12. 緩和法と分枝限定法

1

12．1 数理計画法としての定式化

これまでは、問題を判定問題や言語として捉えてきた。ここでは、数理計画法として問題の定式化を行う。数理計画法は、現実の問題を解くための重要な技術である。

数理計画法では、通常評価関数（目標関数）$f : D \to \mathbb{R}$ の最大化問題かあるいは最小化問題として捉える。したがって、一般形は次のように与えられる。

2

最小化問題

最小化 $f(\boldsymbol{x})$

条件 $g_i(\boldsymbol{x}) = b_i \quad i = 1,2,\cdots,k$

$h_j(\boldsymbol{x}) \geq c_j \quad j = 1,2,\cdots,l$

最大化問題

最大化 $f(\boldsymbol{x})$

条件 $g_i(\boldsymbol{x}) = b_i \quad i = 1,2,\cdots,k$

$h_j(\boldsymbol{x}) \geq c_j \quad j = 1,2,\cdots,l$

最小化問題と最大化問題を区別せずに、最適化と呼ぶこともある。

3

判定問題との関係

数理計画法の定式化と対応するように判定問題を作ることができる。最小化問題に対応する判定問題の記述を示す。

最小化問題

名称：P

インスタンス：変数の集合、$f(\boldsymbol{x})$ の表現、$g_i(\boldsymbol{x})$ の表現、$\{b_1,b_2,\cdots,b_k\}$、$h_j(\boldsymbol{x})$ の表現、$\{c_1,c_2,\cdots,c_l\}$、定数K

問い：$f(\boldsymbol{x}) \leq K$ か？

この判定問題が解ければ、最小値を求める多項式時間アルゴリズムが得られる。

4

判定問題と最適化問題

ここでは、判定問題が多項式時間で解ければ、多項式時間で最小値が求まることを示す。（同様な、手法で最大値も求まる。）

判定問題を、パラメータKを持つ問題P(K)とみなす。判定問題の入力サイズは $\Omega(\log K)$ なので、$\log K$ の多項式のアルゴリズムを構成することを目指す。基本的なアイディアは、Kの値を2分探索で特定することである。

5

```
MIN(){
        K=1;
        while(P(K)==TURE){
                K=2*K;
        }
        K=OPT(K/2, K);
        return(K);
}

OPT(K_l,K_r){
        if(K_l>=K_r)return(K_r);
        K_m=(K_l+K_r)/2;
        if(P(K_m)==TRUE){
                OPT(K_m,K_r);
        }else{
                OPT(K_l,K_m);
        }
}
```

最適値の上界を求める。

最適値そのものを求める。

6

この計算時間を解析しよう。最適値を $K_{\min} \in Z$ とする。アルゴリズムでは、最適値の上界を求める部分で $O(\log K_{\min})$ 回の繰り返し、最適値を求める部分で $O(\log K_{\min})$ の再帰が行われるだけである。したがって、判定問題を $O(\log K_{\min})$ 回呼び出すだけで最適値を特定することができる。

ここで、判定問題を解くための計算量を $T(n)$ 時間とする。(n は変数や、条件式で定まる入力サイズとする。)このとき、上の議論から、$O(T(n)\log K_{\min})$ 時間で最適化問題を解くことができる。

なお、ここでの $K_{\min}$ は最適化問題の出力である。(判定問題では、Kは入力であったことにも注意しよう。)このように、出力サイズに計算時間が依存するアルゴリズムを出力依存型(Output sensitive)という。(判定問題のときには、出力サイズそのものに意味がなかったことに注意しよう。)

以上より、判定問題が多項式時間で解ければ、最適化問題も問題サイズ(入力サイズ+出力サイズ)の多項式時間で解くことができる。

7

線形計画法

数理計画法の問題の中で，評価関数や条件式が全て変数の一次式であるものを、線形計画問題という。

線形計画法は、特徴ベクトルを $\boldsymbol{x} = {}^t(x_1, x_2, \cdots, x_n) \in R^n$ とすると、次のように表せる。

