問題解決

グラフ最適化と整数計画法
7. 辺彩色問題

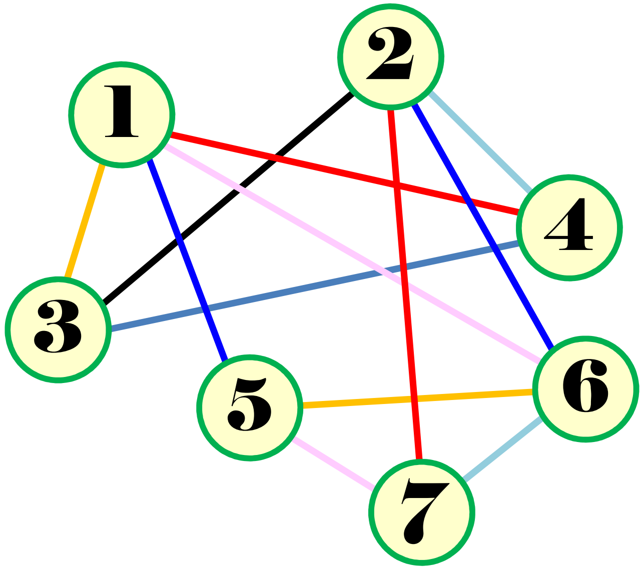
堀田 敬介

辺彩色問題の最適化

- **辺彩色問題 edge coloring problem**
  - 無向グラフ $G = (V, E)$
    - 点集合 $V = \{1, 2, \ldots, n\}$，枝集合 $E=\{1,2,\ldots,m\}$，$|V|=n$, $|E|=m$
  - 辺彩色 = 隣接枝に異なる色を割り当てるとき何色必要か？

- 枝数 $|E|=m$ より，自明解 m色（全枝に異なる色を割当）
- 「点の最大次数＝最小必要色数」だとわかる（最大次数=d_{max} としよう）
- 目的 = 色数k が最小の割当を求める

※ $k \in [d_{max}, m]$ （使用色数 k は d_{max} 以上 m 以下）



$d_{max}=4$, $|E|=m=12$より，この例では最小4色必要で，最大でも12色だと（解く前に）わかる

辺彩色問題の最適化

- 最適化問題の定式化（変数設定・係数表記）
  - 0-1変数 $x_{ek} = \begin{cases} 1 & \ldots \text{枝}e\text{に色}k\text{を割り当て} \\ 0 & \ldots \text{枝}e\text{に色}k\text{を割り当てない} \end{cases}$
  - 0-1変数 $y_k = \begin{cases} 1 & \ldots \text{色}k\text{を使う} \\ 0 & \ldots \text{色}k\text{を使わない} \end{cases}$
  - 接続行列 $\boldsymbol{A} = [a_{ve}]$
- 最適化問題の定式化（Σ表記・ベタ表記）

