問題解決

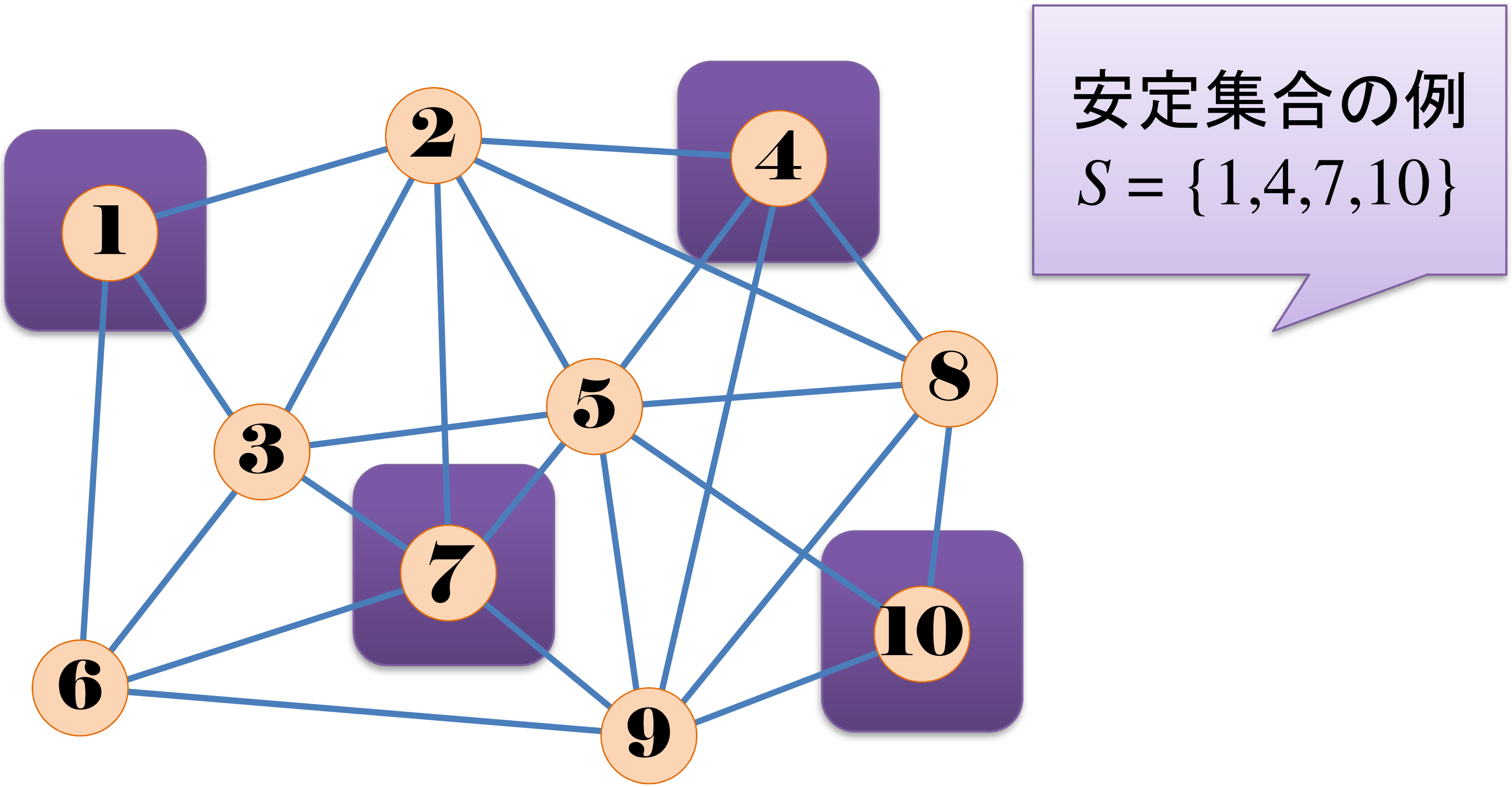
グラフ最適化と整数計画法
3. 最大安定集合問題

堀田 敬介

最大安定集合問題の最適化

- 最大安定集合問題 maximum stable set problem
  - 無向グラフ $G = (V, E)$，点集合 $V = \{1, 2, \ldots, n\}$，枝集合 E, $|V|=n$, $|E|=m$
  - 安定集合 stable set = V の部分集合 S で，任意の2点間に枝が無い集合