線形計画法

P 最小化 $f(\boldsymbol{x}) = \sum_{j=1}^{n} p_j x_j$

条件 $g_i(\boldsymbol{x}) = \sum_{j=1}^{n} a_{ij} x_j = b_i \quad i = 1,2,\cdots,k$

$h_q(\boldsymbol{x}) = \sum_{j=1}^{n} a_{qj} x_j \geq c_j \quad q = 1,2,\cdots,l$

8

整数計画法

線形計画法では、特徴ベクトルの各要素は、実数でかまわなかった。それに対して，整数計画法では、特徴ベクトル(解) $\boldsymbol{x} = {}^t(x_1, x_2, \cdots, x_n) \in Z^n$ の各要素は整数でなければならない。したっがって、次のように表せる。

整数計画法

P 最小化 $f(\boldsymbol{x}) = \sum_{j=1}^{n} p_j x_j$

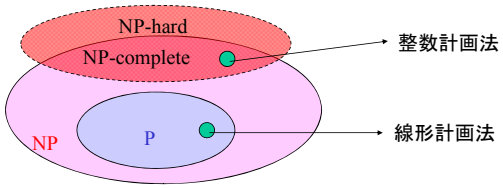
条件 $g_i(\boldsymbol{x}) = \sum_{j=1}^{n} a_{ij} x_j = b_i \quad i = 1,2,\cdots,k$

$h_q(\boldsymbol{x}) = \sum_{j=1}^{n} a_{qj} x_j \geq c_j \quad q = 1,2,\cdots,l$

$x_j \in Z \quad j = 1,2,\cdots,n$

9

数理計画法の計算量



なお、線形計画法がクラスPに属することがわかったのは、1979年にKhachiyanが楕円体法を発表してからである。その後、高速な計算法として、1984年にKarmarkarが内点法の一種を提案している。これらのアルゴリズムの発見以前は、単体法(シンプレックス法)が用いられていた。しかし、線形計画法を解く単体法は多項式時間アルゴリズムではなくて指数時間アルゴリズムである。(ただし、現実問題に単体法を適用すると、高速に解けることが多い。ただし、特別なインスタンスに対しては、多くの計算ステップを要する。)

10

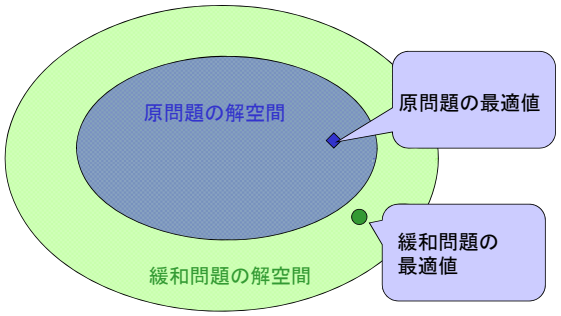
12−2. 緩和法

線形計画問題には実用的な解法がある。しかし、整数計画問題はNP完全なので、多項式時間アルゴリズムの構築は難しい。しかし、問題の性質上どうしても整数解が必要となるときがある。このような場合には、対応する線形計画問題を解くことで、整数計画問題の解に対しての知見が得られることがある。

このように、難しい問題(原問題)の制約条件を緩めて、解き易い問題に変換し、変換後の問題(緩和問題)の解から原問題に対する情報を得る解法のことを緩和法という。

緩和問題では、条件を緩めているので、解が存在できる空間(解空間)が原問題の解空間より広がってしまうので、緩和問題の最適解(緩和解)は原問題の最適値とは限らない。しかし、緩和解が、たまたま原問題の許容解(実行可能解ともいう)であれば、原問題の解が得られたことになる。また、緩和解は、原問題の解の存在範囲をある程度しぼってくれる役目もはたす。すなわち、最大化問題においては、緩和解は原問題の上界になっている。