$$\min. \sum_{k=1}^{m} y_k$$

$$\text{s. t.} \sum_{k=1}^{m} x_{ek} = 1 (\forall e \in E)$$

$$\sum_{e \in E} a_{ve} x_{ek} \leq y_k (\forall v \in V, \forall k)$$

$$y_k \geq y_{k+1} \ (k = 1..m - 1)$$

$$x_{ek}, y_k \in \{0,1\} (\forall e \in E, \forall k)$$

$$\min. \ y_1 + y_2 + \ldots + y_m$$

$$\text{s. t.} \ x_{11} + x_{12} + \ldots + x_{1m} = 1$$

$$\ldots$$

$$x_{m1} + x_{m2} + \ldots + x_{mm} = 1$$

$$a_{11}x_{1k} + a_{12}x_{2k} + \cdots + a_{1m}x_{mk} \leq y_k (\forall k)$$

$$\ldots$$

$$a_{n1}x_{1k} + a_{n2}x_{2k} + \cdots + a_{nm}x_{mk} \leq y_k (\forall k)$$

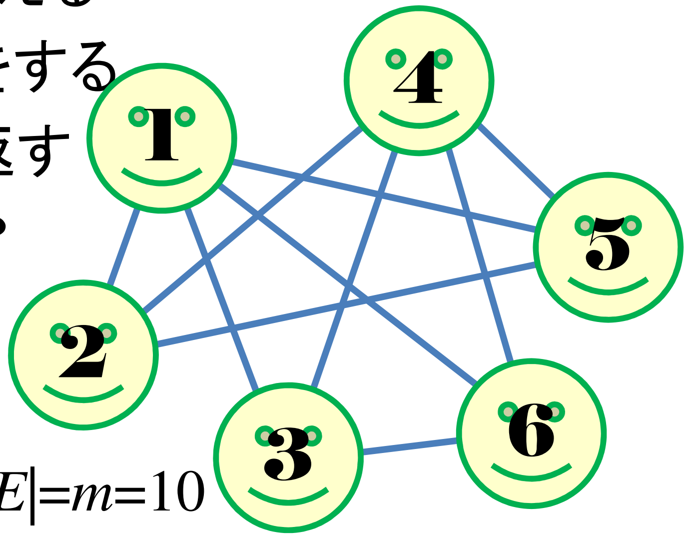
$$y_1 \geqq y_2, y_2 \geqq y_3, \ldots, y_{n-1} \geqq y_m$$

$$x_{e1}, \ldots, x_{em} \ y_1, \ldots, y_m \in \{0,1\}$$

辺彩色問題の最適化

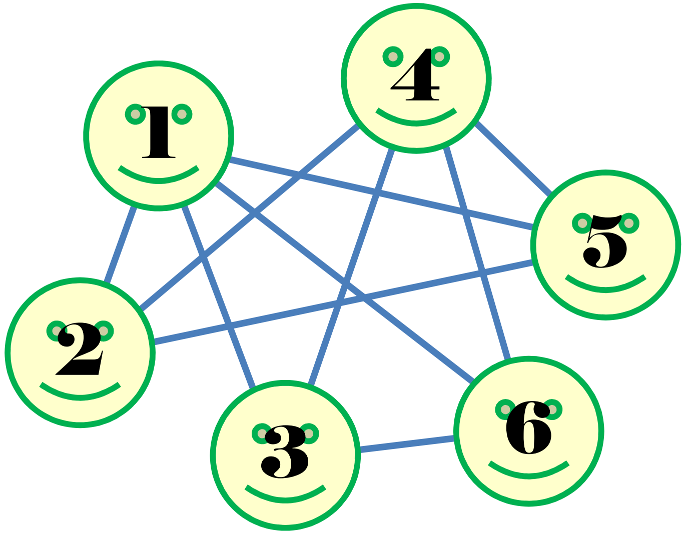
- 辺彩色問題 vertex coloring problem
  - 無向グラフ $G = (V, E)$
    - 点集合 $V = \{1, 2, \ldots, n\}$, 枝集合 $E=\{1,2,\ldots,m\}$, $|V|=n$, $|E|=m$
  - 辺彩色 = 隣接枝に異なる色を割り当てるとき何色必要か？

- 辺彩色でモデル化出来る例（ex1）
  - ペアで出場する競技に参加する人々の集団を考える
  - 全員がペアになれる全ての相手と1回ずつ練習をする
  - 1ペアの練習時間は共通（1セット）でそれを繰り返す
  - 練習時間最小のために必要繰返し数は何回か？



  - 人集合 = 点集合 $V= \{1, 2, \ldots, 6\}$　※$|V|=n=6$
  - ペア可能集合 = 枝集合 $E = \{1, 2, \ldots, 10\}$　※$|E|=m=10$
  - 割り当てた同色 = 1セットで同時に練習可能なペア群
  - 目的：色数最小 = セット数最小

辺彩色問題の最適化



- グラフ $G = (V, E)$
  - 点集合 $V = \{1,2,3,4,5,6\}$
  - 枝集合 $E = \{1,2,3,4,5,6,7,8,9,10\}$
  - $|V|=6,\ |E|=10$

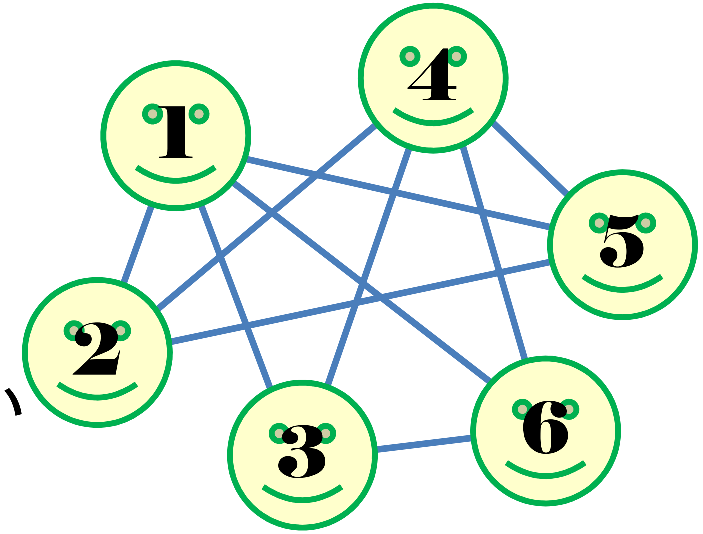
- 接続行列 incident matrix
  - 行に全点，列に全ての枝集合を対応させる
  - 各枝の端点に対応する2箇所の点に 1 と書く

