線形計画法
Linear Programing

計算機アルゴリズム特論：2015年度

只木進一

線形計画法

- 連立一次不等式で表された領域中において、一次式で表された値を最大または最小とする
- 例
  - 条件

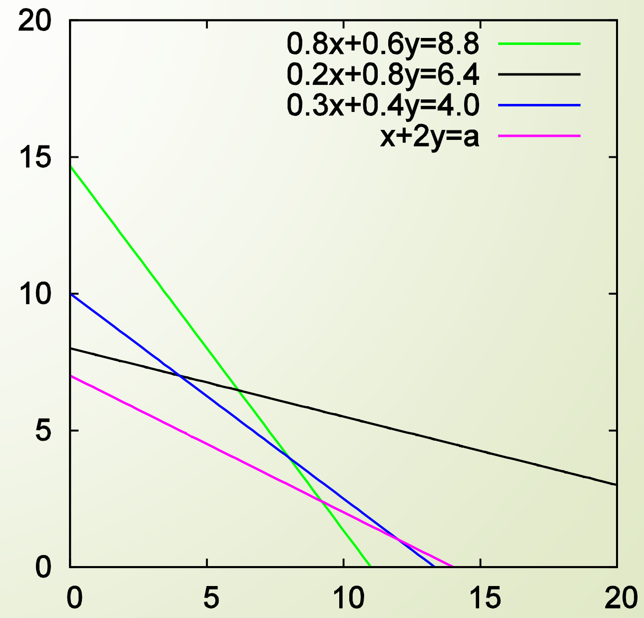
$$0.8x + 0.6y \le 8.8$$
$$0.2x + 0.8y \le 6.4$$
$$0.3x + 0.4y \le 4.0$$
$$x \ge 0,\ y \ge 0$$

  - $f(x, y) = x + 2y$を最大化する

- 線形計画法そのものはアルゴリズムではない
- 様々な問題が線形計画法として扱える
- 最も基本的方法として、simplex法がある

Simplex法の考え方

- 条件で表された凸多角形の頂点のいずれかで最大・最小値が得られる

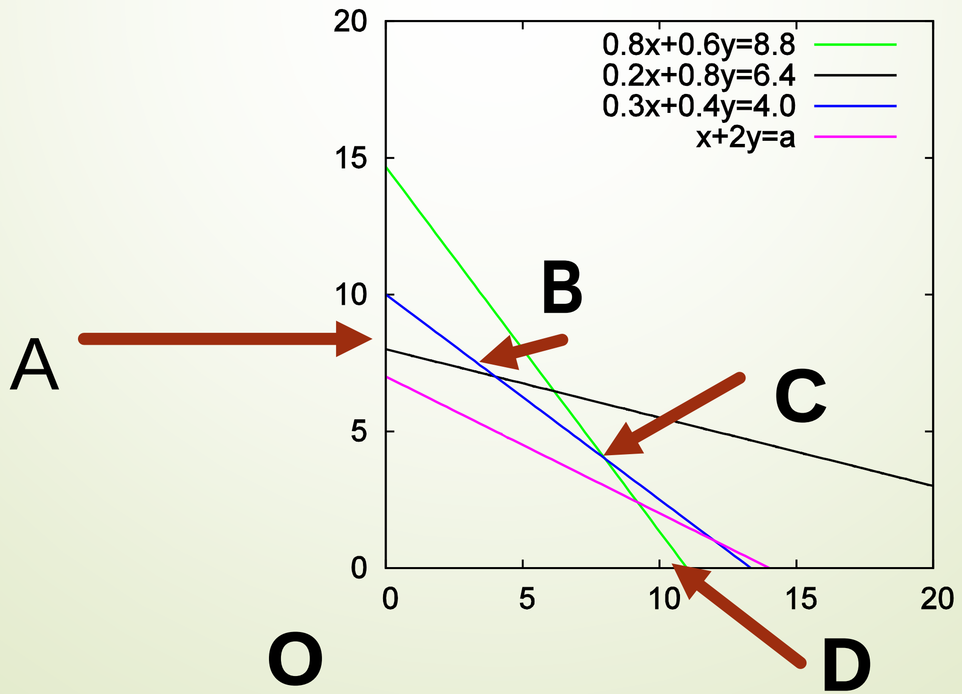


- 連立不等式を連立方程式に変形
  - 新しい変数(slack variables)の導入
  - slack変数は、凸多角形の内側が正になるように導入

$$0.8x + 0.6y + \mu = 8.8$$
$$0.2x + 0.8y + \nu = 6.4$$
$$0.3x + 0.4y + \lambda = 4.0$$
$$x \geq 0,\ y \geq 0,\ \mu \geq 0,\ \nu \geq 0,\ \lambda \geq 0$$

