知の探究

4. 線形計画法

堀田 敬介

2018年11月13日（火）

線形計画問題とは？

- 線形計画問題(Linear Programming Problem)
  - 線形（1次）等式・不等式系であらわされる条件のもとで，線形（1次）の目的関数を最大・最小化する形式の最適化問題

$$\begin{aligned} \text{min.} \quad & 2x_1 + x_2 + 2x_3 + x_4 + 3x_5 \\ \text{s.t.} \quad & x_1 + 2x_3 + x_5 \geq 5 \\ & 9x_1 + 2x_2 + x_4 + 4x_5 \geq 1 \\ & x_2 + 5x_3 + x_5 \geq 3 \\ & x_1 + 3x_3 + x_5 \geq 2 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

目的関数 objective function

制約条件 constraints

非負条件 nonnegativity

- 線形計画問題を解くための主な手法 algorithm
  - 単体法　simplex method,　G.B.Dantzig(1947)
  - 内点法　interior point method,　N.Karmarkar (1984)
  - （楕円体法　ellipsoid method,　Yudin, A.S.Nemirovskii(1976), Khachiyan(1979)）

線形計画問題をソルバーで解く

- 問題の定式化

min. $2x_1 + x_2 + 2x_3 + x_4 + 3x_5$ 目的関数 objective function

s.t.

$$x_1 + 2x_3 + x_5 \geq 5$$
$$9x_1 + 2x_2 + x_4 + 4x_5 \geq 1$$
$$x_2 + 5x_3 + x_5 \geq 3$$
$$x_1 + 3x_3 + x_5 \geq 2$$

制約条件 constraints

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

非負条件 nonnegativity

- LPファイル形式にする（テキストエディタで書く）

```
minimize
        2 x1 +  x2 + 2 x3 +  x4 + 3 x5
subject to
         x1 +       2 x3       +  x5  >= 5
       9 x1 + 2 x2        +  x4 + 4 x5  >= 1
              x2  + 5 x3        +  x5  >= 3
         x1       + 3 x3        +  x5  >= 2
end
```

目的関数 objective function

制約条件 constraints

※非負条件は書かない

非負条件 nonnegativity

→ ファイル名「〇〇〇.lp」で保存

※数値・変数・記号の間は全て「半角スペース」が必要

線形計画問題をソルバーで解く

- よく知られたソルバー（有料[商用]・無償[非商用]）
  - gurobi (Zonghao Gu, Edward Rothberg, Robert Bixby)
  - cplex (IBM ILOG CPLEX)
  - Xpress (FICO, MSI)
  - SCIP (ZIB, Solving Constraint Integer Programs)
  - lp solve
  - GLPK (Gnu Linear Programming Kit)
  - Excel solver (Microsoft)
  - etc.
  - Excel solver以外はlpファイルを読み込んで最適化可

線形計画問題をソルバーで解く

- 準備
  - 「lpファイル」を作成・保存する

※ファイル名は半角英数にし，拡張子は lp とする
※テキストエディタで保存時に，ファイルの種類を「すべてのファイル（*.*）」にする
例）ファイル名「example.lp」
※ファイル名が「example.lp.txt」となってしまったら，ファイル名の変更で「.txt」を削除する

  - 「コマンド プロンプト」を起動する
    - ✓ windows: 検索窓で「cmd [Enter]」
  - 「lpファイル」が保存されているフォルダへ移動する
    - ✓ Y:¥> cd xxx （※ cd = change directory）

1)「エクスプローラ」で「lpファイル」が保存されているフォルダを表示し，
アドレスが書かれている欄のフォルダ名を右クリック
→ 「アドレスをテキストとしてコピー」を選ぶ
2)「コマンドプロンプト」で「cd 」と書いた後右クリック→ 貼り付け→ [Enter]

線形計画問題をソルバーで解く

- gurobiで最適化（解く）

Y:¥xxx> gurobi [Enter] ← 「gurobi」コマンドでgurobi を起動する

LPファイル読込

gurobi> m=read(“xxx.lp”)

最適化（解く）

gurobi> m.optimize()

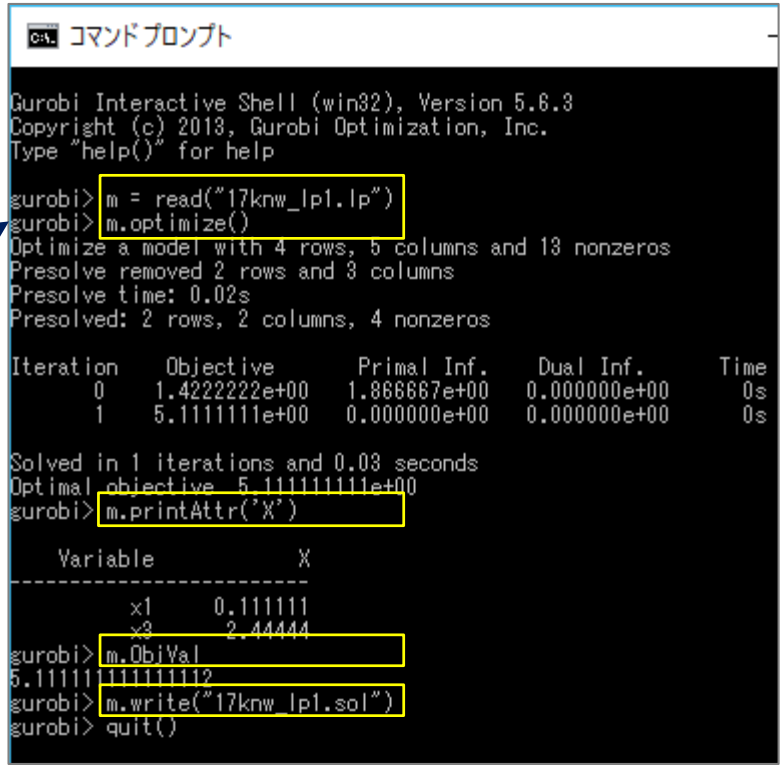
解の表示

gurobi> m.printAttr(‘X’)

gurobi> m.ObjVal

解をファイル保存

gurobi> m.write(“xxx.sol”)



線形計画問題をソルバーで解く

- cplexで最適化（解く）

Y:¥xxx> cplex [Enter]

