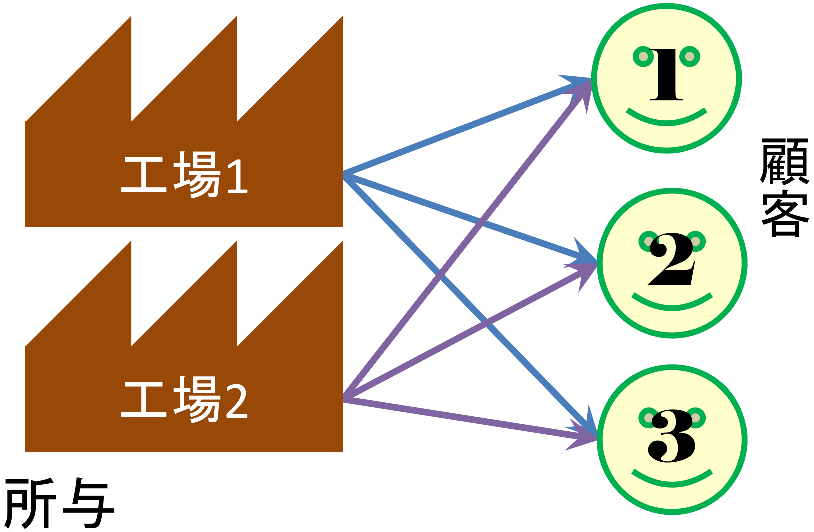
2023/5/29 Mon.

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

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

- 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 $\boldsymbol{X}$

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

- 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 $\boldsymbol{s}$
  - 顧客の需要を表す係数ベクトル $\boldsymbol{d}$
  - 輸送コストを表す係数行列 $\boldsymbol{C}$

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

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

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

輸送問題を解く

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

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

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

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

- 最適化問題の定式化（ベタ表記）

$$\begin{aligned}
\text{min. } & 3x_{11}+2x_{12}+4x_{13}+5x_{21}+6x_{22}+5x_{23} \\
\text{s. t. } & x_{11}+x_{12}+x_{13} \leqq 120 \\
& x_{21}+x_{22}+x_{23} \leqq 130 \\
& x_{11}+x_{21} = 50 \\
& x_{12}+x_{22} = 80 \\
& x_{13}+x_{23} = 60 \\
& x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23} \geqq 0
\end{aligned}$$

- 最適化問題の定式化（Σ表記）

$$\begin{aligned}
\text{min.} & \sum_{i=1}^{2} \sum_{j=1}^{3} c_{ij} x_{ij} \\
\text{s. t. } & \sum_{j=1}^{3} x_{ij} \leq s_i \, (i = 1,2) \\
& \sum_{i=1}^{2} x_{ij} = d_j \, (j = 1, \dots, 3) \\
& x_{ij} \geq 0 \; (i = 1,2;\, j = 1, \dots, 3)
\end{aligned}$$

How to use CPLEX?

