

問題解決

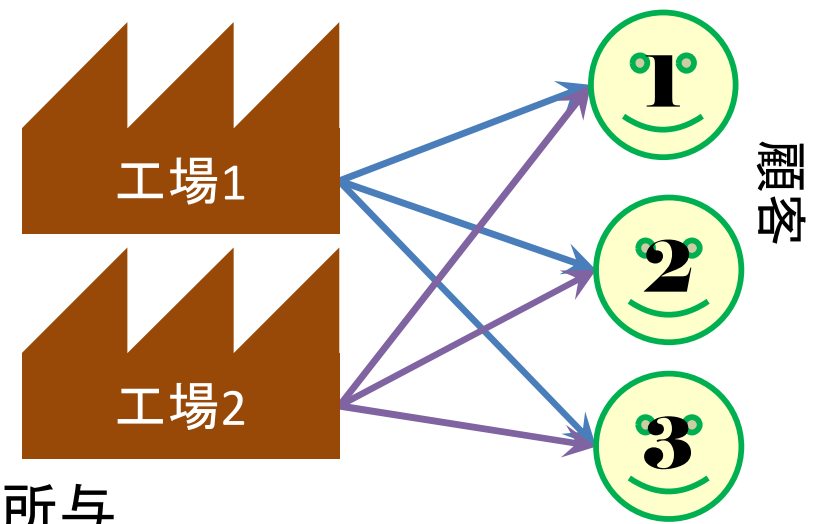
最適化活用  
整数計画ソルバーの利用

堀田 敬介

# 輸送問題を解く

## ➤ 輸送問題の最適化(例1)

- 2工場で製品を供給できる
- 3人の顧客がいて、需要がある
- 2工場→3顧客への単位あたり輸送コストが所与
- **輸送コスト**が最小となる配送計画をたてよ



## ➤ 最適化問題の定式化(変数設定)

- 変数  $x_{ij}$  ... 工場 $i$ →顧客 $j$ への輸送量
- 変数行列  $X$

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix}$$

## ➤ 最適化問題の定式化(係数表記)

- 工場の供給を表す係数ベクトル  $s$
- 顧客の需要を表す係数ベクトル  $d$
- 輸送コストを表す係数行列  $C$

		需要	50	80	60
供給	工場\顧客	1	2	3	
120	1	3	2	4	
130	2	5	6	5	

$$s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 120 \\ 130 \end{pmatrix}$$

$$d = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} 50 \\ 80 \\ 60 \end{pmatrix}$$

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{pmatrix} = \begin{pmatrix} 3 & 2 & 4 \\ 5 & 6 & 5 \end{pmatrix}$$

# 輸送問題を解く

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix}$$

## ➤ 最適化問題の定式化(ベタ表記)

$$\min. 3x_{11} + 2x_{12} + 4x_{13} + 5x_{21} + 6x_{22} + 5x_{23}$$

$$\text{s. t. } x_{11} + x_{12} + x_{13} \leq 120$$

$$x_{21} + x_{22} + x_{23} \leq 130$$

$$x_{11} + x_{21} = 50$$

$$x_{12} + x_{22} = 80$$

$$x_{13} + x_{23} = 60$$

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23} \geq 0$$

$$\mathbf{s} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 120 \\ 130 \end{pmatrix}$$

$$\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} 50 \\ 80 \\ 60 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{pmatrix} = \begin{pmatrix} 3 & 2 & 4 \\ 5 & 6 & 5 \end{pmatrix}$$

## ➤ 最適化問題の定式化( $\Sigma$ 表記)

$$\min. \sum_{i=1}^2 \sum_{j=1}^3 c_{ij} x_{ij}$$

$$\text{s. t. } \sum_{j=1}^3 x_{ij} \leq s_i (i = 1, 2)$$

$$\sum_{i=1}^2 x_{ij} = d_j (j = 1, \dots, 3)$$

$$x_{ij} \geq 0 (i = 1, 2; j = 1, \dots, 3)$$

# How to use CPLEX?

## ➤ 新規プロジェクトの作成

- ① [ファイル(F)]－[新規(N)]－[OPLプロジェクト]を選択
- ② [プロジェクト名]を記入(例: **Transportation**)し, 3カ所にチェックする
  - デフォルトの実行構成の追加
  - モデルの作成
  - データの作成
- ③ [終了]をクリック

プロジェクト名は自由だが, **半角英数**で何の問題を解こうとしているのかが分かる名前が良い

## ➤ プロジェクト内のいくつかの名前を変更

- ✓ [構成1] → [**config1**] ※日本語を英語に変更しないと実行時エラーになる
- ✓ モデルファイル [Transportation.mod] → [**tr.mod**]
- ✓ データファイル [Transportation.dat] → [**trax1.dat**]