「cplex」コマンドでcplex を起動する

LPファイル読込

CPLEX> read xxx.lp

問題の表示

CPLEX> d p a

最適化（解く）

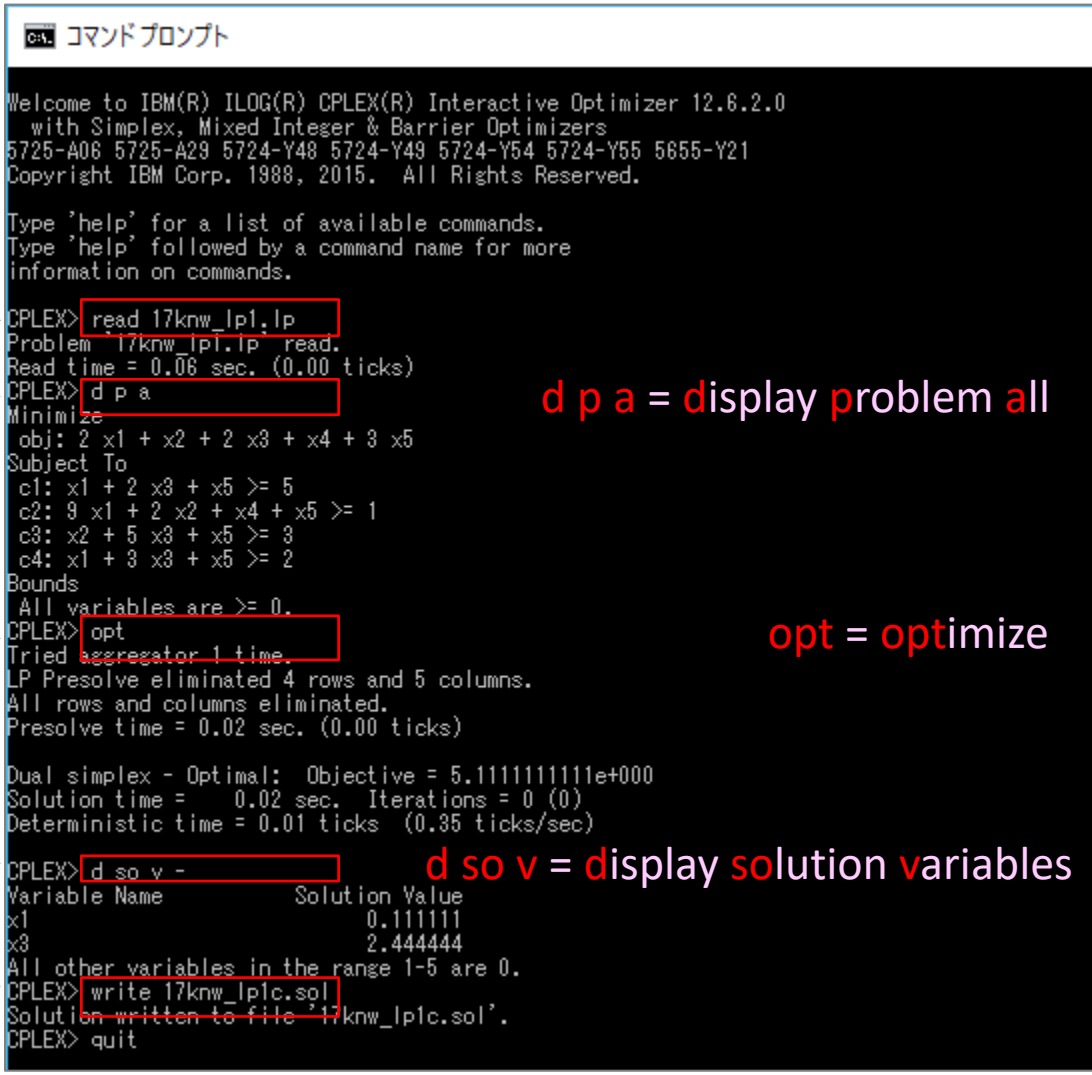
CPLEX> opt

解の表示

CPLEX> d so v －

解をファイル保存

CPLEX> write xxx.sol



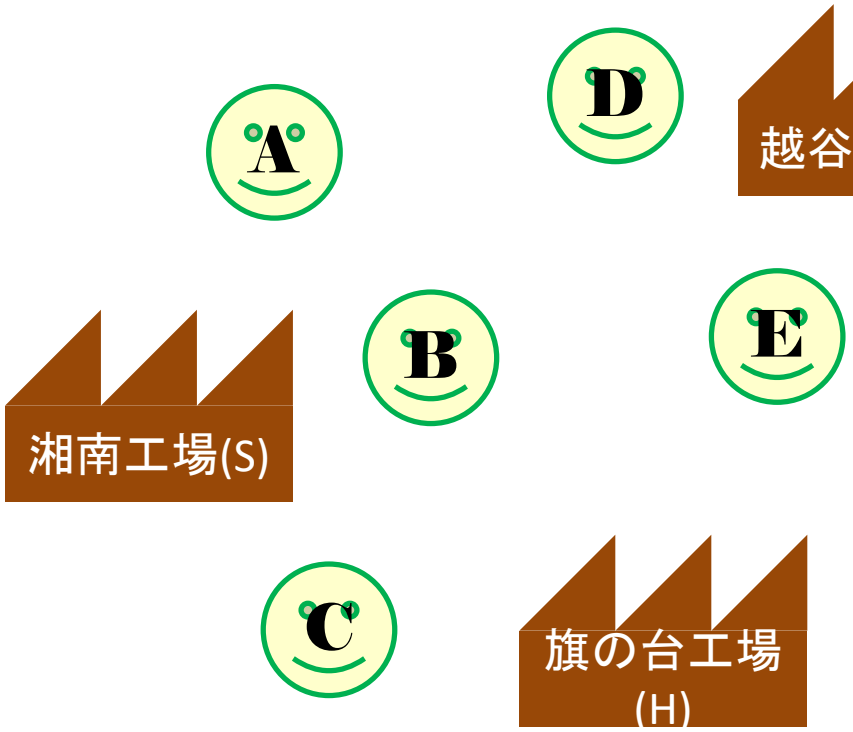
輸送問題

問）文教重工には3工場（湘南・越谷・旗の台）あり，製品を供給（製品生産量）できる

顧客は5人いて，需要（製品を欲しい量）がある

3工場から5人の顧客それぞれへの単位あたり輸送コストは表の通り

輸送コストが最小となる配送計画をたてよ



- ✓ 工場の供給量
- ✓ 顧客の需要量
- ✓ 工場から顧客へ製品を1単位

配送するのにかかる輸送コスト表

	需要	50	80	60	70	40
供給	工場＼顧客	A	B	C	D	E
120	湘南(S)	3	2	4	5	8
130	越谷(K)	5	6	5	3	2
70	旗の台(H)	7	3	1	2	3

線形計画法

- 線形計画法 Linear Programming, LP
- 問題のモデル化
  - 目的：輸送コストを最小
  - 条件1：顧客の需要を満たす
  - 条件2：工場の出荷量は供給量まで
  - 条件3：輸送量は非負

目的関数 objective function

制約条件 constraints

非負条件 nonnegativity

- 変数設定

x_{ij} ： 工場i →顧客jへの輸送量

ex) $x_{SB} = 30$: 湘南工場(S)から顧客Bへ製品を30輸送する
その輸送コスト: $2 \times 30 = 60$

	需要	50	80	60	70	40
供給	工場＼顧客	A	B	C	D	E
120	湘南(S)	3	2	4	5	8
130	越谷(K)	5	6	5	3	2
70	旗の台(H)	7	3	1	2	3

線形計画法

	需要	50	80	60	70	40
供給	工場＼顧客	A	B	C	D	E
120	湘南(S)	3	2	4	5	8
130	越谷(K)	5	6	5	3	2
70	旗の台(H)	7	3	1	2	3

- 問題のモデル化
  - 目的：輸送コストを最小
  - 条件1：顧客の需要を満たす
  - 条件2：工場の出荷量は供給量まで
  - 条件3：輸送量は非負
- 問題の定式化（LPファイル形式）

minimize

$$3\,x_{SA} + 2\,x_{SB} + 4\,x_{SC} + 5\,x_{SD} + 8\,x_{SE}$$
$$+ 5\,x_{KA} + 6\,x_{KB} + 5\,x_{KC} + 3\,x_{KD} + 2\,x_{KE}$$
$$+ 7\,x_{HA} + 3\,x_{HB} + 1\,x_{HC} + 2\,x_{HD} + 3\,x_{HE}$$

subject to

$$x_{SA} + x_{KA} + x_{HA} = 50$$
$$\ldots$$
$$x_{SA} + x_{SB} + x_{SC} + x_{SD} + x_{SE} <= 120$$
$$\ldots$$

目的関数 objective function

制約条件 constraints

非負条件 nonnegativity

線形計画法

- 問題の定式化

	需要	50	80	60	70	40
供給	工場＼顧客	A	B	C	D	E
120	湘南(S)	3	2	4	5	8
130	越谷(K)	5	6	5	3	2
70	旗の台(H)	7	3	1	2	3

minimize

$$3\,x_{SA} + 2\,x_{SB} + 4\,x_{SC} + 5\,x_{SD} + 8\,x_{SE} + 5\,x_{KA} + 6\,x_{KB} + 5\,x_{KC} + 3\,x_{KD} + 2\,x_{KE} + 7\,x_{HA} + 3\,x_{HB} + 1\,x_{HC} + 2\,x_{HD} + 3\,x_{HE}$$