- 新規プロジェクトの作成
  ① [ファイル(F)]ー[新規(N)]ー[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] → [trex1.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];
  };
};
```

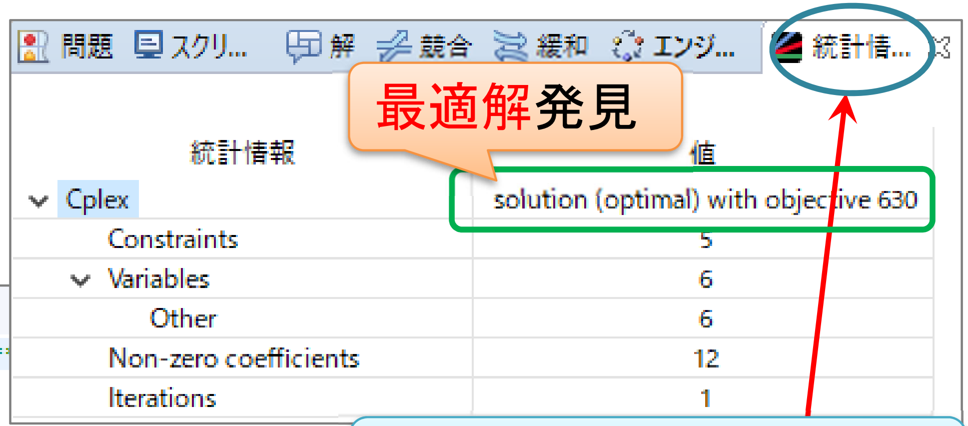
How to use CPLEX?

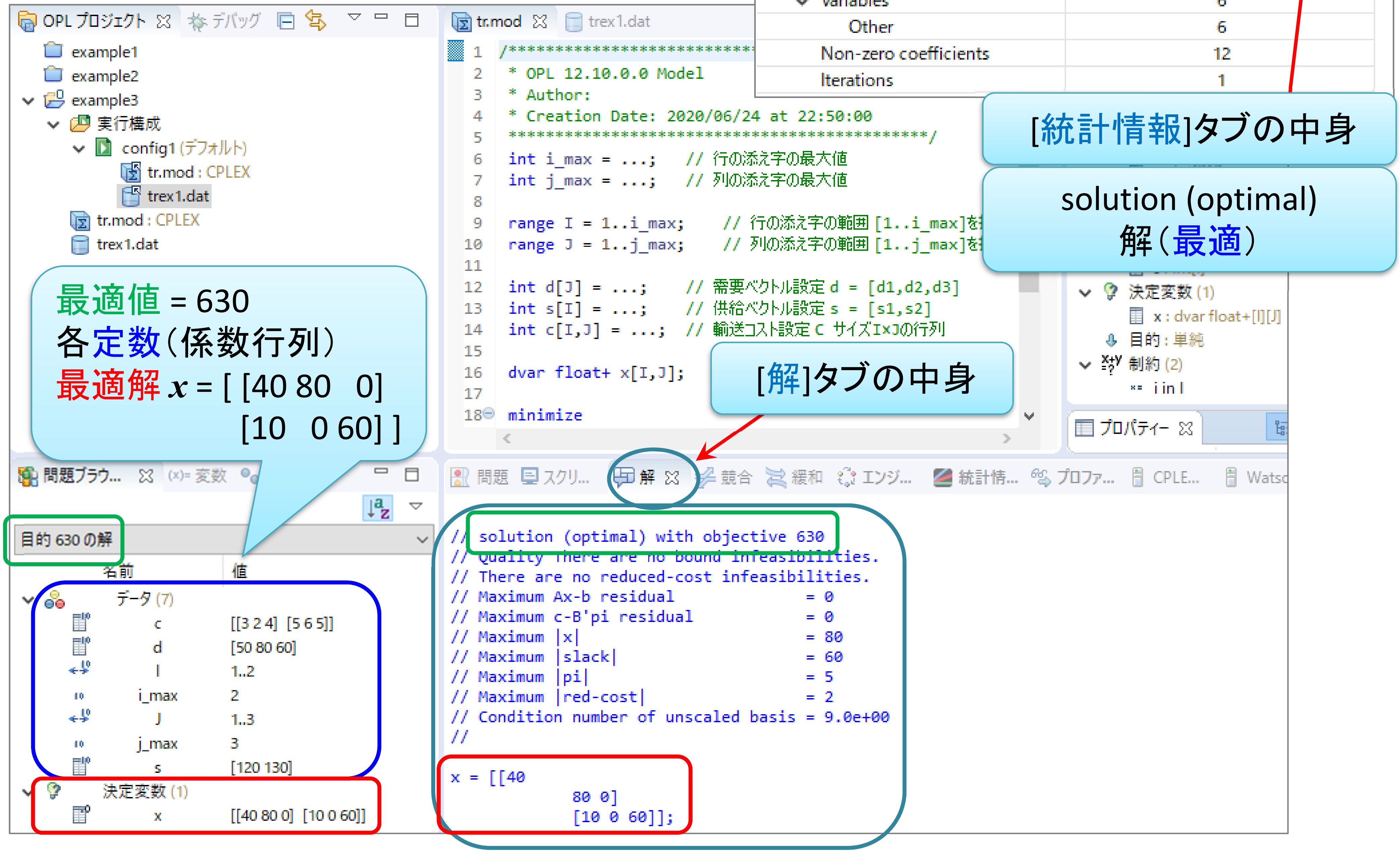
- trex1.dat

```
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





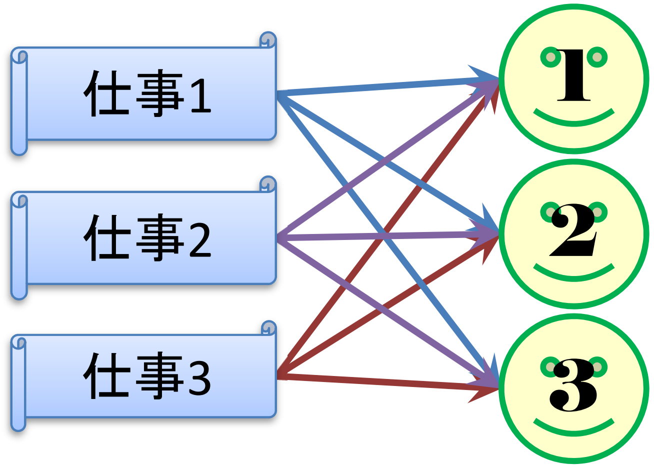
輸送問題を解く

- 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)

割当問題を解く



- 割当問題の最適化（例1）
  - 今すべき仕事は3つあり，3人の部下がいる
  - 各部下は1つの仕事を引き受けられる
  - 上司は各仕事を任せたときの部下の出来具合を5段階で評価済（右下表）
  - この部署のパフォーマンスが最大になるように部下へ仕事を割り当てよ

仕事＼部下	1	2	3
1	2	2	4
2	5	3	2
3	4	1	3

- 最適化問題の定式化（変数設定）
  - 0-1変数 $x_{ij} = \begin{cases} 1 \ldots \text{仕事}i\text{を部下}j\text{へ割り当てる} \\ 0 \ldots \text{仕事}i\text{を部下}j\text{へ割り当てない} \end{cases}$
  - 変数行列X

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$$

- 最適化問題の定式化（係数表記）
  - 評価を表す係数行列 $\boldsymbol{C}$

$$\boldsymbol{C} = \begin{pmatrix} 2 & 2 & 4 \\ 5 & 3 & 2 \\ 4 & 1 & 3 \end{pmatrix}$$