## ➤ モデルファイル・データファイルを記述し保存(次ページ参照)

## ➤ [config1]にモデルファイルとデータファイルをセットする

# How to use CPLEX?

➤ tr.mod

```
int i_max = ...; // 行の添え字の最大値
int j_max = ...; // 列の添え字の最大値

range I = 1..i_max; // 行の添え字の範囲 [1..i_max]を指定
range J = 1..j_max; // 列の添え字の範囲 [1..j_max]を指定

int d[J] = ...; // 需要ベクトル設定 d = [d1,d2,d3]
int s[I] = ...; // 供給ベクトル設定 s = [s1,s2]
int c[I,J] = ...; // 輸送コスト設定 C サイズI×Jの行列

dvar float+ x[I,J]; // 変数宣言:変数ベクトル(size: I×J)

minimize
    sum(i in I) sum(j in J) c[i,j]*x[i,j];
subject to{
    forall(i in I) {
        sum(j in J) x[i,j] <= s[i];
    };
    forall(j in J) {
        sum(i in I) x[i,j] == d[j];
    };
};
```

# How to use CPLEX?

➤ trex1.dat

```
i_max = 2; // 行数の最大値を指定  
j_max = 3; // 列数の最大値を指定
```

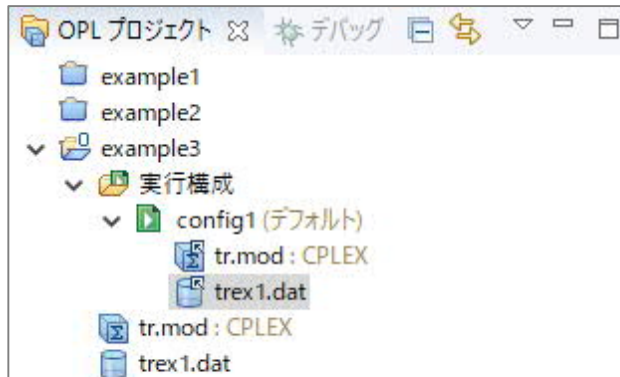
```
d = [50 80 60];
```

```
c = [  
[3 2 4]  
[5 6 5]  
];
```

```
s = [  
120  
130  
];
```

# How to use CPLEX?

## ▶ 計算結果の確認



```
1 /*****  
2 * OPL 12.10.0.0 Model  
3 * Author:  
4 * Creation Date: 2020/06/24 at 22:50:00  
5 *****/  
6 int i_max = ...; // 行の添え字の最大値  
7 int j_max = ...; // 列の添え字の最大値  
8  
9 range I = 1..i_max; // 行の添え字の範囲 [1..i_max]を  
10 range J = 1..j_max; // 列の添え字の範囲 [1..j_max]を  
11  
12 int d[J] = ...; // 需要ベクトル設定 d = [d1,d2,d3]  
13 int s[I] = ...; // 供給ベクトル設定 s = [s1,s2]  
14 int c[I,J] = ...; // 輸送コスト設定 C サイズI×Jの行列  
15  
16 dvar float+ x[I,J];  
17  
18 minimize
```

統計情報	値
▼ Cplex	
solution (optimal) with objective 630	
Constraints	5
▼ Variables	6
Other	6
Non-zero coefficients	12
Iterations	1

最適解発見

[統計情報]タブの中身

solution (optimal)  
解(最適)

[解]タブの中身

最適値 = 630  
各定数(係数行列)  
最適解  $x = \begin{bmatrix} 40 & 80 & 0 \\ 10 & 0 & 60 \end{bmatrix}$

目的 630 の解

名前	値
▼ データ (7)	
c	[[3 2 4] [5 6 5]]
d	[50 80 60]
I	1..2
i_max	2
J	1..3
j_max	3
s	[120 130]
▼ 決定変数 (1)	
x	[[40 80 0] [10 0 60]]

```
// solution (optimal) with objective 630  
// Quality there are no bound infeasibilities.  
// There are no reduced-cost infeasibilities.  
// Maximum Ax-b residual = 0  
// Maximum c-B'pi residual = 0  
// Maximum |x| = 80  
// Maximum |slack| = 60  
// Maximum |pi| = 5  
// Maximum |red-cost| = 2  
// Condition number of unscaled basis = 9.0e+00  
//  
x = [[40  
      80 0]  
      10 0 60]];
```

# 輸送問題を解く

## ➤ 輸送問題の最適化(例2)

- 工場が製品を供給, 顧客が需要, 工場→顧客コスト
- 輸送コストが最小となる配送計画をたてよ

## ➤ 最適化問題の定式化(ベタ・Σ表記)

- 変数  $x_{ij}$  ... 工場*i*→顧客*j*への輸送量
- 係数vector *s*:供給supply, *d*:需要demand
- 係数matrix *C*:輸送コストcost

	需要	40	30	70	80	60	50
供給	工場\顧客	1	2	3	4	5	6
80	1	1	2	4	3	1	2
90	2	2	1	5	2	4	1
100	3	3	6	2	1	2	3
110	4	4	3	5	2	3	2

- CPLEXで解く(モデルファイル[tr.mod]は共通で使えるので[trex2.dat]のみ作り解く)