目的関数 objective function

subject to

$$x_{SA} + x_{KA} + x_{HA} = 50$$
$$x_{SB} + x_{KB} + x_{HB} = 80$$
$$x_{SC} + x_{KC} + x_{HC} = 60$$
$$x_{SD} + x_{KD} + x_{HD} = 70$$
$$x_{SE} + x_{KE} + x_{HE} = 40$$
$$x_{SA} + x_{SB} + x_{SC} + x_{SD} + x_{SE} <= 120$$
$$x_{KA} + x_{KB} + x_{KC} + x_{KD} + x_{KE} <= 130$$
$$x_{HA} + x_{HB} + x_{HC} + x_{HD} + x_{HE} <= 70$$

制約条件 constraints

end

非負条件 nonnegativity

※LPファイル形式では書かない

→ ファイル名「ex.lp」で保存（LPファイル）

線形計画法

- gurobiで最適化（解く）

Y:¥LP>gurobi [Enter]

LPファイル読込

gurobi> m=read(“ex.lp”)

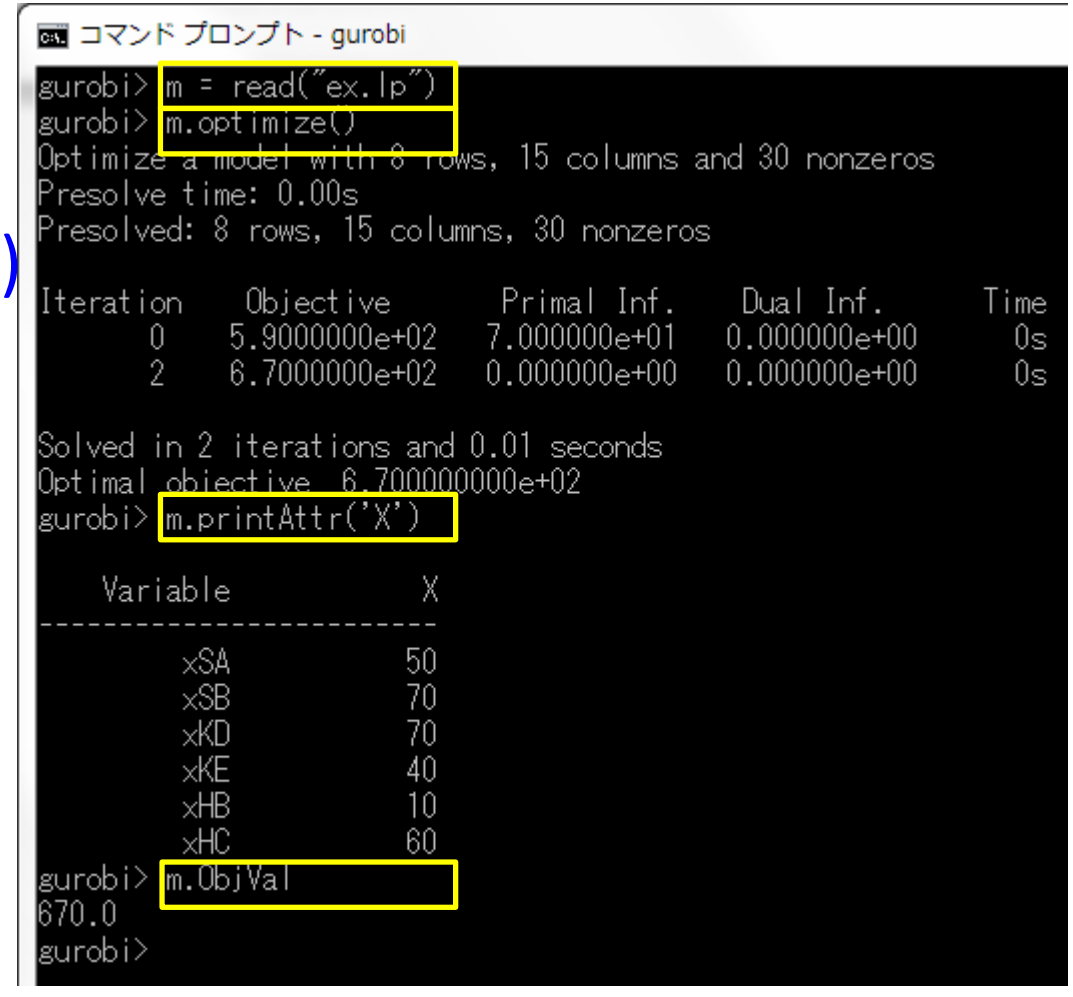
最適化（解く）

gurobi> m.optimize()

解の表示

gurobi> m.printAttr(‘X’)

gurobi> m.ObjVal



線形計画法

- cplexで最適化（解く）

Y:¥LP>cplex [Enter]