割当問題を解く

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$$

$$\boldsymbol{C} = \begin{pmatrix} 2 & 2 & 4 \\ 5 & 3 & 2 \\ 4 & 1 & 3 \end{pmatrix}$$

- 割当問題の最適化（（ベタ表記）

$$\begin{aligned}
\text{max.}\ & 2x_{11}+2x_{12}+4x_{13} + 5x_{21}+3x_{22}+2x_{23} + 4x_{31}+1x_{32}+3x_{33} \\
\text{s. t.}\ & x_{11}+x_{12}+x_{13}=1 \\
& x_{21}+x_{22}+x_{23}=1 \\
& x_{31}+x_{32}+x_{33}=1 \\
& x_{11}+x_{21}+x_{31} \leqq 1 \\
& x_{12}+x_{22}+x_{32} \leqq 1 \\
& x_{13}+x_{23}+x_{33} \leqq 1 \\
& x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23} \in \{0,1\}
\end{aligned}$$

- 最適化問題の定式化（Σ表記）

$$\begin{aligned}
\text{max.}\ & \sum_{i=1}^{3} \sum_{j=1}^{3} c_{ij} x_{ij} \\
\text{s. t.}\ & \sum_{j=1}^{3} x_{ij} = 1 \ (i = 1, \dots, 3) \\
& \sum_{i=1}^{3} x_{ij} \leq 1 \ (j = 1, \dots, 3) \\
& x_{ij} \in \{0,1\} (i = 1, \dots, 3; j = 1, \dots, 3)
\end{aligned}$$

How to use CPLEX?

- 新規プロジェクトの作成
  - ① [ファイル(F)]ー[新規(N)]ー[OPLプロジェクト]を選択
  - ② [プロジェクト名] を記入（例：Assignment）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 [Assignment.mod] → [as.mod]
  - ✓ データファイル [Assignment.dat] → [asex1.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?

- as.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 c[I,J] = ...;// 評価値Cij(size:I×J)

dvar int+ x[I,J] in 0..1;// 変数宣言:0-1変数(size:I×J)

maximize
  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] == 1;
  };
  forall(j in J) {
    sum(i in I) x[i,j] <= 1;
  };
};
```

How to use CPLEX?

- asex1.dat

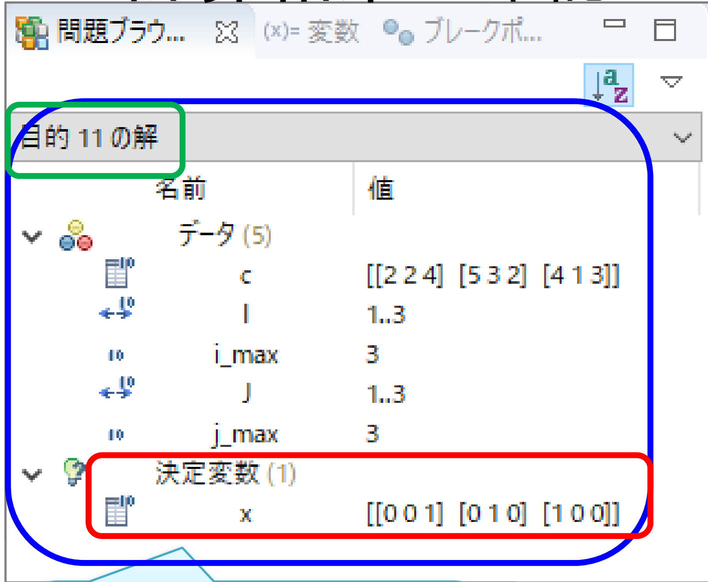
```
i_max = 3;
j_max = 3;

