうるため，_____ _____.

この他，**遺伝的アルゴリズム**や**ニューラルネットワーク**といった手法もある．

メタヒューリスティックスは，様々な問題に対しする解法となりうる．しかし，パラメータをどのように選ぶかによって計算の効率がまったく変わってくる．現在のところ，どのようにパラメータを決定したら効率がよいかについて，理論的に解明されている部分は少ない．したがって，与えられた問題に対する適切なパラメータの値は，算実験を繰り返すことによって求める必要がある．

4.7 まとめ

[**課題 4.1**] **欲張り法**の特徴，長所，短所を説明せよ．

[**課題 4.2**] 0-1 問題に対する**連続緩和問題**とはどういうものか，また，0-1 問題と連続緩和問題の最適解の関係はどのようになっているか，説明せよ．

[**課題 4.3**] 連続緩和問題を利用した組合せ最適化問題の解き方に，問題を**部分問題**に分解し，最適解を持たない部分問題を終端させることによって，しらみつぶしに解くよりも効率的に最適解を探索する方法がある．これを何法というか．

[**課題 4.4**] ダイクストラ法のように，問題の最適解が，小さい問題の最適解となっていることを利用した解き方を何法というか．

[**課題 4.5**] ヒューリスティック解法とはどのようなものか．また，メタヒューリスティック解法とはどのようなものか．