LPファイル読込

CPLEX> read ex.lp

問題の表示

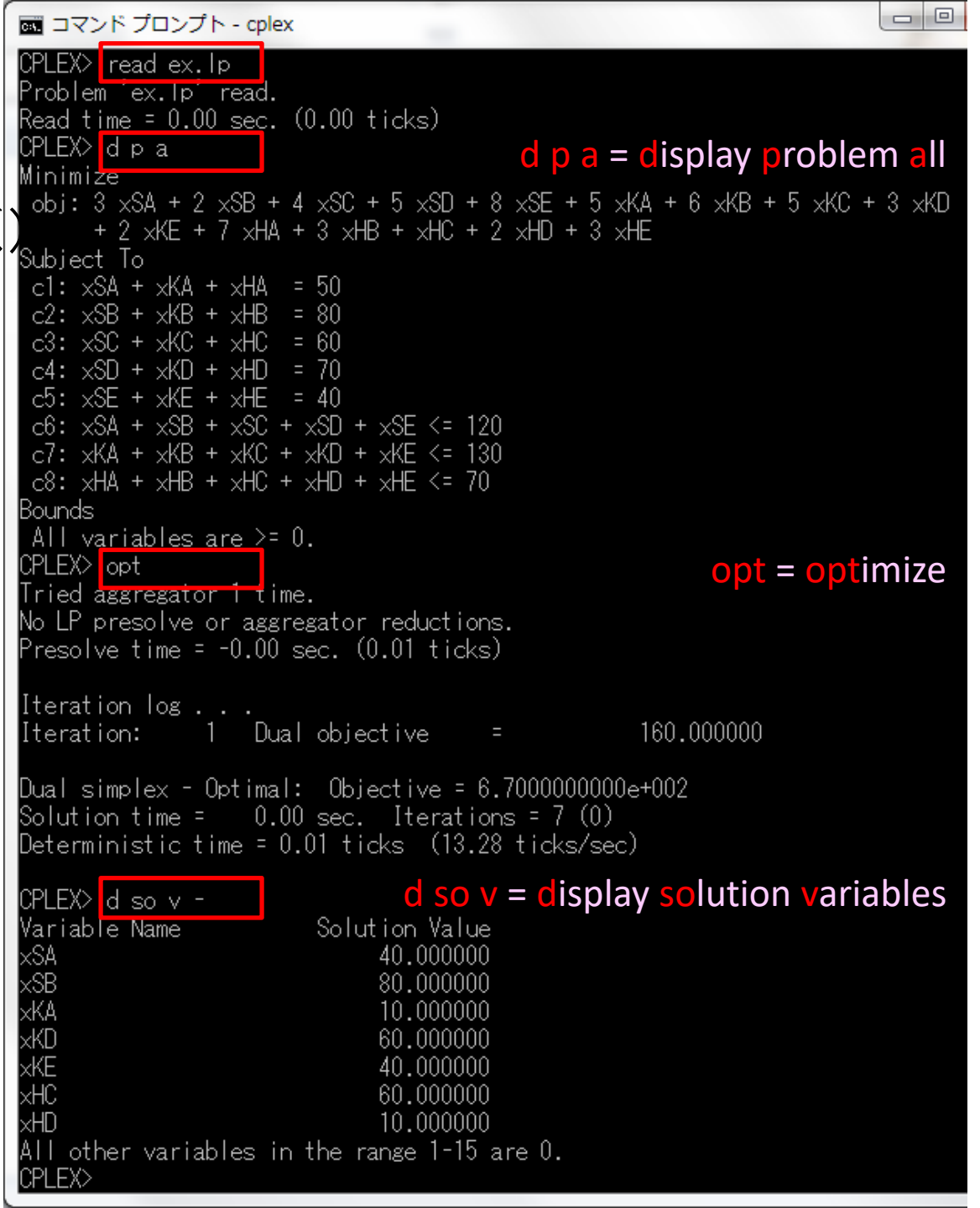
CPLEX> d p a

最適化（解く）

CPLEX> opt

解の表示

CPLEX> d so v -



輸送問題

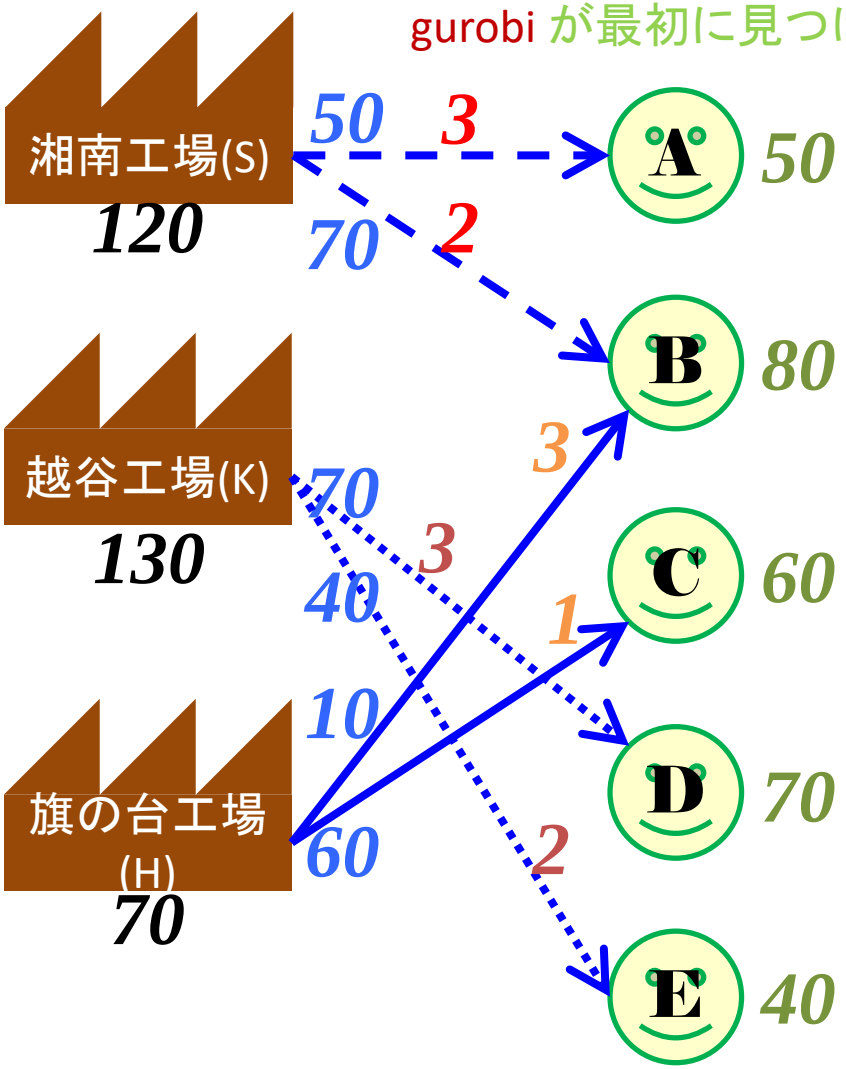
最適値
$670 = 6.7 \times 10^2$

```
Optimal objective  6.70000000e+02
gurobi> m.printAttr('X')

    Variable            X
-------------------------
         xSA           50
         xSB           70
         xKD           70
         xKE           40
         xHB           10
         xHC           60
gurobi> m.ObjVal
670.0
```

最適解・最適値の評価・検証）

gurobi が最初に見つけた解



- ✓ 工場の供給量
- ✓ 顧客の需要量
- ✓ 工場から顧客へ製品を1単位

配送するのにかかる輸送コスト表

	需要	50	80	60	70	40
供給	工場＼顧客	A	B	C	D	E
120	湘南(S)	3	2	4	5	8
130	越谷(K)	5	6	5	3	2
70	旗の台(H)	7	3	1	2	3

輸送問題

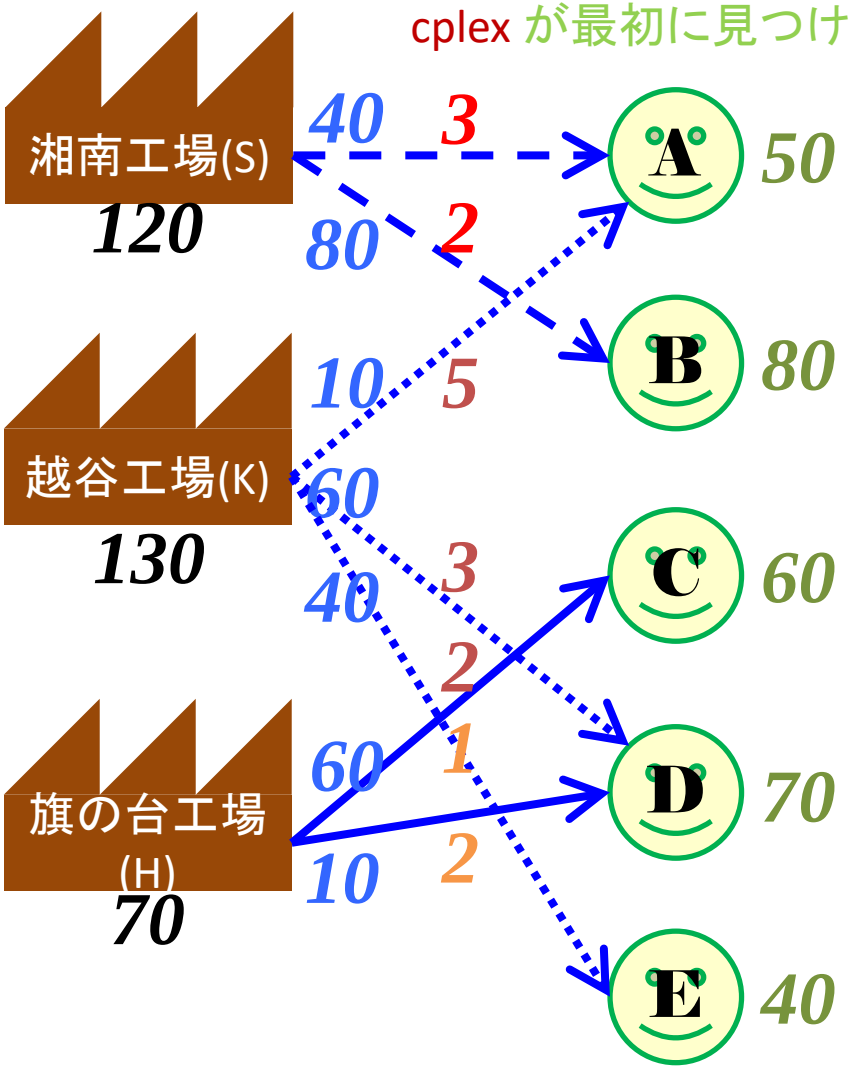
最適値
$670 = 6.7 \times 10^2$

```
Dual simplex - Optimal:  Objective = 6.7000000000e+002
Solution time =    0.00 sec.  Iterations = 7 (0)
Deterministic time = 0.01 ticks  (13.28 ticks/sec)

CPLEX> d so v -
Variable Name           Solution Value
xSA                          40.000000
xSB                          80.000000
xKA                          10.000000
xKD                          60.000000
xKE                          40.000000
xHC                          60.000000
xHD                          10.000000
All other variables in the range 1-15 are 0.
```

最適解・最適値の評価・検証）

cplex が最初に見つけた解



- ✓ 工場の供給量
- ✓ 顧客の需要量
- ✓ 工場から顧客へ製品を1単位

配送するのにかかる輸送コスト表

	需要	50	80	60	70	40
供給	工場＼顧客	A	B	C	D	E
120	湘南(S)	3	2	4	5	8
130	越谷(K)	5	6	5	3	2
70	旗の台(H)	7	3	1	2	3