c = [
[2 2 4]
[5 3 2]
[4 1 3]
];
```

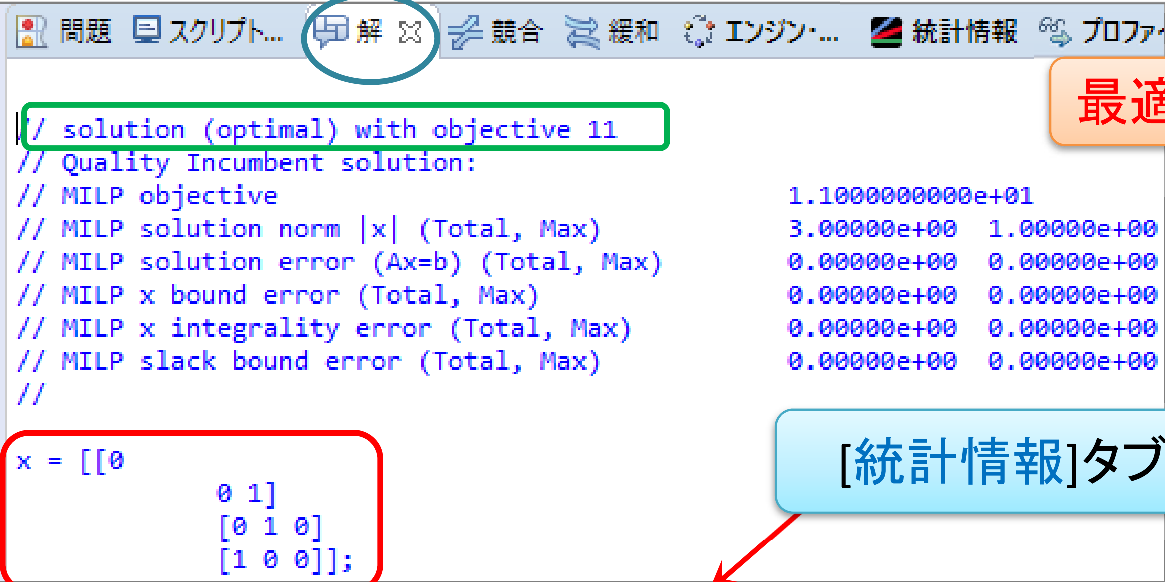
How to use CPLEX?

➢ 計算結果の確認

[解]タブの中身

solution (optimal)
解（最適）

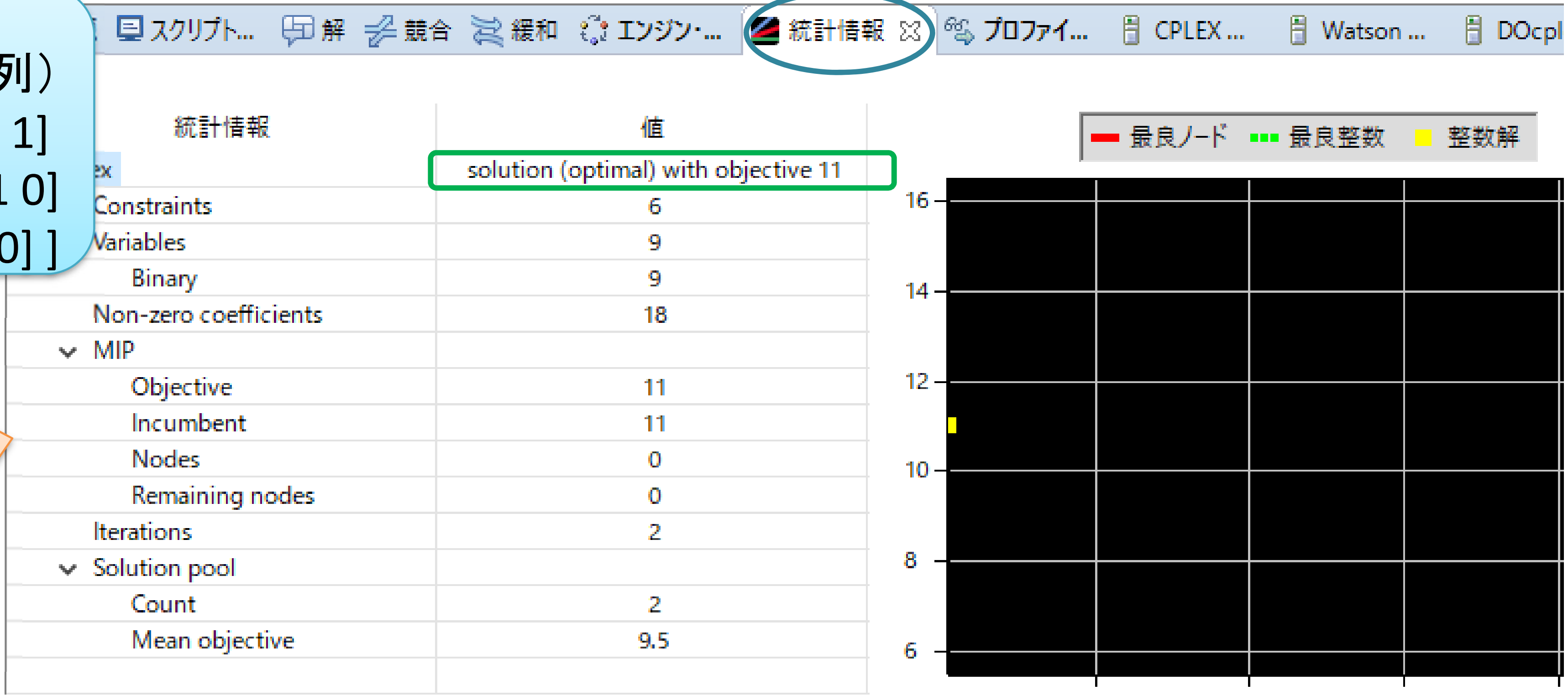




最適解発見

[統計情報]タブの中身

最適値 = 11
各定数（係数行列）
最適解 x = [[0 0 1]
[0 1 0]
[1 0 0]]



LPではなく
IPなので
情報が少し
違う

割当問題を解く2

- 割当問題の最適化(例2)
  - 仕事は4つ，部下は7人，各部下は1つの仕事を引き受けられる
  - 上司は部下を5段階[1,2,3,4,5]で評価済(下表)
  - この部署のパフォーマンスが最大になるように部下へ仕事を割り当てよ
- 最適化問題の定式化(ベタ・Σ表記)
  - 0-1変数 x_{ij} =1 … 仕事i を部下j へ割り当てる，x_{ij} =0 … 割り当てない
  - 係数matrix ***C***:上司による部下評価値

仕事＼部下	1	2	3	4	5	6	7
1	*2*	*4*	*2*	*1*	*3*	*1*	*4*
2	*3*	*1*	*3*	*2*	*5*	*4*	*2*
3	*1*	*3*	*4*	*4*	*4*	*3*	*3*
4	*4*	*2*	*1*	*4*	*5*	*2*	*1*

- CPLEXで解く(モデルファイル[as.mod]は共通で使えるので[asex2.dat]のみ作り解く)

生産計画をたてる

材料＼製品	α	β	γ	δ	ε	量
A	3	0	1	3	1	80
B	1	2	0	3	2	75
C	0	4	2	5	0	95
D	2	1	0	1	2	70
利益	6	7	3	10	5	

- 生産計画の最適化（例1）
  - 4種の材料 A, B, C, D がある
  - 5種の製品 $\alpha, \beta, \gamma, \delta, \varepsilon$ をつくる
  - 各材料の所有数，各製品を1単位作るのに必要な材料数，利益は右表
  - **総利益**を**最大**にする生産計画をたてたい

- 最適化問題の定式化（変数設定）
  - 変数 x_i … 製品i の作成数（単位）
  - 変数ベクトル $\boxed{\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)}$

- 最適化問題の定式化（係数表記）
  - 利益を表す係数ベクトル $\boldsymbol{c}$
  - 材料所持数を表す係数ベクトル $\boldsymbol{b}$
  - 必要材料数を表す係数行列 A

$$\boldsymbol{c} = (c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5)$$
$$= (6 \quad 7 \quad 3 \quad 10 \quad 5)$$
$$\boldsymbol{b} = (b_1 \quad b_2 \quad b_3 \quad b_4)$$
$$= (80 \quad 75 \quad 95 \quad 70)$$