  - 目的 = 安定集合のうち要素数 $|S|$ が最大のものを求める



最大安定集合問題の最適化

- 最適化問題の定式化（変数設定）
  - 0-1変数 $x_v = \begin{cases} 1 \ldots \text{点 } v \text{ が安定集合 } S \text{ に含まれる}(v \in S) \\ 0 \ldots \text{点 } v \text{ が安定集合 } S \text{ に含まれない}(v \notin S) \end{cases}$

- 最適化問題の定式化（Σ表記・ベタ表記）

$$\begin{aligned} &\max. \textstyle\sum_{i=1}^{n} x_i \\ &\text{s. t. } x_i + x_j \leq 1 \ (\forall (i,j) \in E) \\ &\qquad\quad x_i \in \{0,1\} (\forall i \in V) \end{aligned}$$

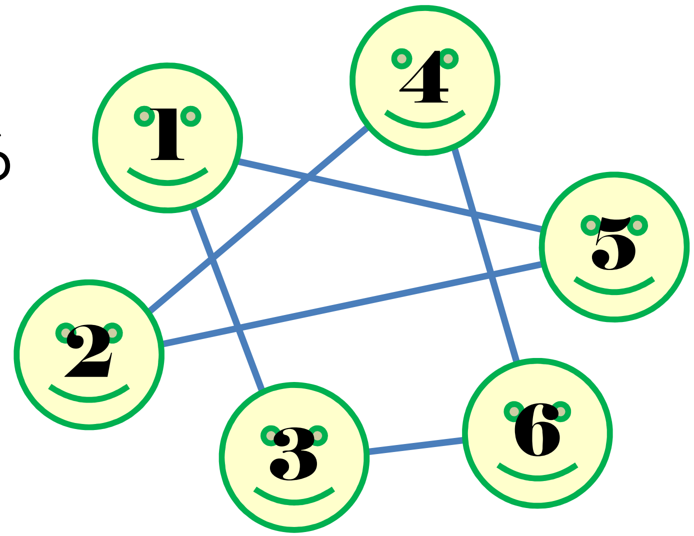
$$\begin{aligned} &\max. \ x_1 + x_2 + \ldots + x_n \\ &\text{s. t. } \ x_i + x_j \leqq 1 (\forall (i,j) \in E) \\ &\quad x_1, x_2, \ldots, x_n \in \{0,1\} \end{aligned}$$

制約式「 $x_i + x_j \leqq 1$ 」は,
「枝 (i, j) の両端点である点 i と点 j は,
どちらかしか安定集合 S に含まれない」という意味

最大安定集合問題の最適化

- <u>最大安定集合問題 maximum stable set problem</u>
  - 無向グラフ $G = (V, E)$，点集合 $V = \{1, 2, \ldots, n\}$，枝集合 E, $|V|=n$, $|E|=m$
  - 安定集合 stable set = V の部分集合 S で，任意の2点間に枝が無い集合
  - 目的 = 安定集合のうち要素数 $|S|$ が最大のものを求める

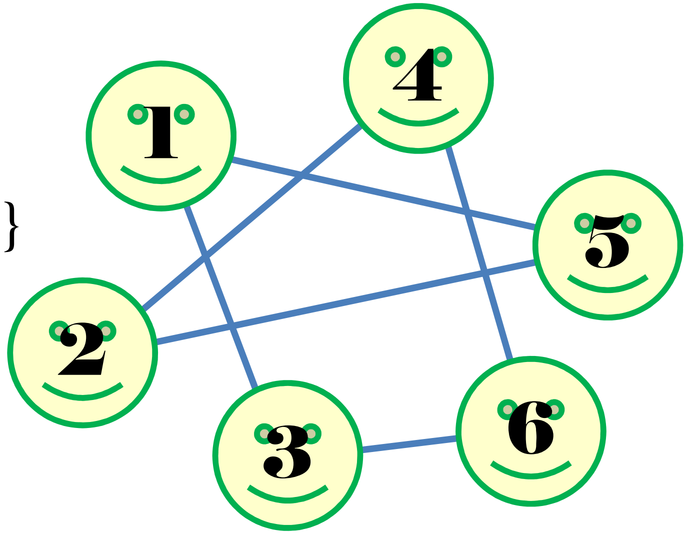
- 安定集合でモデル化出来る例（ex1）
  - 6人の学生から競技に参加する代表者を選出する
  - 仲の悪い学生を2人共はメンバーに選ばない
  - 最大人数の代表者を選出したい