栄養問題 diet problem

- 例題

『なめがやわーるど』では，「神秘ケーキ」「魅惑菓子」「苦渋野菜」「過酸果物」の4つの食べ物と，「だんはっく」「ガルジウム」「ヒタビン」という3つの栄養素，3つの食品含有物「糖分」「塩分」「ガロリー」が存在する．4つの食べ物は3つの栄養素と3つの食品含有物を，1単位当たり各々下表に示す量だけ含む

栄養素	神秘ケーキ	魅惑菓子	苦渋野菜	過酸果物
だんはっく	3	1	4	2
ガルジウム	1	2	2	1
ヒタビン	2	1	2	5
糖分	7	5	3	4
塩分	1	2	4	8
ガロリー	40	50	55	20

『なめがやわーるど』人は，3栄養素を一日に各々最低50, 40, 60は摂取しないと死んでしまう！　また，糖分と塩分は各々一日に150を超えると過剰摂取で死んでしまう！！
ダイエットしたい花子さんのために，ガロリーを最小にする食べ物の量を教えて欲しい

【演習】 LPに定式化して Solver で求解せよ

栄養問題の定式化と求解

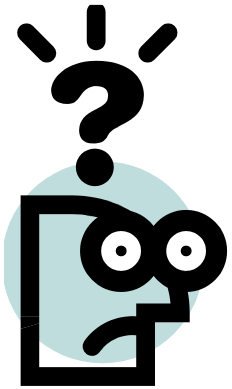
- 例題：定式化例

$$
\begin{aligned}
\text{min.} \quad & 40x_1 + 50x_2 + 55x_3 + 20x_4 \\
\text{s.t.} \quad & 3x_1 + x_2 + 4x_3 + 2x_4 \geq 50 \\
& x_1 + 2x_2 + 2x_3 + x_4 \geq 40 \\
& 2x_1 + x_2 + 2x_3 + 5x_4 \geq 60 \\
& 7x_1 + 5x_2 + 3x_3 + 4x_4 \leq 150 \\
& x_1 + 2x_2 + 4x_3 + 8x_4 \leq 150 \\
& x_1, x_2, x_3, x_4 \geq 0
\end{aligned}
$$

割当問題 assignment problem

- 例題

上司が10人の部下に仕事をまかせようとしている
仕事は全部で15種類ある（A,B,C,...,O）
10人の部下の内，4人は新人で6人はベテランである
各々の仕事をどの部下がやるかにより，上司は事前に5段階評価している（1,...,5）
ベテランは同時に2つまで仕事をまかせられる
新人は同時に1つまでしか仕事をまかせられず，どの仕事の最大評価も3（1,...,3）
評価値総和が最大になるように，各部下に仕事を割り振りたい



【演習】 LPに定式化して Solver で求解せよ

割当問題の定式化と求解

- 例題：定式化例

$$\text{min.} \quad 3x_{1A} + x_{1B} + x_{1C} + \cdots + 3x_{1N} + 3x_{1O} + \cdots + 4x_{10A} + 5x_{10B} + 5x_{10C} + \cdots + 3x_{10N} + 5x_{10O}$$

$$\text{s.t.} \quad \text{①} \begin{cases} x_{1A} + x_{1B} + x_{1C} + \cdots + x_{1O} \leq 1 \\ \cdots \\ x_{10A} + x_{10B} + x_{10C} + \cdots + x_{10O} \leq 2 \end{cases}$$

$$\text{②} \begin{cases} x_{1A} + x_{2A} + \cdots + x_{10A} = 1 \\ \cdots \\ x_{1O} + x_{2O} + \cdots + x_{10O} = 1 \end{cases}$$

$$x_{1A}, \ldots, x_{10O} \in \{0,1\}$$

制約式①は，部下達は1ないし2の仕事を担当できる
制約式②は，各仕事は必ず誰かが担当する
0-1制約は $0 \leqq x_{ij} \leqq 1$ にして解いてよい（補足：完全単模行列）

【補足】

- 単模行列(unimodular matrix)
  - def) 整数正方行列$A \in R^{n \times m}$が単模行列 $\Leftrightarrow$ det$A$=1 or -1
  - th) 単模行列の逆行列も，整数行列で単模行列
- 完全単模行列(totally unimodular matrix)
  - def) 整数行列$A \in R^{m \times n}$が完全単模行列

    $\Leftrightarrow$ 任意の小行列式の値が 0 or 1 or -1
  - th) 完全単模行列の各要素は 0 or 1 or -1
  - ex) 有向グラフの接続行列は完全単模
  - ex) 無向グラフの接続行列が完全単模となる必要十分条件はグラフが2部であること（cf. 3点奇数サイクルのグラフはdet$A$=±2）
  - th) LP(P)が最適解をもち，係数行列Aが完全単模とする．bが整数ベクトルなら，(P)は整数最適解 $x \in Z^n$をもつ

$$(P) \quad max. \quad c^t x$$
$$s.t. \quad Ax = b$$
$$x \geqq 0$$

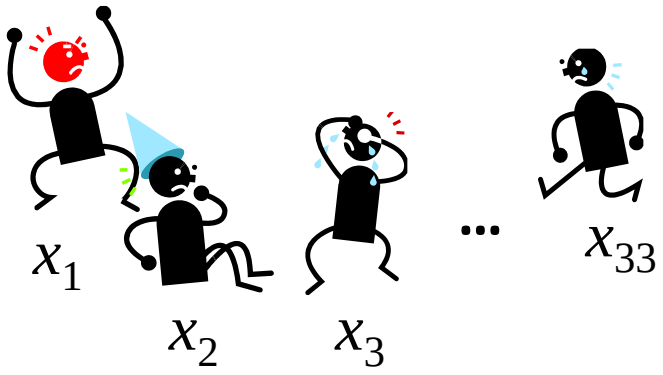
proof) (P)の基底Bに対する基底解は$(B^{-1}b, 0)$
Aが完全単模なので，Bは単模行列
よって，B^{-1}は整数行列．故に$B^{-1}b$は整数ベクトル■

クラス編成問題 student sectioning

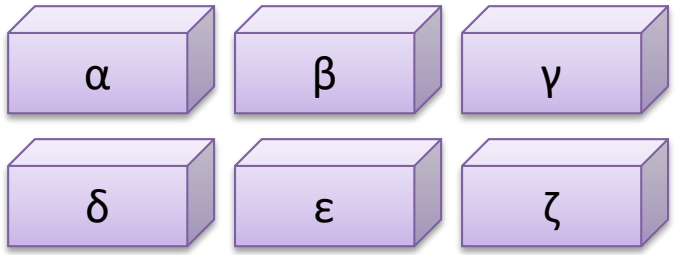
- 例題

33人の学生を6つのクラスに配属させたい
各学生は丁度1つのクラスに所属させ，配属しないという選択はないとする
各学生は6つのクラスへの希望を持っている（第1志望～第3志望）のみ
クラスには定員があり，全て6人である（容量6人×6クラス=36人で充分）
全学生の満足度総和が最大になるように学生をクラスへ配属させなさい

学生33人



6クラス（各定員6人）



【演習】 LPに定式化して Solver で求解せよ

クラス編成問題の定式化と求解

- 例題：定式化例

min. $-999x_{1\alpha} + 30x_{1\beta} + 100x_{1\gamma} - 999x_{1\delta} - 999x_{1\varepsilon} + 60x_{1\zeta}$

$$+ \cdots +$$

$$+30x_{1\alpha} + 100x_{1\beta} + 60x_{1\gamma} - 999x_{1\delta} - 999x_{1\varepsilon} - 999x_{1\zeta}$$

s.t.