$$\mathrm{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{pmatrix}$$
$$= \begin{pmatrix} 3 & 0 & 1 & 3 & 1 \\ 1 & 2 & 0 & 3 & 2 \\ 0 & 4 & 2 & 5 & 0 \\ 2 & 1 & 0 & 1 & 2 \end{pmatrix}$$

生産計画をたてる

- 最適化問題の定式化(ベタ表記)

$$
\begin{aligned}
\text{max.}\quad & 6x_1+7x_2+3x_3+10x_4+5x_5 \\
\text{s. t.}\quad & 3x_{11}+0x_{12}+1x_{13}+3x_{14}+1x_{15} \leqq 80 \\
& 1x_{21}+2x_{22}+0x_{23}+3x_{24}+2x_{25} \leqq 75 \\
& 0x_{31}+4x_{32}+2x_{33}+5x_{34}+0x_{35} \leqq 95 \\
& 2x_{41}+1x_{42}+0x_{43}+1x_{44}+2x_{45} \leqq 70 \\
& x_1, x_2, x_3, x_4, x_5 \geqq 0
\end{aligned}
$$

$$
\boldsymbol{c} = (c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5) = (6 \quad 7 \quad 3 \quad 10 \quad 5)
$$

$$
\boldsymbol{b} = (b_1 \quad b_2 \quad b_3 \quad b_4) = (80 \quad 75 \quad 95 \quad 70)
$$

$$
A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{pmatrix} = \begin{pmatrix} 3 & 0 & 1 & 3 & 1 \\ 1 & 2 & 0 & 3 & 2 \\ 0 & 4 & 2 & 5 & 0 \\ 2 & 1 & 0 & 1 & 2 \end{pmatrix}
$$

- 最適化問題の定式化(Σ表記)

$$
\begin{aligned}
\text{max.}\ & \textstyle\sum_{j=1}^{5} c_j x_j \\
\text{s. t.}\ & \textstyle\sum_{j=1}^{5} a_{ij} x_j \leq b_i \, (i = 1, \dots, 4) \\
& x_j \geq 0 \ (j = 1, \dots, 5)
\end{aligned}
$$

How to use CPLEX?

- 新規プロジェクトの作成
  1. [ファイル(F)]ー[新規(N)]ー[OPLプロジェクト]を選択
  2. [プロジェクト名] を記入（例：ProductionPlanning）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 [ProductionPlanning.mod] → [pp.mod]
  - ✓ データファイル [ProductionPlanning.dat] → [ppex1.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?

- pp.mod