  - 学生集合 = 点集合 $V = \{1,2,3,4,5,6\}$　※n=6
  - 不仲集合 = 枝集合 $E = \{(1,3),(1,5),(2,4),(2,5),(3,6),(4,6)\}$
  - 安定集合 = 選抜メンバー
  - 目的：最大安定集合 = 選抜メンバー構成員数最大

最大安定集合問題を解く

- グラフ $G = (V, E)$
  - 点集合 $V = \{1,2,3,4,5,6\}$　学生の集合
  - 枝集合 $E = \{(1,3),(1,5),(1,6),(2,4),(2,5),(3,6),(4,6)\}$
  - $|V|=6$, $|E|=7$



- 接続行列 incident matrix
  - 行に全点，列に全ての枝集合を対応させる
  - 各枝の端点に対応する2箇所の点に 1 と書く

学生	(1,3)	(1,5)	(2,4)	(2,5)	(3,6)	(4,6)
1	1	1				
2			1	1		
3	1				1	
4			1			1
5		1		1		
6					1	1

接続行列 A

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

最大安定集合問題の最適化

- 例1の定式化（Σ表記）

$$\max. \sum_{v=1}^{6} x_v$$
$$\text{s. t. } x_i + x_j \le 1 \ (\forall (i,j) \in E)$$
$$x_v \in \{0,1\} (v = 1, \dots, 6)$$

$$
\begin{array}{llllllll}
\max. & x_1+x_2+x_3+x_4+x_5+x_6 \\
\text{s. t.} & x_1 & & +x_3 & & & & \leqq 1 \\
& x_1 & & & & +x_5 & & \leqq 1 \\
& & x_2 & & +x_4 & & & \leqq 1 \\
& & x_2 & & & +x_5 & & \leqq 1 \\
& & & x_3 & & & +x_6 & \leqq 1 \\
& & & & x_4 & & +x_6 & \leqq 1
\end{array}
$$
$$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$$

- 例1の定式化（行列表記）

$$\max. \sum_{v=1}^{6} x_v$$
$$\text{s. t. } \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} \le \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$
$$x_v \in \{0,1\} (v = 1, \dots, 6)$$

接続行列 A の転置行列

$$\boldsymbol{A}^T = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$\max. \sum_{v=1}^{6} x_v$$
$$\text{s. t. } \sum_{(i,j)\in E} a^T_{(i,j),v} x_{(i,j)} \le 1 (v = 1, \dots, 6)$$
$$x_v \in \{0,1\} (v = 1, \dots, 6)$$

最大安定集合問題をCPLEXで解く

- 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：MaxStableSet）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 [MaxStableSet.mod] → [mss.mod]
  - ✓ データファイル [MaxStableSet.dat] → [mssex1.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し，解く

最大安定集合問題をCPLEXで解く

- モデルファイル（mss.mod）の中身の記述

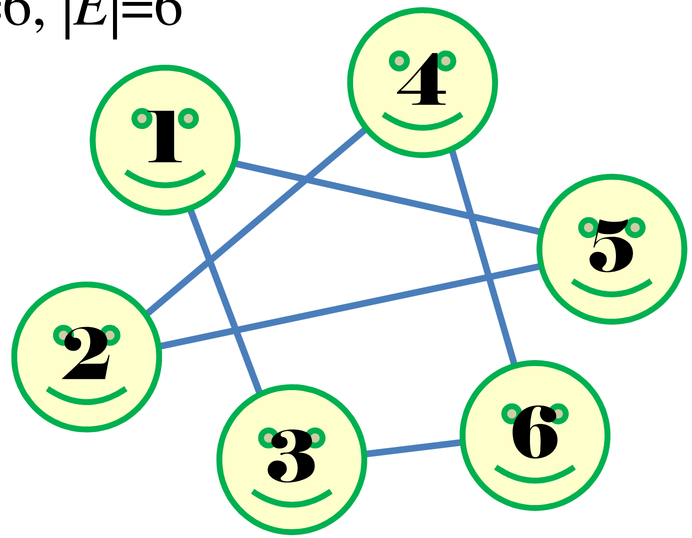
```
int e_max = ...;// 枝集合E の要素数|E|(接続行列A の転置行列AT の列数)
int v_max = ...;// 点集合V の要素数|V|(接続行列A の転置行列AT の行数)

range E = 1..e_max;// 枝集合E の範囲 [1..e_max] を指定
range V = 1..v_max;// 点集合V の範囲 [1..v_max] を指定

int AT[E,V] = ...;// 接続行列A の転置行列AT [size: E×V]

dvar int x[V] in 0..1;// 変数宣言:0-1変数ベクトル(size: V)

maximize
  sum(v in V) x[v];
subject to{
  forall(e in E) {
    sum(v in V) AT[e,v]*x[v] <= 1;
  };
};
```