11



12

線形緩和

整数問題において、整数制約をはずしてしまう緩和法を線形緩和という。

例題：ナップザック問題

まず、整数計画問題としてのナップザック問題を定式化する。

インスタンス例（原問題）

P 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, x_2, \cdots, x_7)$

最大化

$f(\boldsymbol{x}) = 9x_1 + 30x_2 + 21x_3 + 15x_4 + 23x_5 + 28x_6 + 7x_7$

条件

$1x_1 + 8x_2 + 6x_3 + 9x_4 + 15x_5 + 24x_6 + 21x_7 \leq 34$

$x_1, x_2, x_3, x_4, x_5, x_6, x_7 \in \{0,1\}$

整数条件

インスタンス例（緩和問題）

P' 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, x_2, \cdots, x_7)$

最大化

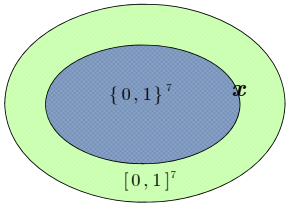
$f(\boldsymbol{x}) = 9x_1 + 30x_2 + 21x_3 + 15x_4 + 23x_5 + 28x_6 + 7x_7$

条件

$1x_1 + 8x_2 + 6x_3 + 9x_4 + 15x_5 + 24x_6 + 21x_7 \leq 34$

$x_1, x_2, x_3, x_4, x_5, x_6, x_7 \in [0,1]$

緩和した条件



14

このナップザック問題の場合には、緩和解が次のように得られる。

$$\frac{9}{1} > \frac{30}{8} > \frac{21}{6} > \frac{15}{9} > \frac{23}{24} > \frac{28}{24} > \frac{7}{21}$$

であるので、

$$\boldsymbol{x}_r = (1,1,1,1,\tfrac{2}{3},0,0) \qquad f(\boldsymbol{x}_r) = 90.33\cdots$$

と緩和解が得られる。ここで、原問題の評価値が整数であることに注意すると、原問題の最大値の上界が

$$90$$

であることがわかる。

15

部分列挙法

ある緩和を行った際に、場合分けを行ってさらに改善することがある。つまり、場合分けを行うことによって、緩和を強化できる。この方法を部分列挙法という。（なお、部分列挙法を系統的に繰り返して行う手法が分枝限定法である。）

ここでは、先ほどの例に対して部分列挙方の考え方を示す。

先ほどの、線形緩和で得られた緩和解は、

$$\boldsymbol{x}_r = (1,1,1,1,\tfrac{2}{3},0,0)$$

によって、緩和解の切り捨てによって、上界値90を求めることができた。ここで、非整数解 x_5 に注目して、問題を2つの分けることができる。

16

$x_5 = 0$ としたときの問題

子問題1

P0 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, x_2, x_3, x_4, 0, x_6, x_7)$

最大化

$f(\boldsymbol{x}) = 9x_1 + 30x_2 + 21x_3 + 15x_4 + 28x_6 + 7x_7$

条件

$1x_1 + 8x_2 + 6x_3 + 9x_4 + 24x_6 + 21x_7 \leq 34$

$x_1, x_2, x_3, x_4, x_6, x_7 \in \{0,1\}$

$x_5 = 1$ としたときの問題

子問題2

P1 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, x_2, x_3, x_4, 1, x_6, x_7)$

最大化

$f(\boldsymbol{x}) = 9x_1 + 30x_2 + 21x_3 + 15x_4 + 28x_6 + 7x_7 + 23$

条件

$1x_1 + 8x_2 + 6x_3 + 9x_4 + 24x_6 + 21x_7 \leq 19$

$x_1, x_2, x_3, x_4, x_6, x_7 \in \{0,1\}$

17

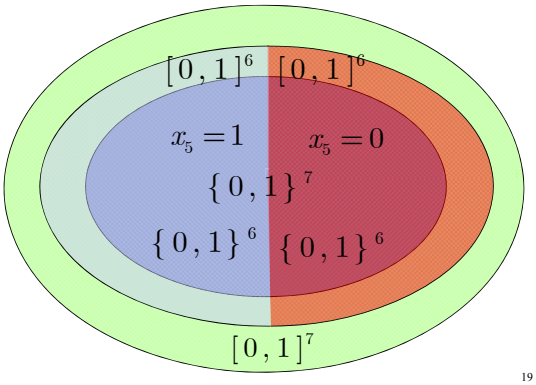
P0の緩和問題の解は、$\boldsymbol{x} = {}^t(1,1,1,1,0,\tfrac{5}{12},0)$で緩和値は$86.66\cdots$である。よって、P0の上界としては、その値の切り捨ての 86 が得られる。