ペア 人＼	1 (1,2)	2 (1,3)	3 (1,5)	4 (1,6)	5 (2,4)	6 (2,5)	7 (3,4)	8 (3,6)	9 (4,5)	10 (4,6)
1	*1*	*1*	*1*	*1*						
2	*1*				*1*	*1*				
3		*1*					*1*	*1*		
4					*1*		*1*		*1*	*1*
5			*1*			*1*			*1*	
6				*1*				*1*		*1*

接続行列 A

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

辺彩色問題の最適化



- 例1の定式化(変数設定)
  - 0-1変数 $x_{ek} = \begin{cases} 1 & \ldots \text{枝}e\text{に色}k\text{を割り当て} \\ 0 & \ldots \text{枝}e\text{に色}k\text{を割り当てない} \end{cases}$
  - 0-1変数 $y_k = \begin{cases} 1 & \ldots \text{色}k\text{を使う} \\ 0 & \ldots \text{色}k\text{を使わない} \end{cases}$
- 例1の定式化(ベタ表記)

min. $y_1+y_2+y_3+y_4+y_5+y_6+y_7+y_8+y_9+y_{10}$

s. t. $x_{11}+x_{12}+x_{13}+x_{14}+x_{15}+x_{16}+x_{17}+x_{18}+x_{19}+x_{1,10}=1$

$x_{21}+x_{22}+x_{23}+x_{24}+x_{25}+x_{26}+x_{27}+x_{28}+x_{29}+x_{2,10}=1$

…

$x_{10,1}+x_{10,2}+x_{10,3}+x_{10,4}+x_{10,5}+x_{10,6}+x_{10,7}+x_{10,8}+x_{10,9}+x_{10,10}=1$

$x_{11}+x_{21}+x_{31}+x_{41} \leqq y_1,\ x_{12}+x_{22}+x_{32}+x_{42} \leqq y_2,\ \ldots,\ x_{1,10}+x_{2,10}+x_{3,10}+x_{4,10} \leqq y_{10}$

$x_{11}+x_{51}+x_{61} \leqq y_1,\ x_{12}+x_{52}+x_{62} \leqq y_2,\ \ldots,\ x_{1,10}+x_{5,10}+x_{6,10} \leqq y_{10}$

…

$x_{31}+x_{81}+x_{10,1} \leqq y_1,\ x_{32}+x_{82}+x_{10,2} \leqq y_2,\ \ldots,\ x_{3,10}+x_{8,10}+x_{10,10} \leqq y_{10}$

$y_1 \geqq y_2,\ y_2 \geqq y_3, \ldots,\ y_9 \geqq y_{10}$

$x_{11}, x_{12}, \ldots, x_{1,10},$
$x_{21}, x_{22}, \ldots, x_{2,10},$
…
$x_{10,1}, x_{10,2}, \ldots, x_{10,10},$
$y_1, y_2, \ldots, y_{10} \in \{0,1\}$

辺彩色問題の最適化

- 例1の定式化(Σ表記)

$$\min. \sum_{k=1}^{10} y_k$$

$$\text{s. t. } \sum_{k=1}^{10} x_{ek} = 1 \; (e = 1, \dots, 10)$$

$$\sum_{e=1}^{10} a_{ve} x_{ek} \leq y_k$$

$$(v = 1, \dots, 10; k = 1, \dots, 10)$$

$$y_k \geq y_{k+1} \; (k = 1, \dots, 9)$$

$$x_{ek}, y_k \in \{0,1\} (e = 1, \dots, 10; k = 1, \dots, 10)$$

$$\boldsymbol{x}_k = \begin{pmatrix} x_{1k} \\ x_{2k} \\ x_{3k} \\ x_{4k} \\ \dots \\ x_{10k} \end{pmatrix}, \boldsymbol{x}_e = \begin{pmatrix} x_{e1} \\ x_{e2} \\ x_{e3} \\ x_{e4} \\ \dots \\ x_{e10} \end{pmatrix}$$

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \dots \\ y_{10} \end{pmatrix}, \boldsymbol{e} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{pmatrix}$$