```
int i_max = ...;// 材料数
int j_max = ...;// 製品数

range I = 1..i_max;// 材料集合の範囲
range J = 1..j_max;// 製品集合の範囲

int a[I,J] = ...;// 各製品1単位あたりの必要材料数を表す行列
int b[I] = ...;// 材料i の所持数
int c[J] = ...;// 製品j の1単位あたり利益

dvar float+ x[J];// 製品j の生産量(非負)

maximize
  sum(j in J) c[j]*x[j];
subject to {
  forall(i in I) {
    sum(j in J) a[i,j]*x[j] <= b[i];
  }
}
```

How to use CPLEX?

- ppex1.dat

```
i_max = 4;
j_max = 5;

c = [6 7 3 10 5];
b = [80 75 95 70];

a = [
[3 0 1 3 1]
[1 2 0 3 2]
[0 4 2 5 0]
[2 1 0 1 2]
];
```

How to use CPLEX?

- 計算結果の確認([解]タブの中身)

最適値

```
// solution (optimal) with objective 316
// 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|                        = 45.5
// Maximum |slack|                    = 4
// Maximum |pi|                       = 2.2
// Maximum |red-cost|                 = 0.2
// Condition number of unscaled basis = 1.2e+01
//

x = [0 1 45.5 0 34.5];
```

optimal solution
最適解

生産計画をたてる

- 生産計画の最適化（例2）
  - 7種の材料 A, B, …, G がある
  - 5種の製品 $\alpha, \beta, \gamma, \delta, \varepsilon$ をつくる
  - 各材料の所有数，各製品を1単位作るのに必要な材料数，利益は右表
  - **総利益最大**の生産計画をたてよ

材料＼製品	α	β	γ	δ	ε	量
A	*2*	*0*	*1*	*3*	*1*	*80*
B	*3*	*1*	*0*	*3*	*1*	*75*
C	*0*	*2*	*3*	*5*	*0*	*95*
D	*3*	*2*	*0*	*1*	*2*	*70*
E	*1*	*0*	*3*	*0*	*3*	*60*
F	*0*	*3*	*0*	*2*	*4*	*80*
G	*1*	*5*	*0*	*4*	*1*	*90*
利益	*8*	*7*	*3*	*9*	*6*	

- 線形最適化問題として定式化し，CPLEXで解く
  - モデルファイル[pp.mod]は共通なので，データファイル[ppex2.dat]を作り解く

栄養問題を解く

栄養素＼食品	α	β	γ	δ	ε	必要量
A	2.5	0.7	1.3	3.1	4.1	70
B	11	21	5	13	12	300
C	0.6	4.2	2.5	5.1	0.8	50
D	23	16	15	0	12	150
コスト（円）	52	64	32	46	38	

- 栄養問題の最適化（例1）
  - 4種の栄養素 A〜D がある
  - 5種の食品 α〜ε がある
  - 各食品を1単位摂取すると得られる栄養素，コストは表の通り
  - 摂取必要量を満たし，コスト最小となる各食品の摂取量を知りたい

- 最適化問題の定式化（変数設定）
  - 変数 x_i … 食品i の摂取量（単位）
  - 変数ベクトル $\boxed{\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)}$

- 最適化問題の定式化（係数表記）
  - 摂取必要量を表す係数ベクトル $\boldsymbol{b}$
  - コストを表す係数ベクトル $\boldsymbol{c}$
  - 含有栄養素を表す係数行列 A

$$\boldsymbol{b} = (b_1 \quad b_2 \quad b_3 \quad b_4) = (70 \quad 300 \quad 50 \quad 150)$$

$$\boldsymbol{c} = (c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5) = (52 \quad 64 \quad 32 \quad 46 \quad 38)$$

$$\mathrm{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{pmatrix} = \begin{pmatrix} 2.5 & 0.7 & 1.3 & 3.1 & 4.1 \\ 11 & 21 & 5 & 13 & 12 \\ 0.6 & 4.2 & 2.5 & 5.1 & 0.8 \\ 23 & 16 & 15 & 0 & 12 \end{pmatrix}$$

栄養問題を解く

- 最適化問題の定式化(ベタ表記)

$$
\begin{aligned}
\text{min. } & 52x_1+64x_2+32x_3+46x_4+38x_5 \\
\text{s. t. } & 2.5x_1+0.7x_2+1.3x_3+3.1x_4+4.1x_5 \geqq 70 \\
& 11x_1+21x_2+5x_3+13x_4+12x_5 \geqq 300 \\
& 0.6x_1+4.2x_2+2.5x_3+5.1x_4+0.8x_5 \geqq 50 \\
& 23x_1+16x_2+15x_3+0x_4+12x_5 \geqq 150 \\
& x_1,\ x_2,\ x_3,\ x_4,\ x_5 \geqq 0
\end{aligned}
$$

$$
\boldsymbol{b} = (b_1 \quad b_2 \quad b_3 \quad b_4) = (70 \quad 300 \quad 50 \quad 150)
$$

$$
\boldsymbol{c} = (c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5) = (52 \quad 64 \quad 32 \quad 46 \quad 38)
$$

$$
A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{pmatrix} = \begin{pmatrix} 2.5 & 0.7 & 1.3 & 3.1 & 4.1 \\ 11 & 21 & 5 & 13 & 12 \\ 0.6 & 4.2 & 2.5 & 5.1 & 0.8 \\ 23 & 16 & 15 & 0 & 12 \end{pmatrix}
$$

- 最適化問題の定式化(Σ表記)

$$
\begin{aligned}
\text{min.} & \sum_{j=1}^{5} c_j x_j \\
\text{s. t. } & \sum_{j=1}^{5} a_{ij} x_j \geq b_i \ (i = 1, \dots, 4) \\
& x_j \geq 0 \ (j = 1, \dots, 5)
\end{aligned}
$$

How to use CPLEX?

- 新規プロジェクトの作成
  ① [ファイル(F)]ー[新規(N)]ー[OPLプロジェクト]を選択
  ② [プロジェクト名] を記入（例：Nutrients）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 [Nutrients.mod] → [nt.mod]
  - ✓ データファイル [Nutrients.dat] → [ntex1.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?

- nt.mod