Simplex法の考え方

- 頂点は、二つ以上の変数がゼロになる部分に対応



- 連立方程式の解は二つのパラメタで表される

$$\left(\begin{array}{ccccc|c} 0.8 & 0.6 & 1 & 0 & 0 & 8.8 \\ 0.2 & 0.8 & 0 & 1 & 0 & 6.4 \\ 0.3 & 0.4 & 0 & 0 & 1 & 4.0 \end{array}\right)$$

$$\begin{pmatrix} x \\ y \\ \mu \\ \nu \\ \lambda \end{pmatrix} = x' \begin{pmatrix} 1 \\ 0 \\ -0.8 \\ -0.2 \\ -0.3 \end{pmatrix} + y' \begin{pmatrix} 0 \\ 1 \\ -0.6 \\ -0.8 \\ -0.4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 8.8 \\ 6.4 \\ 4.0 \end{pmatrix}$$

- x'とy'を与えると、空間内の点を特定

$$\begin{pmatrix} x \\ y \\ \mu \\ \nu \\ \lambda \end{pmatrix} = x' \begin{pmatrix} 1 \\ 0 \\ -0.8 \\ -0.2 \\ -0.3 \end{pmatrix} + y' \begin{pmatrix} 0 \\ 1 \\ -0.6 \\ -0.8 \\ -0.4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 8.8 \\ 6.4 \\ 4.0 \end{pmatrix}$$

$$z = x' + 2y'$$

- $x' = y' = 0$(頂点Oに対応)からパラメタを変化させてzを増大させる。
- $x' = 0$のままy'を増大させると効果的

$$0.6y + \mu = 8.8$$

$$0.8y + \nu = 6.4$$

$$0.4y + \lambda = 4.0$$

- 変数域の制約から$y' = 8$まで増加させる。
  - 定数項をyの係数で除した値の最小値が制約
  - このとき$\nu = 0$となる(頂点A)

- パラメタの組を(x, y)から(x, v)へ
- 二番目の式でyの係数を1に

$$0.2x + 0.8y + v = 6.4 \Rightarrow 0.25\mathrm{x} + \mathrm{y} + 1.25v = 8$$

- 1番目と3番目の式で、yの係数を0に

$$\begin{array}{rr} & 0.8x + 0.6y = 8.8 \\ -\quad) & 0.15x + 0.6y + 0.75v = 4.8 \\ & 0.65 - 0.75v = 4 \end{array}$$

表形式に整理

- パラメタの組を(x, y)から(x, v)へ

$$\left(\begin{array}{ccccc|c} 0.8 & 0.6 & 1 & 0 & 0 & 8.8 \\ 0.2 & 0.8 & 0 & 1 & 0 & 6.4 \\ 0.3 & 0.4 & 0 & 0 & 1 & 4.0 \end{array}\right)$$

$$\Downarrow$$

$$\left(\begin{array}{ccccc|c} 0.65 & 0 & 1 & -0.75 & 0 & 4 \\ 0.25 & 1 & 0 & 1.25 & 0 & 8 \\ 0.2 & 0 & 0 & -0.5 & 1 & 0.8 \end{array}\right)$$