最大安定集合問題をCPLEXで解く

- データファイル（mssex1.dat）の中身の記述

```
e_max = 6;
v_max = 6;

AT = [
[1 0 1 0 0 0]
[1 0 0 0 1 0]
[0 1 0 1 0 0]
[0 1 0 0 1 0]
[0 0 1 0 0 1]
[0 0 0 1 0 1]
];
```

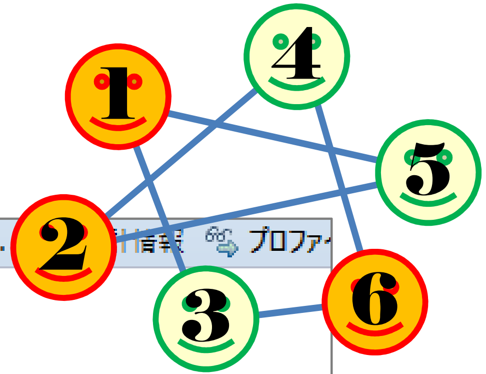
- 例1）グラフ $G = (V, E)$
  - 点集合 $V = \{1,2,3,4,5,6\}$
  - 枝集合 $E =$ $\{(1,3),(1,5),(2,4),(2,5),(3,6),(4,6)\}$
  - $|V|=6$, $|E|=6$