① $x_{1\alpha} + x_{1\beta} + x_{1\gamma} + \cdots + x_{1\zeta} = 1$

$$\cdots$$

$$x_{33\alpha} + x_{33\beta} + x_{33\gamma} + \cdots + x_{33\zeta} = 1$$

② $x_{1\alpha} + x_{2\alpha} + \cdots + x_{33\alpha} \leq 6$

$$\cdots$$

$$x_{1\zeta} + x_{2\zeta} + \cdots + x_{33\zeta} \leq 6$$

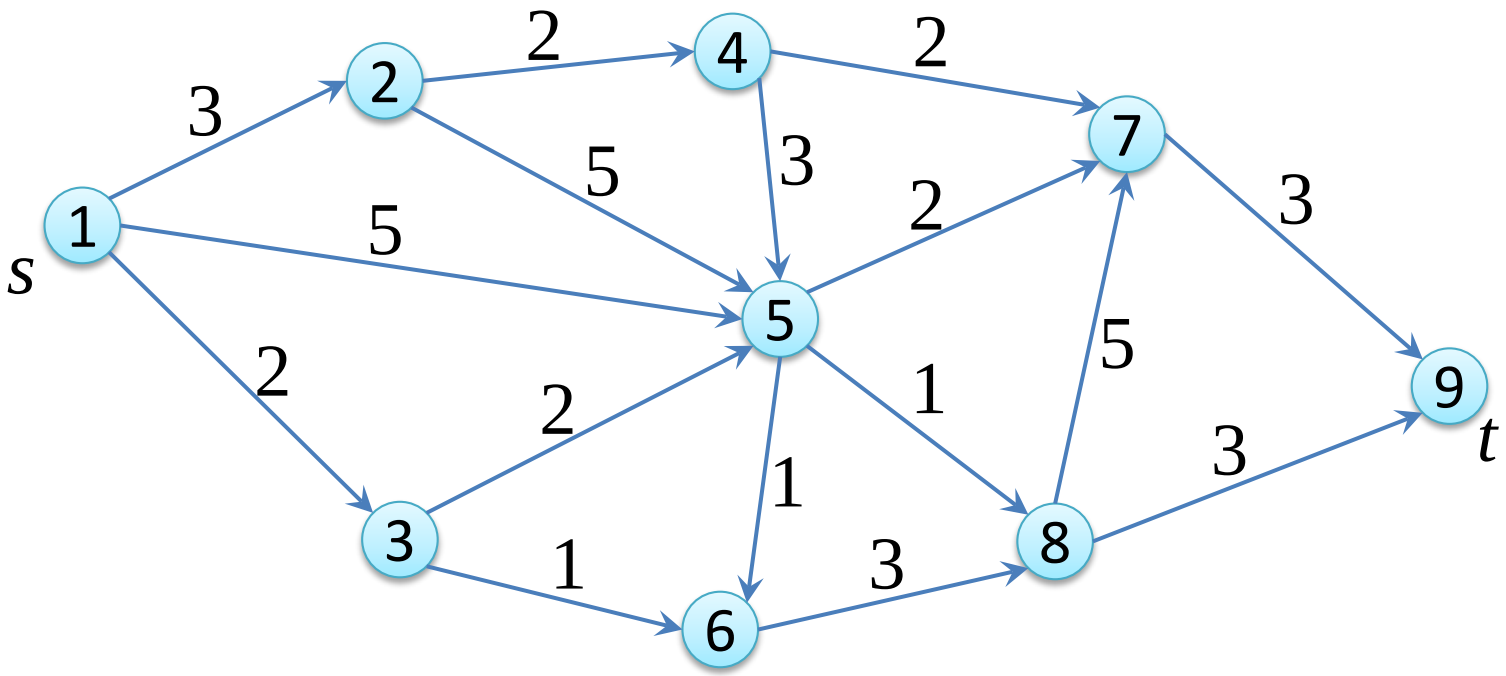
$$x_{1\alpha}, \ldots, x_{33\zeta} \in \{0,1\}$$

制約式①は，各学生は6クラスのどこか1か所に所属する
制約式②は，各クラスの定員は6人
0-1制約は $0 \leqq x_{ij} \leqq 1$ にして解いてよい（補足：完全単模行列）

最短路問題 shortest path problem

- 例題

グラフG=(V,E)と枝上のコスト（cost）が与えられている
スタート地点（点1）からゴール地点（点9）まで，コストの総和が最小となる路をもとめよ



【演習】 LPに定式化して Excel Solver で求解せよ
（LPファイルで定式化を書くより，Excel の方が定式化が楽）

最短路問題の定式化と求解

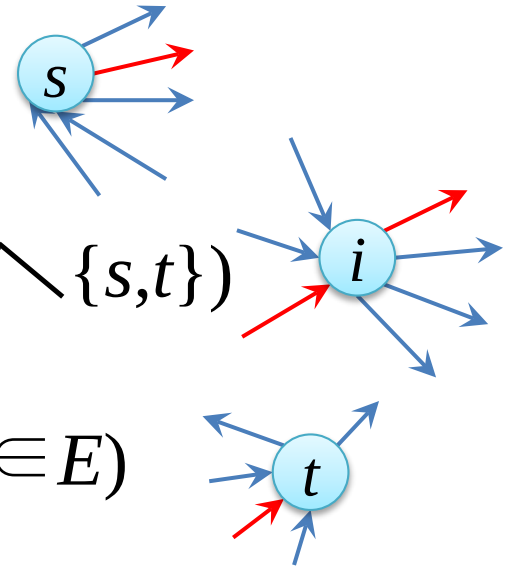
- 例題：定式化例

0-1変数x_{ij}は，枝$(i,j)\in E$について，
経路として使うなら1，使わないなら0

$$min. \quad \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$s.t. \quad \sum_{j\in V} x_{ij} - \sum_{j\in V} x_{ji} = \begin{cases} 1 & (i=s) \\ 0 & (\forall i \in V \setminus \{s,t\}) \\ -1 & (i=t) \end{cases}$$

$$x_{ij} \in \{0,1\} \qquad (\forall (i,j)\in E)$$



点iからの流出量の和

点iへの流入量の和

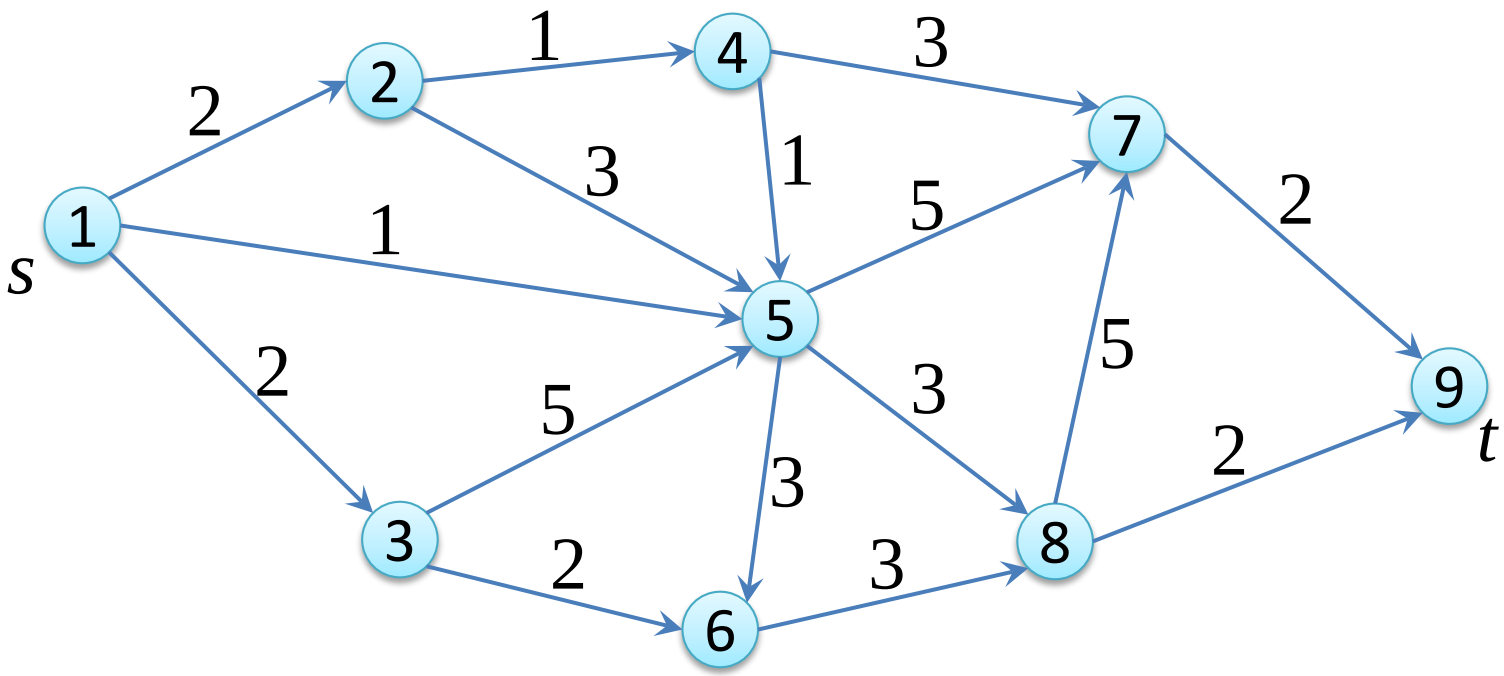
制約式は，「点iからの流出量の和」と「点iへの流入量の和」との差に関するもので
点iがスタート地点（i=s）なら1，ゴール地点（i=t）なら-1，それ以外なら0とする
（スタートは流出のみ，途中は流入分すべて流出し，ゴールは流入のみということ）