一方、P1の緩和問題の解は、$\boldsymbol{x} = {}^t(1,1,1,\tfrac{4}{9},1,0,0)$で緩和値は$89.66\cdots$である。よって、P1の上界としては、その値の切り捨ての 89 が得られる。

ここで、原問題Pの上界値を考えよう。Pでは、x_5に関しては、0か1のいずれの値にしかならないので、Pの最適値は、P0の緩和値かP1の緩和値の大きい方よりは小さい。より大きい89がこの原問題Pの上界として求まる。この値は、単に線形緩和した場合より良い値になっている。

18

部分列挙法の問題領域



19

罰金法

緩和法の一種に罰金法というものがある。これは、ラグランジュ緩和法とも呼ばれる。この方法は、制約式を単純に取り除く変わりに、制約式を破るような解にはペナルティ(罰金)がかかるようにする方法である。次のような最大化問題に対して、ラグランジュ緩和の例を示す。

最大化問題

P 最大化 $f(\boldsymbol{x})$

条件 $g_i(\boldsymbol{x}) = b_i \quad i = 1,2,\cdots,k$

$h_j(\boldsymbol{x}) \geq c_j \quad j = 1,2,\cdots,l$

20

等式制約を取り除き、適当な数値 α_i 倍して目的関数に組み込めば、最大化問題に対するラグランジュ緩和問題が構成できる。

ラグランジュ緩和問題

$L(P,\alpha)$最大化 $f'(\boldsymbol{x}) = f(\boldsymbol{x}) - \sum_{i=1}^{k} \alpha_i |b_i - g_i(\boldsymbol{x})|$

条件

$h_j(\boldsymbol{x}) \geq c_j \quad j = 1,2,\cdots,l$

この問題は、ベクトル $\alpha = {}^t(\alpha_1,\alpha_2,\cdots,\alpha_k)$ を1とつ固定すると問題が一つ特定される。任意のベクトル α と問題 P の任意の許容解 $\bar{x}$ に対して $b_i - g_i(\bar{\boldsymbol{x}}) = 0$ が成り立ち、問題 $L(P,\alpha)$ の関数値は原問題 P の関数値と等しい。α の各値を∞にしたものを原問題とみなすことができる。したがって、問題 $L(P,\alpha)$ の最適値は、原問題の最適値以上になっている。

21

12−3. 分枝限定法

分枝限定法は、最も基本的でかつ適応範囲の広い方法である。分枝限定法では、緩和問題を何度も解くことによって原問題を解く方法である。解法の中心は、分枝操作と限定操作である。

分枝操作:
問題を子問題に分ける(場合分けする)操作である。緩和問題で、整数になっていない変数を固定することで場合分けすることが多い。

22

限定操作:
解かなくても良い子問題を見つけ，分枝操作を省く操作である。限定操作を行うことができるのは次のような状況である。以下では、最大化問題を扱っている。

(i)子問題が実行不可能である。

(ii)子問題の最適値の上界が、暫定解(原問題の許容解で、現在までわかった最も良い解)の目的関数値より良くない(同じか小さい。)