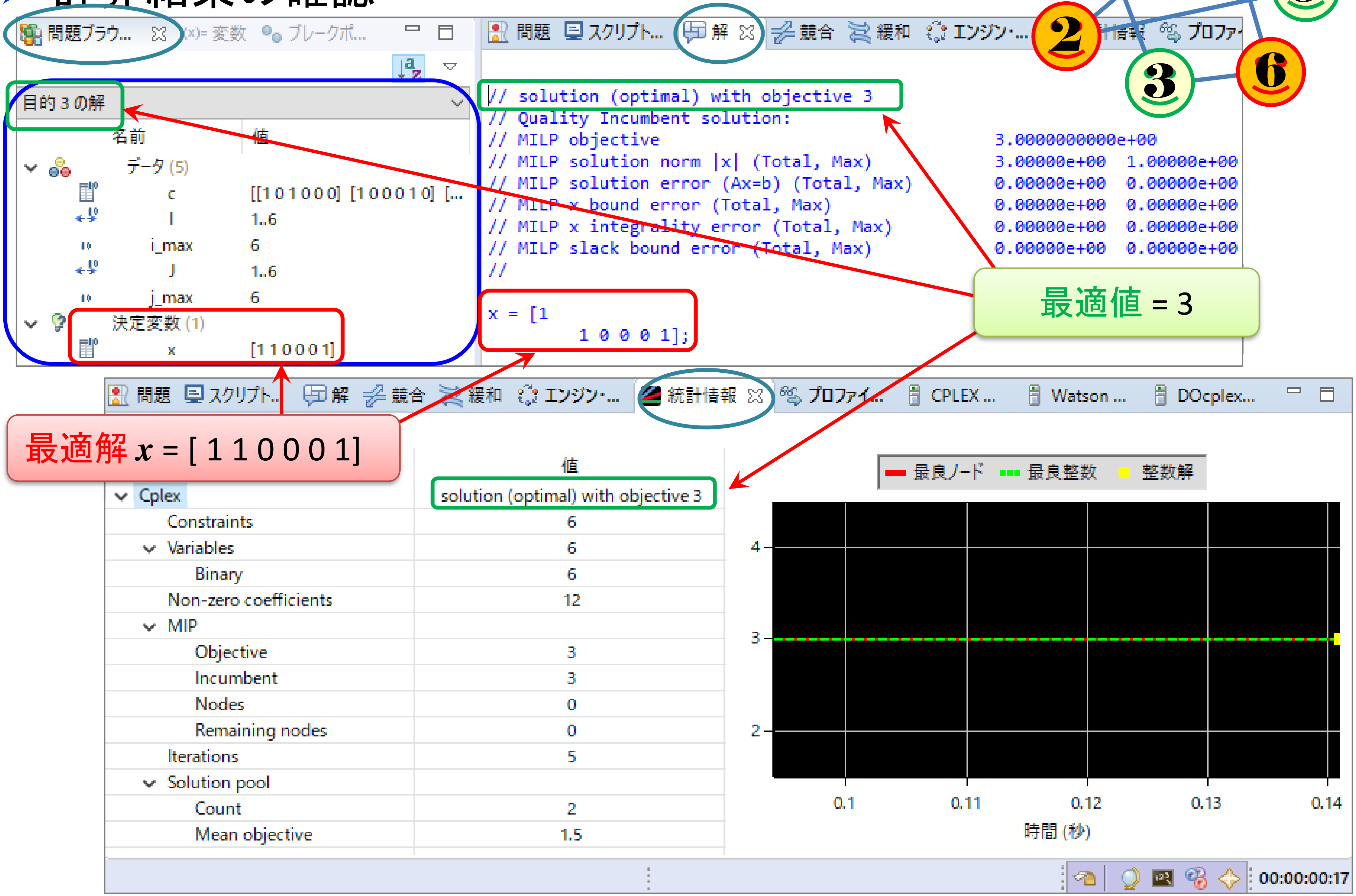
接続行列 A の
転置行列 A^T を
使うことに注意

接続行列 $\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$

最大安定集合問題をCPLEXで解く



- 計算結果の確認



最大安定集合問題をgurobiで解く(1)

- cplexの「モデルファイル(*.mod)」と「データファイル(*.dat)」を使って「lpファイル(*.lp)」を生成する
  - 例） モデルファイル [mss.mod], データファイル [mssex1.dat]
    → 生成する lpファイル [mssex1.lp]
  - [Win]+[R] キー で [ファイル名を指定して実行] d-boxを起動する
    - 枠内で cmd [Enter]
  - コマンドプロンプト command prompt のウィンドウ(黒い画面)が起動する

  - 以降, コマンドプロンプト内でコマンド(命令文)を打って順次命令を実行する

    (1) モデルファイルとデータファイルがあるフォルダに移動する

    ```
    cd [フォルダへのパス] [Enter]
    ```

    (2) 以下のコマンドを実行する

    ```
    oplrun -e mssex1.lp mss.mod mssex1.dat [Enter]
    ```

  - この結果, モデルファイル [mss.mod] とデータファイル [mssex1.dat] と同じフォルダ内に, lpファイル [mssex1.lp] が出来る(※確認すること)

最大安定集合問題をgurobiで解く(1)

- gurobi を起動して問題を解き，最適解を得る
  - コマンドプロンプトで，以下の命令文を打って gurobi を起動する

```
gurobi   [Enter]
```

  - 起動した gurobi 内で，順次，以下の命令文を打って問題を解いていく

    (1) 問題を記述してある lpファイル(mssex1.lp)を読み込み，model へセット

```
model = read("mssex1.lp")   [Enter]
```

    (2) 解く(最適化計算を開始する)　※読込に失敗しているとエラーとなる

```
model.optimize()   [Enter]
```

    (3) 最適解を表示する　※最適解が求まっていない場合はエラーとなる

```
model.printAttr(‘X’)   [Enter]
```

    (4) 最適値(目的関数値)を表示する　※同上

```
model.ObjVal   [Enter]
```

    (5) 最適解をファイル(*.sol)に出力する　※ファイル名は好きに

```
model.write("mssex1.sol")   [Enter]
```

最大安定集合問題をgurobiで解く(1)

- gurobi のその他，知っておくと便利な命令文
  - いずれも gurobi を起動して，gurobi内で実行する

(a) ヘルプを表示する

```
help()   [Enter]
```

(b) 全ての最適解(値が0の解)を表示する

```
for v in model.getVar() :    [Enter]
    print( v.VarName, ":", v.X)   [Enter]
```