```
int i_max = ...;// 栄養素数
int j_max = ...;// 食品数

range I = 1..i_max;// 栄養素集合の範囲
range J = 1..j_max;// 食品集合の範囲

float a[I,J] = ...;// 各食品1単位あたりの栄養素含有量
float c[J] = ...;// 各食品1単位あたりの摂取カロリー
float b[I] = ...;// 各栄養素の摂取必要量

dvar float+ x[J];// 食品j の摂取量（非負）

minimize
  sum(j in J) c[j]*x[j];
subject to {
  forall(i in I) {
    sum(j in J) a[i,j]*x[j] >= b[i];
  }
}
```

How to use CPLEX?

- ntex1.dat

```
i_max = 4;
j_max = 5;

c = [52 64 32 46 38];
b = [70 300 50 150];

a = [
[2.5 0.7 1.3 3.1 4.1]
[11 21 5 13 12]
[0.6 4.2 2.5 5.1 0.8]
[23 16 15 0 12]
];
```

How to use CPLEX?

- 計算結果の確認([解]タブの中身)

最適値

```
// solution (optimal) with objective 960.810287123481
// Quality There are no bound infeasibilities.
// There are no reduced-cost infeasibilities.
// Max. unscaled (scaled) Ax-b resid.          = 2.84217e-14 (1.77636e-15)
// Max. unscaled (scaled) c-B'pi resid.        = 7.10543e-15 (7.10543e-15)
// Max. unscaled (scaled) |x|                  = 13.0703 (13.0703)
// Max. unscaled (scaled) |slack|              = 72.681 (4.54256)
// Max. unscaled (scaled) |pi|                 = 2.72066 (43.5306)
// Max. unscaled (scaled) |red-cost|           = 18.6449 (26.8182)
// Condition number of scaled basis            = 6.6e+00
//

x = [0 4.1148 0 4.365 13.07];
```

optimal solution
最適解

栄養問題を解く

- 栄養問題の最適化（例2）　※データはExcelファイル
  - 18種の栄養素がある
  - 23種の食品がある
  - 各食品を1単位摂取すると得られる栄養素，カロリーはデータ表の通り
  - 摂取必要量を満たし，コスト最小の食品摂取量を求めよ

- 線形最適化問題として定式化し，CPLEXで解く
  - モデルファイル[nt.mod]は共通なので，データファイル[ntex2.dat]を作り解く

多品種輸送問題を解く

- 多品種輸送問題の最適化（例1）
  - 3つの工場で4種の製品A,B,C,Dを作っている．ただし，工場1はBとD，工場2はA,B,C，工場3はB,C,D のみを生産できる
  - 各工場の生産可能量は製品の種類に関係なく，それぞれ最大3000個
  - 5人の顧客に必要量を輸送する．輸送コスト最小となる輸送計画をたてたい

＜各工場で生産可能な製品と合計生産可能量＞

工場＼製品	*A*	*B*	*C*	*D*	生産可能量
1	—	*OK*	—	*OK*	*3000*
2	*OK*	*OK*	*OK*	—	*3000*
3	—	*OK*	*OK*	*OK*	*3000*

＜製品1単位あたり輸送コストと需要＞

	工場輸送コスト			製品需要			
顧客	*1*	*2*	*3*	*A*	*B*	*C*	*D*
α	*4*	*6*	*9*	*80*	*85*	*300*	*6*
β	*5*	*4*	*7*	*270*	*160*	*400*	*7*
γ	*6*	*3*	*4*	*250*	*130*	*350*	*4*
δ	*8*	*5*	*3*	*160*	*60*	*200*	*3*
ε	*10*	*8*	*4*	*180*	*40*	*150*	*5*

- 最適化問題の定式化（変数/係数）
  - 変数x_{ijk}：工場j→顧客i の製品k輸送量
  - 生産可能量の係数ベクトル $\boldsymbol{b}$
  - 輸送コストを表す係数行列 $\boldsymbol{C}$
  - 需要を表す係数行列 $\boldsymbol{D}$

多品種輸送問題を解く

➢ 最適化問題の定式化(ベタ表記)

min. $4(x_{111}+x_{112}+x_{113}+x_{114}) + \dots + 4(x_{531}+x_{532}+x_{533}+x_{534})$

s. t. $x_{111}+x_{121}+x_{131}=80,\ x_{112}+x_{122}+x_{132}=85$

$x_{113}+x_{123}+x_{133}=300,\ x_{114}+x_{124}+x_{134}=6$

…

$(x_{111}+x_{112}+x_{113}+x_{114})+\dots+(x_{511}+x_{512}+x_{513}+x_{514}) \leqq 3000$

$(x_{121}+x_{122}+x_{123}+x_{124})+\dots+(x_{521}+x_{522}+x_{523}+x_{524}) \leqq 3000$

$(x_{131}+x_{132}+x_{133}+x_{134})+\dots+(x_{531}+x_{532}+x_{533}+x_{534}) \leqq 3000$

$(x_{111}+x_{113})+\dots+(x_{511}+x_{513})=0$

$(x_{124})+\dots+(x_{524})=0$

$(x_{131})+\dots+(x_{531})=0$

使わない変数を0に(各工場で生産してない製品を0に)

$x_{111},\dots,x_{534} \geqq 0$

➢ 最適化問題の定式化(Σ表記)

$$\min. \sum_{i=1}^{5}\sum_{j=1}^{3}\sum_{k=1}^{4} c_{ij}x_{ijk}$$

$$\text{s. t. } \sum_{j=1}^{3} x_{ijk} = d_{ik}\ (i=1,\dots,5;\ k=1,\dots,4)$$

$$\sum_{i=1}^{5}\sum_{k=1}^{4} x_{ijk} \leq b_j\ (j=1,\dots,3)$$

$$\sum_{i=1}^{5}\sum_{k=1}^{4} m_{jk}x_{ijk} = 0\ (j=1,\dots,3)$$

$$x_{ijk} \geq 0\ (i=1,\dots,5;\ j=1,\dots,3;\ k=1,\dots,4)$$

$$\boldsymbol{b} = (b_1 \quad b_2 \quad b_3) = (3000 \quad 3000 \quad 3000)$$

$$\mathrm{C} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ c_{41} & c_{42} & c_{43} \\ c_{51} & c_{52} & c_{53} \end{pmatrix} = \begin{pmatrix} 4 & 6 & 9 \\ 5 & 4 & 7 \\ 6 & 3 & 4 \\ 8 & 5 & 3 \\ 10 & 8 & 4 \end{pmatrix}$$

$$\mathrm{D} = \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \\ d_{51} & d_{52} & d_{53} & d_{54} \end{pmatrix} = \begin{pmatrix} 80 & 85 & 300 & 6 \\ 270 & 160 & 400 & 7 \\ 250 & 130 & 350 & 4 \\ 160 & 60 & 200 & 3 \\ 180 & 40 & 150 & 5 \end{pmatrix}$$

$$\mathrm{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

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：MultiTransport）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 [MultiTransport.mod] → [mt.mod]
  - ✓ データファイル [MultiTransport.dat] → [mtex1.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?

- mt.mod