- $y = -0.25x - 1.25v + 8$を$z = x + 2y$に代入

$$\begin{aligned} z &= x + 2y \\ &= 0.5x - 2.5v + 16 \end{aligned}$$

- 目的関数の変化も一緒に、表形式で表しておく

	x	y	μ	ν	λ	定数	θ
μ	0.8	0.6	1	0	0	8.8	
ν	0.2	0.8	0	1	0	6.4	
λ	0.3	0.4	0	0	1	4.0	
z	-1	-2	0	0	0	0	

z行の最小負数の列に縦枠
定数列を縦枠の値で除した値をθ列へ

	x	y	μ	ν	λ	定数	θ
μ	0.8	0.6	1	0	0	8.8	$\frac{8.8}{0.6} = 14.6 \ldots$
ν	0.2	0.8	0	1	0	6.4	$\frac{6.4}{0.8} = 8$
λ	0.3	0.4	0	0	1	4.0	4.0/0.4=10
z	-1	-2	0	0	0	0	

θ列の正の最小値に対応する行に横枠を

縦枠と横枠の交点（pivot）の値で、横枠の各数値を除する

	x	y	μ	ν	λ	**定数**	θ
μ	0.8	0.6	1	0	0	8.8	
ν	0.25	1	0	1.25	0	8	
λ	0.3	0.4	0	0	1	4.0	
z	-1	-2	0	0	0	0	

横枠行に適当な数値を乗じて、他の行からpivot以外の縦枠の数値をゼロとする

	x	y	μ	ν	λ	**定数**	θ
μ	0.65	0	1	-0.75	0	4	
ν	0.25	1	0	1.25	0	8	
λ	0.2	0	0	-0.5	1	0.8	
z	-0.5	0	0	2.5	0	16	

	x	y	μ	ν	λ	定数	θ
μ	0.65	0	1	-0.75	0	4	
ν	0.25	1	0	1.25	0	8	
λ	0.2	0	0	-0.5	1	0.8	
z	-0.5	0	0	2.5	0	16	

	x	y	μ	ν	λ	定数	θ
μ	0.65	0	1	-0.75	0	4	6.15…
ν	0.25	1	0	1.25	0	8	32
λ	0.2	0	0	-0.5	1	0.8	4
z	-0.5	0	0	2.5	0	16	

	x	y	μ	ν	λ	定数	θ
μ	0.65	0	1	-0.75	0	4	
ν	0.25	1	0	1.25	0	8	
λ	1	0	0	-2.5	5	4	
z	-0.5	0	0	2.5	0	16	

	x	y	μ	ν	λ	定数	θ
μ	0	0	1	0.875	-3.25	1.4	
ν	0	1	0	1.875	-1.25	1	
λ	1	0	0	-2.5	5	4	
z	0	0	0	1.25	2.5	18	

z行に負の係数は無い→これ以上増えない
最大値は18

n個の変数とm個の条件式に一般化

- 変数　$x_i (0 \le i < n)$
- 条件式

$$\sum_{j=0}^{n-1} a_{ij} x_j = b_i \ (0 \le i < m)$$

- 目的関数

$$z = \sum_{i=0}^{n-1} c_j x_j$$

- Slack variables $\xi_k \ (0 \le k < m - n)$

$$\sum_{j=0}^{n-1} a_{ij} x_j + \xi_i = b_i \ (0 \le i < m)$$

$$\sum_{i=0}^{n-1} -c_i x_i = 0$$

- 行列形式

$$\sum_{j=0}^{n+m} A_{ij} y_j = B_i \ (0 \le i < m+1)$$

$$B_i = \begin{cases} b_i & 0 \le i < m \\ 0 & i = m \end{cases}$$

$$y_i = \begin{cases} x_i & 0 \le i < n \\ \xi_i & n \le i < n + m \end{cases}$$

$$A_{ij} = \begin{cases} a_{ij} & 0 \le i < m, 0 \le j < n \\ -c_i & i = m, 0 \le j < n \\ 1 & 0 \le i < m, j = i + n \\ 0 & \text{otherwise} \end{cases}$$

```
f = false;
while(!f){
    k=評価式中の係数の最小値; q = A_mk;
    if(q ≥ 0) f = true;
    else{
        l =pivotの行; p = A_lk;
        for(0 ≤ j < n + m) A_lj = A_lj / p;
        B_l = B_l / p;
        for(0 ≤ i < m + 1){
            if(i ≠ l){
                r = A_ik / A_lk;
                for(0 ≤ j < n + m) A_ij = A_ij − r × A_lj;
                B_i = B_i − r × B_l;
            }
        }
    }
}
```