0-1制約「$x_{ij}\in\{0,1\}$」は線形緩和「$0\leqq x_{ij}\leqq 1$」して解いてよい

最大流問題 maximum flow problem

- 例題

グラフG=(V,E)と枝$(i,j)\in E$上の容量（capacity）u_{ij}が与えられている
スタート地点（点1）からゴール地点（点9）まで，最大流量を流せ



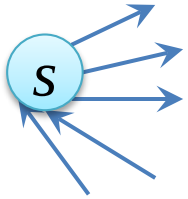
【演習】 LPに定式化して Excel Solver で求解せよ
（LPファイルで定式化を書くより，Excel の方が定式化が楽）

最大流問題の定式化と求解

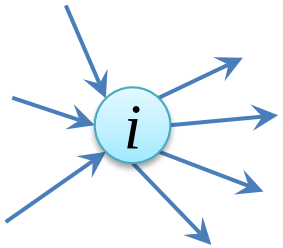
- 例題：定式化例

実数変数x_{ij}は，枝$(i,j) \in E$に流れる流量

$$max. \quad \sum_{j \in V} x_{sj} - \sum_{j \in V} x_{js}$$



$$s.t. \quad \sum_{j \in V} x_{ij} - \sum_{j \in V} x_{ji} = 0 \quad (\forall i \in V \setminus \{s,t\})$$



$$0 \le x_{ij} \le u_{ij} \quad (\forall (i,j) \in E)$$

点iからの流出量の和

点iへの流入量の和

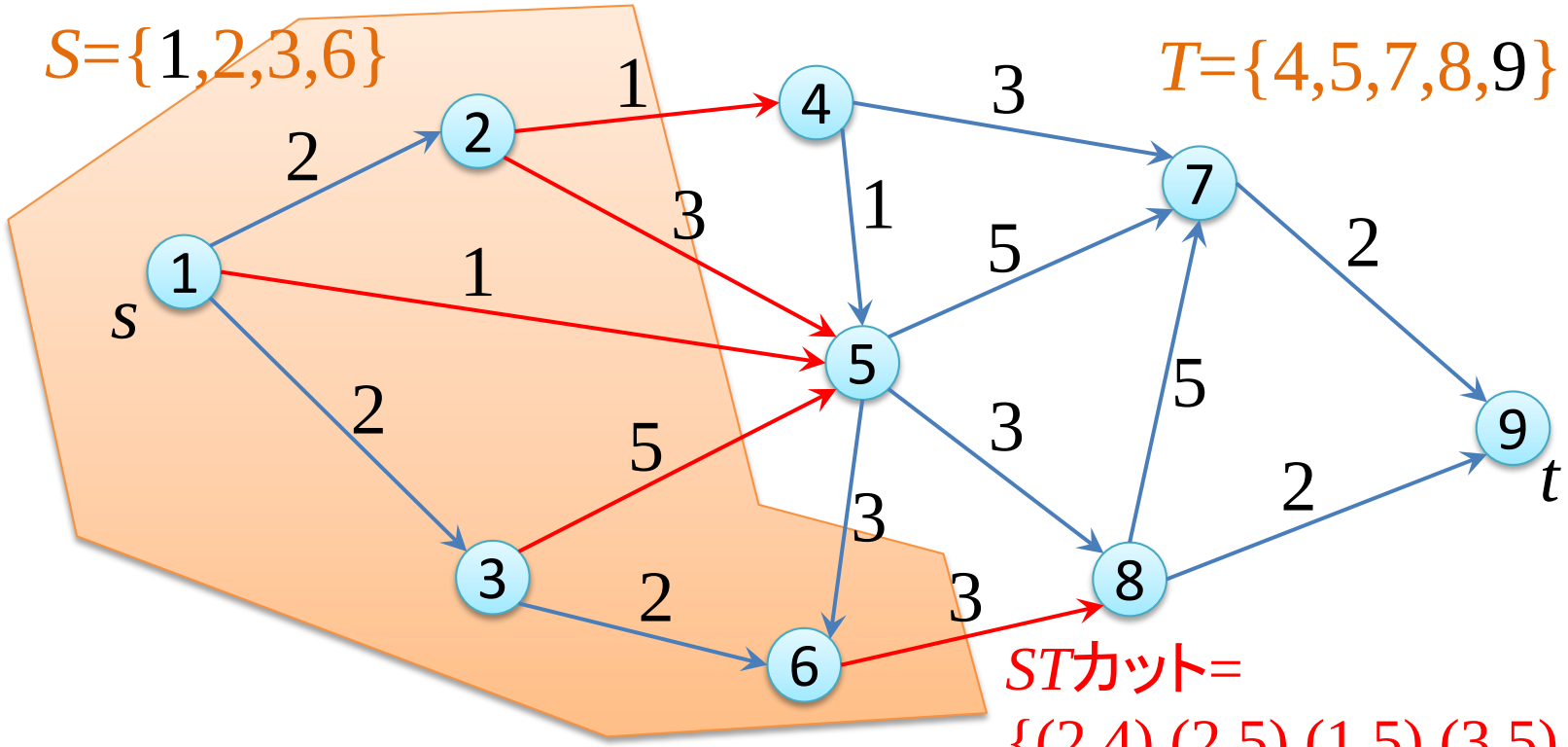
1つ目の制約式は，流量保存則を表す．即ち，start/goal以外の任意の点iについて「点iからの流出量の和」と「点iへの流入量の和」との差が0（流量保存）である（s[start]/t[goal]は流量保存制約から除外されることに注意）

目的関数は点s[start]の「流出量の和と流入量の和の差」を最大化することとなる

最小カット問題 minimum cut problem

- 例題

グラフG=(V,E)と枝$(i,j)\in E$上の容量（capacity）u_{ij}が与えられている
スタート点（点1）を含む点集合をS，ゴール点（点9）を含む点集合をTとし，VをSとTに分割
このとき，SからTへの枝の集合をSTカットとよぶ．容量が最小となるSTカットを求めよ