```
int i_max = ...;// 顧客数
int j_max = ...;// 工場数
int k_max = ...;// 製品数

range I = 1..i_max;// 顧客集合の範囲
range J = 1..j_max;// 工場集合の範囲
range K = 1..k_max;// 製品集合の範囲

int c[I,J] = ...; // 工場 j → 顧客 i への製品1単位あたり輸送コスト
int d[I,K] = ...; // 顧客 i の 製品 k の需要量
int b[J] = ...;   // 工場 j の生産可能合計量
int m[J,K] = ...; // 工場 j の 製品 k の生産有無を表す係数行列

dvar float+ x[I,J,K];// 工場 j → 顧客 i への 製品 k の輸送量(非負)

minimize
  sum(i in I) sum(j in J) sum(k in K) c[i,j]*x[i,j,k];
subject to {
  forall(i in I) {
    forall(k in K) {
      sum(j in J) x[i,j,k] == d[i,k];
    }
  }
  forall(j in J) {
    sum(i in I) sum(k in K) x[i,j,k] <= b[j];
    sum(i in I) sum(k in K) m[j,k]*x[i,j,k] == 0;
 }
}
```

How to use CPLEX?

- mtex1.dat

```
i_max = 5;
j_max = 3;
k_max = 4;

b = [3000 3000 3000];
c = [
[4 6 9]
[5 4 7]
[6 3 4]
[8 5 3]
[10 8 4]
];
d = [
[ 80  85 300 6]
[270 160 400 7]
[250 130 350 4]
[160  60 200 3]
[180  40 150 5]
];
m = [
[1 0 1 0]
[0 0 0 1]
[1 0 0 0]
];
```

How to use CPLEX?

工場＼製品	A	B	C	D	生産可能量
1	―	OK	―	OK	3000
2	OK	OK	OK	―	3000
3	―	OK	OK	OK	3000

- 計算結果の確認([解]タブ)

最適値

```
// solution (optimal) with objective 12014
// 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|                        = 400
// Maximum |slack|                    = 2902
// Maximum |pi|                       = 8
// Maximum |red-cost|                 = 8
// Condition number of unscaled basis = 1.4e+02
x = [[[  0  85   0   6]      …工場1(B=85,D=6)
      [ 80   0 300   0]      …工場2(A=80,C=300)
      [  0   0   0   0]]     …工場3(0)
     [[  0   0   0   7]
      [270 160 400   0]
      [  0   0   0   0]]
     [[  0   0   0   0]
      [250 130 350   0]
      [  0   0   0   4]]
     [[  0   0   0   0]
      [160   0   0   0]
      [  0  60 200   3]]
     [[  0   0   0   0]
      [180   0   0   0]
      [  0  40 150   5]]];
```

顧客αへの輸送

Optimal solution 最適解

顧客	製品需要 A	B	C	D
α	80	85	300	6
β	270	160	400	7
γ	250	130	350	4
δ	160	60	200	3
ε	180	40	150	5

多品種輸送問題を解く

- 多品種輸送問題の最適化（例2）　※データはExcelファイル
  - 〇つの工場で△種の製品を作っている．ただし，各工場で生産できる製品の種類は異なる
  - 各工場の生産可能量は製品の種類に関係なく，それぞれ最大◇個
  - □人の顧客に必要量を輸送する．輸送コスト最小となる輸送計画をたてよ

- 線形最適化問題として定式化し，CPLEXで解く
  - モデルファイル[mt.mod]は共通なので，データファイル[mtex2.dat]を作り解く