- 最適解を表示する命令文「m.printAttr(‘X’)」は，値が0となる解は表示しない
- 2行目の print 文は，必ず字下げ(インデント)して書くこと(Pythonの文法)
- 字下げは[Tab]キーを使うと良い(※面倒でなければ，半角スペースでも可)
- model.getVar() でモデルから変数Var(variableの頭3文字) を get する命令
- get した各変数をインデックス v として，for文で繰り返す(2行目を繰り返す)
- v.VarName は，ゲットした各変数の「名称」を意味する予約語
- v.X は，ゲットした各変数の「値」を意味する予約語
- 以上より，各変数を1つずつ「名称 : 値」の形で画面に表示(print)する

最大安定集合問題をgurobiで解く(2)

➢ 問題(ex1)を python & gurobi で記述(mss.py)

①

```
# coding: Shift_JIS
from gurobipy import *

# ####### 例題設定 #######
def make_data_ex1():
	V = [1,2,3,4,5,6]
	E = [(1,3),(1,5),(2,4),(2,5),(3,6),(4,6)]
	return V,E
```

②

```
# ####### 定式化 #######
def mss(V,E):
	mod = Model("maximum stable set problem")

	# 変数設定
	x = {}
	for i in V:
		x[i] = mod.addVar(vtype="B", name="x(%s)" % i)
	mod.update()

	# 制約条件の設定
	for (i,j) in E:
		mod.addConstr(x[i] + x[j] <= 1)

	# 目的関数の設定
	mod.setObjective(quicksum(x[i] for i in V), GRB.MAXIMIZE)
	mod.update()
	mod.__data = x
	return mod
```

③

```
# ####### 実行 #######
if __name__=="__main__":
	V,E = make_data_ex1()			# データの生成
	mod = mss(V,E)				# モデルの生成
	mod.write("mssex1.lp")			# lpファイルを出力
	mod.optimize()				# 最適化実行
	print("¥n optimal value = ", mod.ObjVal)	# 最適値の表示
	mod.printAttr('X')			# 最適解の表示
	mod.write("mssex1.sol")			# 最適解をsolファイルに出力
```

1つのファイル「mss.py」に
①②③の順に記述して保存

最大安定集合問題をgurobiで解く(2)

- Pythonファイル(mss.py)をgurobi上で実行し，解く
  - [Win]+[R] キー で [ファイル名を指定して実行] d-boxを起動する
    - 枠内で `cmd [Enter]`
  - コマンドプロンプト command prompt のウィンドウ(黒い画面)が起動する
  - コマンドプロンプト内でコマンド(命令文)を打って順次命令を実行する

  (1) 実行ファイルがあるフォルダに移動する

```
cd [フォルダへのパス]   [Enter]
```

  (2) 以下の命令文を打って gurobi を起動する

```
gurobi   [Enter]
```

  - 起動した gurobi 内で，以下の命令文を打って問題を解く

```
gurobi> exec( open(“mss.py”).read() )  [Enter]
```

※python3系の場合

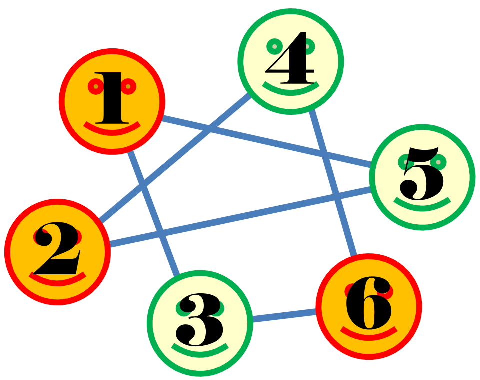
※python2系の場合の命令文は以下

```
gurobi> execfile(“mss.py”)  [Enter]
```

最大安定集合問題をgurobiで解く(2)