(iii)子問題の緩和問題の最適解が原問題の許容解である。

子問題の性質から、子問題の最適解の目的関数値は原問題の最適解より悪い。

23

ここでは、分枝限定法の基本的な枠組みを与える。

分枝限定法

1. 適当な方法で初期実行解を求め暫定解 $\boldsymbol{x}$ とし、その目標関数値を暫定値 $z = f(\boldsymbol{x})$ とする。問題の集合を $\mathcal{N} = \{P_0\}$ とする。(P_0 は原問題である。)
2. $\mathcal{N} = \phi$ ならば暫定解 を最適解として出力し終了する。そうでなければ、$\mathcal{N}$ から適当な子問題を選びそれを P' とし、 $\mathcal{N}$ から P' を取り除く。
3. P' の緩和問題を解き、得られた解を $\bar{x}'$ とし、上界値を $\bar{z}'$ とする。緩和問題が許容解を持たないならばステップ2へ。
4. $\bar{x}'$ が原問題 P_0 の実行可能解かつ $\bar{z}' > z$ の場合 $\boldsymbol{x} := \bar{\boldsymbol{x}}', z := \bar{z}'$ と更新し、ステップ2へ。 暫定解の更新
5. $\bar{z}' \leq z$ の場合、ステップ2へ。
6. $\bar{x}'$ が原問題 P_0 の実行可能解でないかつ $\bar{z}' > z$ の場合の実行可能領域を分割した子問題を生成し、それらを $\mathcal{N}$ に加えてステップ2へ。 子問題の生成

ここでは、以下の例題を基に分枝限定法を見ていく。

インスタンス例

P0 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, x_2, x_3, x_4)$

最大化

$$f(\boldsymbol{x}) = 3x_1 + 4x_2 + 1x_3 + 2x_4$$

条件

$$2x_1 + 3x_2 + 1x_3 + 3x_4 \le 4$$
$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

このとき、まず、$\mathcal{N} = \{P_0\}$ である。

まず、暫定解を $(z; x_1, x_2, x_3, x_4) = (2; 0, 0, 0, 1)$を発見したとする。

25

P0の線形緩和

P0' 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, x_2, x_3, x_4)$

最大化

$$f(\boldsymbol{x}) = 3x_1 + 4x_2 + 1x_3 + 2x_4$$

条件

$$2x_1 + 3x_2 + 1x_3 + 3x_4 \le 4$$
$$x_1, x_2, x_3, x_4 \in [0,1]$$

この緩和解は$(z; x_1, x_2, x_3, x_4) = (17/3; 1, 1/2, 0, 0)$ と求まる。よって、上界値として切り捨て5を得る。

この結果より、次の2つの子問題が生成される。

26

P1 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, 0, x_3, x_4)$

最大化

$$f(\boldsymbol{x}) = 3x_1 + 1x_3 + 2x_4$$

条件

$$2x_1 + 1x_3 + 3x_4 \le 4$$
$$x_1, x_3, x_4 \in \{0,1\}$$

P2 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, 1, x_3, x_4)$

最大化

$$f(\boldsymbol{x}) = 3x_1 + 1x_3 + 2x_4 + 4$$

条件

$$2x_1 + 1x_3 + 3x_4 \le 1$$
$$x_1, x_3, x_4 \in \{0,1\}$$

$$\mathcal{N} = \{P_1, P_2\}$$

27

P1を線形緩和して、緩和解 $(z; x_1, x_2, x_3, x_4) = (14/3; 1, 0, 1, 1/3)$ を得る。よって、上界値4が得られる。
ここで、ステップ6が実行され、次の2つの子問題を生成する。

P3 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, 0, x_3, 0)$

最大化

$$f(\boldsymbol{x}) = 3x_1 + 1x_3$$

条件

$$2x_1 + 1x_3 \le 4$$
$$x_1, x_3 \in \{0,1\}$$

P4 特徴ベクトル $\boldsymbol{x} = {}^t(x_1, 0, x_3, 1)$

最大化

$$f(\boldsymbol{x}) = 3x_1 + 1x_3 + 2$$

条件

$$2x_1 + 1x_3 \le 1$$
$$x_1, x_3 \in \{0,1\}$$

$$\mathcal{N} = \{P_3, P_4, P_2\}$$

28

P3を線形緩和して、緩和解 $(z; x_1, x_2, x_3, x_4) = (4; 1, 0, 1, 1)$ を得る。このとき、上界値4が得られる。この解は、全て整数であり、原問題の実行可能解である。よって、暫定解が得られる。したっがって、暫定解を更新する。また、P3はもはや子問題を生成しない。