STカット=
{(2,4),(2,5),(1,5),(3,5),(6,8)}

STカット容量=13

【演習】 LPに定式化して Excel Solver で求解せよ
（LPファイルで定式化を書くより，Excel の方が定式化が楽）

最小カット問題の定式化と求解

- 例題：定式化例

0-1整数変数z_{ij}は，
枝$(i,j) \in E$がSTカットなら1，違うなら0
0-1整数変数y_iは，
点$i \in V$が集合Sに含まれるなら1，違うなら0

$$min. \quad \sum_{(i,j)\in E} u_{ij} z_{ij}$$

$$s.t. \quad y_i - y_j - z_{ij} \leq 0 \quad (\forall (i,j) \in E) \quad \text{...①}$$

$$y_s = 1, y_t = 0 \quad \text{...②}$$

$$z_{ij} \in \{0,1\} \quad (\forall (i,j) \in E)$$

$$y_i \in \{0,1\} \quad (\forall i \in V)$$

カット制約①は，$i \in S, j \in T$ のときy_i=1, y_j=0で，z_{ij}=1となることを要求する制約
それ以外，$i, j \in S$（y_i=y_j=1）や $i, j \in$ T（y_i=y_j=0）では，z_{ij}=1でも0でもよい．つまり，制約として意味がない．しかし目的関数が最小化なので，これらのときはz_{ij}=0となる
（z_{ij}, y_i の0-1制約は，LP緩和して$0 \leqq z_{ij} \leqq 1$, $0 \leqq y_i \leqq 1$で解いてよい）

【補足】

- 最大フロー・最小カット定理（max-flow min-cut theorem）
  - th) 最大フローが存在するとき，

    最大流量　＝　最小カット容量

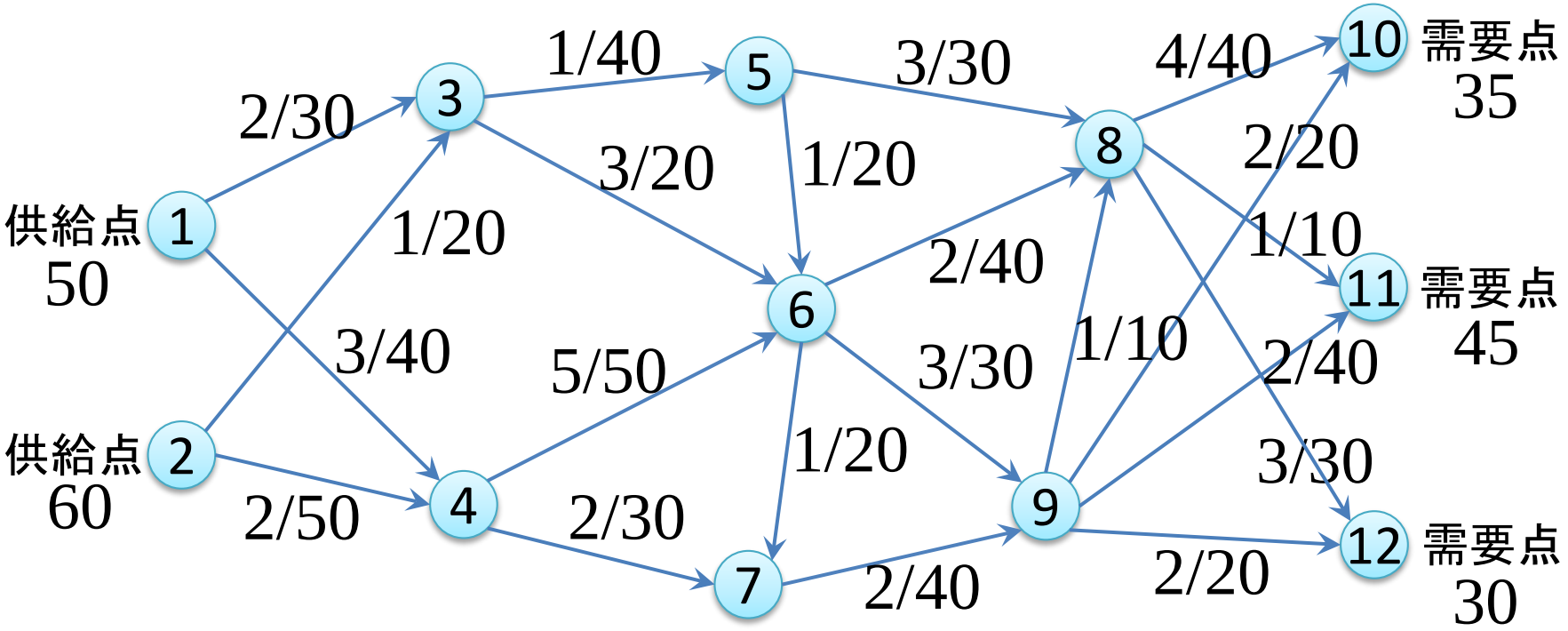
- 最大流問題を主問題(P)としたとき，双対問題(D)が最小カット問題となる
  - 最大フロー・最小カット定理は，双対定理の特殊ケース

- 双対定理（Duality Theorem）
  - th) LPの主問題(P)と双対問題(D)がどちらも実行可能なら，いずれも最適解を持ち最適値が一致する

最小費用流問題 minimum cost flow problem

- 例題

グラフG=(V,E)と枝$(i,j)\in E$上のコスト（cost）c_{ij}と容量（capacity）u_{ij}が与えられている
与えられた需要点の需要と供給点の供給量を満たすフローを考える
実行可能なフローのうちで費用最小となるものを求めよ



【演習】 LPに定式化して Excel Solver で求解せよ
（LPファイルで定式化を書くより，Excel の方が定式化が楽）

最小費用流問題の定式化と求解

- **例題**：定式化例

実数変数x_{ij}は，枝$(i,j) \in E$に流れる流量

$$min. \quad \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$s.t. \quad \sum_{j\in V} x_{ij} - \sum_{j\in V} x_{ji} = b_i \quad (\forall i \in V) \qquad \cdots①$$

$$0 \le x_{ij} \le u_{ij} \qquad (\forall (i,j) \in E)$$

点iからの流出量の和

点iへの流入量の和

流量保存制約①の右辺定数 b_i の値は以下の通り

- ✓ 供給点$i \in V$について　　b_i=その点の供給量
- ✓ 需要点$i \in V$について　　b_i=－その点の需要量
- ✓ それ以外の点$i \in V$について　　b_i=0（流量保存）

【補足】

- 最小費用流問題は，最短路問題と最大流問題を含む
  - 最小費用流問題の定式化において以下の設定をすれば良い
    - ✓スタート点$i=s$ について，　$b_s=1$
    - ✓ゴール点$i=t$ について，　$b_t=-1$
    - ✓それ以外の点i について，　$b_i=0$
    - ✓全ての枝(i,j)の容量　$u_{ij}=\infty$

    最短路問題

  - 最小費用流問題の定式化において以下の設定をすれば良い
    - ✓スタート点$i=s$ について，　$b_s=f$　※$f=\Sigma c_{sj}x_{sj}$
    - ✓ゴール点$i=t$ について，　$b_t=-f$　※この流量制約冗長（削除可）
    - ✓それ以外の点i について，　$b_i=0$
    - ✓スタート点からの枝(s,j)のコスト　$c_{sj}=1$
    - ✓それ以外の枝(i,j)のコスト　$c_{ij}=0$

    最大流問題

参考文献

1. 今野浩 「線形計画法」 日科技連（**1987**）
2. 藤田・今野・田邉 「最適化法」 岩波書店（**1994**）
3. 田村明久・村松正和 「最適化法」 共立出版（**2002**）
4. 坂和正敏 「線形計画法の基礎と応用」 朝倉書店（**2012**）
5. 小島・土谷・水野・矢部 「内点法」 朝倉書店（**2001**）
6. ***A. Schrijver: Theory of Linear and Integer Programming, John Wiley and Sons, 1986.***
7. ***L.A. Wolsey: Integer Programming, John Wiley and Sons, 1998.***
8. ***M. Conforti, G. Cornuejols and G.Zambelli: Integer Programming, Springer, 2014.***
9. 久保幹雄, **J.P.**ペドロソ, 村松正和, **A.**レイス：あたらしい数理最適化, 近代科学社,**2012.**
10. 久保幹雄, 小林和博, 斉藤努, 並木誠, 橋本英樹：**Python**言語によるビジネスアナリティクス, 近代科学社, **2016.**
11. 藤澤克樹, 後藤順哉, 安井雄一郎：**Excel**で学ぶ**OR**, オーム社, **2011.**