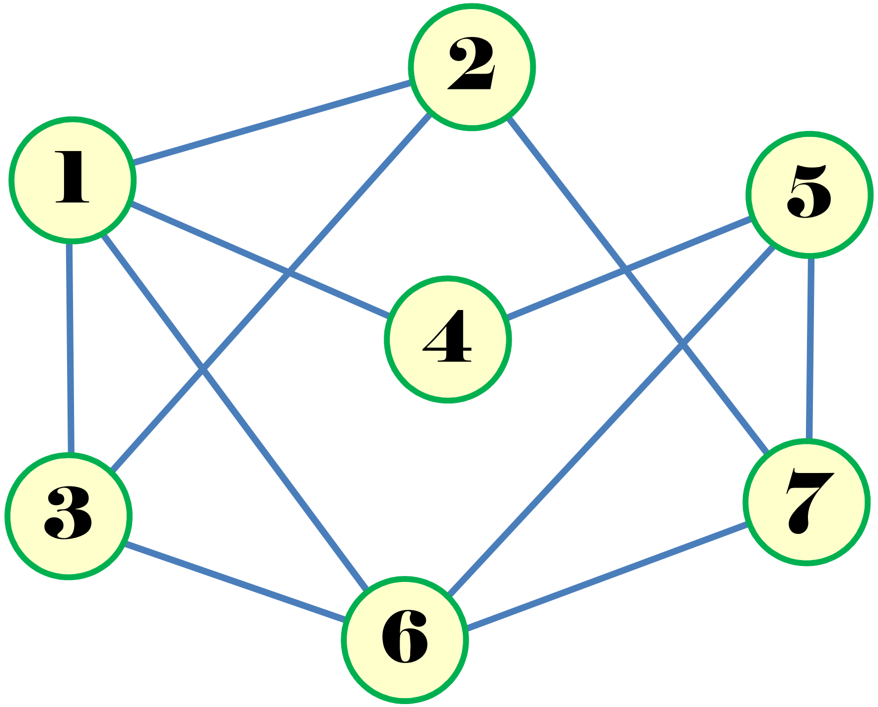
- 実行結果

```
gurobi> exec(open("mss.py").read())
Gurobi Optimizer version 9.5.2 build v9.5.2rc0 (win64)
Thread count: 10 physical cores, 20 logical processors, using up to 20 threads
Optimize a model with 6 rows, 6 columns and 12 nonzeros
Model fingerprint: 0x70420705
Variable types: 0 continuous, 6 integer (6 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 3.0000000
Presolve time: 0.00s
Presolved: 6 rows, 6 columns, 12 nonzeros
Variable types: 0 continuous, 6 integer (6 binary)

Root relaxation: cutoff, 2 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0     cutoff    0          3.00000    3.00000  0.00%     -    0s

Explored 1 nodes (2 simplex iterations) in 0.00 seconds (0.00 work units)
Thread count was 20 (of 20 available processors)

Solution count 1: 3

Optimal solution found (tolerance 1.00e-04)
Best objective 3.000000000000e+00, best bound 3.000000000000e+00, gap 0.0000%

 optimal value =  3.0

    Variable            X
-------------------------
        x(1)            1
        x(2)            1
        x(6)            1
gurobi>
```



【演習】最大安定集合問題をsolverで解く

- ex2) グラフ $G = (V, E)$
  - 点集合 $V = \{1,2,3,4,5,6,7\}$,
  - 枝集合 $E = \{(1,2), (1,3), (1,4), (1,6), (2,3), (2,3), (2,7), (3,6), (4,5), (5,6), (5,7), (6,7)\}$



- 問
  1. $|V| = ?$ $|E| = ?$
  2. 接続行列A をつくれ
  3. 例1と同様に0-1変数 $x_{(i,j)}$ を設定し，定式化せよ
  4. 整数計画ソルバー（cplex）を用いて，最大安定集合を求めよ
  5. oplrun を使って，mod file / dat file から lp file を作れ
  6. 整数計画ソルバー（gurobi）で5のlp file を解き，最大安定集合を求めよ
  7. 整数計画ソルバー（gurobi）とpython で解き，最大安定集合を求めよ
  8. 結果を networkx でグラフ描画せよ

【演習】最大安定集合問題をsolverで解く

- ex3）最大安定集合問題の最適化
  - ランダムグラフ $G = (V, E)$ で問題を作る（python/networkx等を利用）
    - 点集合の要素数 $|V|$ を適当に設定（$n = 5$〜20 程度）
    - 枝集合 E の密度を適当に設定（0.0〜1.0）

- 問
  1. $|V| = ?$　$|E| = ?$
  2. 接続行列A をつくれ
  3. 例1と同様に0-1変数 $x_{(i,j)}$ を設定し，定式化せよ
  4. 整数計画ソルバー（cplex）を用いて，最大安定集合を求めよ
  5. oplrun を使って，mod file / dat file から lp file を作れ
  6. 整数計画ソルバー（gurobi）で5のlp file を解き，最大安定集合を求めよ
  7. 整数計画ソルバー（gurobi）とpython で解き，最大安定集合を求めよ
  8. 結果を networkx でグラフ描画せよ