$$\mathcal{N} = \{P_4, P_2\}$$

P4を線形緩和すると、緩和解$(z; x_1, x_2, x_3, x_4) = (7/2; 1/2, 0, 0, 1)$ を得る。よって、上界値3が得られる。しかし、この上界値は、暫定解より小さいのでこれ以上分枝操作を繰り返す必要はない。(枝刈りが生じる。)

$$\mathcal{N} = \{P_2\}$$

まだ、子問題が残っているので、この問題に対する分枝可能性をチェックする。

29

P2を線形緩和して、緩和解 $(z; x_1, x_2, x_3, x_4) = (11/2; 1/2, 1, 0, 0)$ を得る。このとき、上界値5が得られる。この場合は次の2つの子問題を生成する。

P5 特徴ベクトル $\boldsymbol{x} = {}^t(0, 1, x_3, x_4)$

最大化

$$f(\boldsymbol{x}) = 1x_3 + 2x_4 + 4$$

条件

$$1x_3 + 3x_4 \le 1$$
$$x_3, x_4 \in \{0,1\}$$

P6 特徴ベクトル $\boldsymbol{x} = {}^t(1, 1, x_3, x_4)$

最大化

$$f(\boldsymbol{x}) = 1x_3 + 2x_4 + 7$$

条件

$$1x_3 + 3x_4 \le -1$$
$$x_3, x_4 \in \{0,1\}$$

$$\mathcal{N} = \{P_5, P_6\}$$

30

P5を線形緩和して、緩和解 $(z;x_1,x_2,x_3,x_4)=(5;0,1,1,0)$ を得る。このとき、上界値5が得られる。この解は、全て整数であり、原問題の実行可能解である。よって、暫定解が得られる。したっがって、暫定解を更新する。また、P5はもはや子問題を生成しない。

$$\mathcal{N}=\{P_6\}$$

P6は明らかに緩和解が存在しない(実行不能)。よって、無条件で削除する。

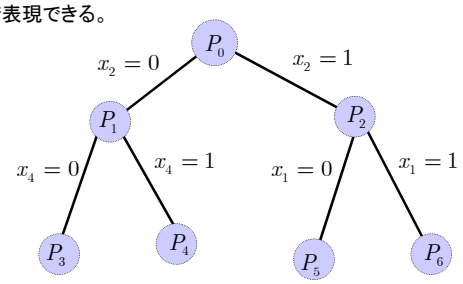
$$\mathcal{N}=\{\}$$

以上で全ての子問題を処理したので、暫定解の $(z;x_1,x_2,x_3,x_4)=(5;0,1,1,0)$ が真の最適値である。

31

分枝木

分枝限定法は、問題をノードとし、子問題生成を枝とする木構造で表現できる。



32

分枝限定法の性能

分枝限定法は、分枝木に基づく列挙法の一種であり、最悪の場合には、すべての許容解を列挙してしまう。しかし、現実問題に適用してみると、性能の良い緩和法を用いることによりきわめて効果的に働き、ほんのわずかな回数だけ緩和問題を解くことにより、大規模な問題の最適解(厳密解)を得ることに成功している。(成功例が多数報告されている。)

33

分枝限定法の方針

分枝限定法では、子問題の集合$\mathcal{N}$からどの順序で次に処理する子問題を選択するかに自由度がある。この選択法には、主に3つの方法がとられることが多い。

- 深さ優先探索
  選択可能な子問題の内で、最も後に生成されたものを選ぶ方法。スタックを用いて $\mathcal{N}$ を実現することで、深さ優先探索型の分枝限定法になる。メリットとしては、記憶容量をあまり必要としないことがあげられる。大規模な問題を解くにはこの選択枝しかないこともある。
- 幅優先探索
  選択可能な子問題の内で、最も前に生成されたものを選ぶ方法。キューを用いて $\mathcal{N}$ を実現することで、幅優先探索型の分枝限定法になる。メリットとしては、問題を高速に解くことがあげられる。しかし、必要な記憶量が膨大になることが多い。
- 最良値探索
  緩和値が最も良いものを選ぶ方法。緩和問題の性質に依存する。