- 例1の定式化(行列表記)

$$\min. \; \boldsymbol{e}^T \boldsymbol{y}$$

$$\text{s. t. } \; \boldsymbol{e}^T \boldsymbol{x}_e = 1 (\forall e \in E)$$

$$A \boldsymbol{x}_k \leq y_k \; (\forall k)$$

$$y_k \geq y_{k+1} \; (k = 1, \dots, 9)$$

$$x_{ek}, y_k \in \{0,1\} (\forall e \in E; \forall k)$$

接続行列 A

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

辺彩色問題を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：EdgeColoring）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 [EdgeColoring.mod] → [ec.mod]
  - ✓ データファイル [EdgeColoring.dat] → [ecex1.dat]
- 空のExcelファイル[ec.xlsx]を作り，プロジェクト内にドラッグ&ドロップする（※これでプロジェクトの保存フォルダにコピーされる）
- 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

辺彩色問題をCPLEXで解く

- モデルファイル(ec.mod)の中身の記述

```
int e_max = ...;// 枝集合E の要素数|E|
int v_max = ...;// 点集合V の要素数|V|

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
range K = 1..e_max;// 色集合K の範囲 [1..e_max] を指定(最大値はe_max)

int A[V,E] = ...;// 接続行列A [size: |V|×|E|]
int AT[e in E, v in V] = A[v,e];     // 接続行列Aの転置行列AT(出力用)

dvar int x[E,K] in 0..1;// 変数宣言:0-1変数(size:|E|×|K|)
dvar int y[K] in 0..1;  // 変数宣言:0-1変数(size:|K|)

minimize
  sum(k in K) y[k];
subject to{
  forall(e in E) {          // 各枝eへ割り当てる色は丁度1色
    sum(k in K) x[e,k] == 1;
  };
  forall(v in V) {          // 隣接2枝(同じ点に接続する2つの枝)へは異なる色を割り当てる
    forall(k in K) {
      sum(e in E) A[v,e]*x[e,k] <= y[k];
    };
  };
  forall(k in 1..e_max-1) {          // 色変数y[k]は添え字の小さい方から使う
    y[k] >= y[k+1];
  };
};
```

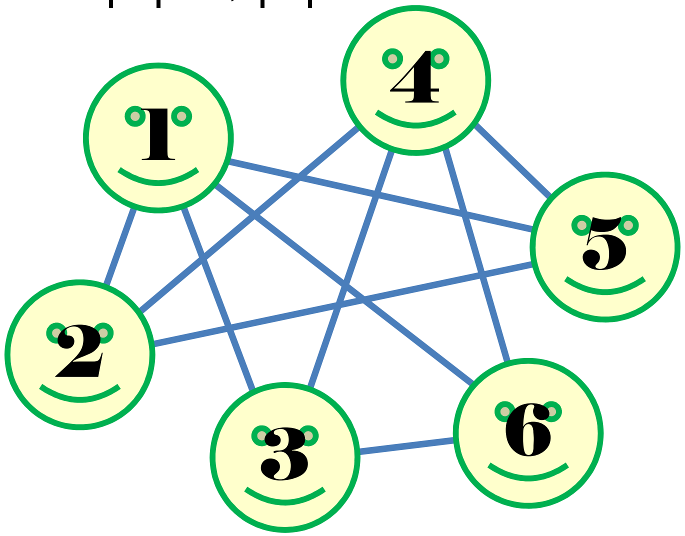
辺彩色問題をCPLEXで解く

- データファイル（ecex1.dat）の中身の記述

```
 e_max = 10;   // 枝集合E の要素数|E|
 v_max = 6;    // 点集合V の要素数|V|

 A = [         // 接続行列A
 [1 1 1 1 0 0 0 0 0 0]
 [1 0 0 0 1 1 0 0 0 0]
 [0 1 0 0 0 0 1 1 0 0]
 [0 0 0 0 1 0 1 0 1 1]
 [0 0 1 0 0 1 0 0 1 0]
 [0 0 0 1 0 0 0 1 0 1]
];

SheetConnection sheet("ec.xlsx");
AT to SheetWrite(sheet, "Sheet1!B3:G12");
y to SheetWrite(sheet, "Sheet1!J2:S2");
x to SheetWrite(sheet, "Sheet1!J3:S12");
```

計算結果を
Excelファイル[ec.xlsx]に出力

- 例1）グラフ $G=(V,E)$
  - 点集合 $V=\{1,2,\ldots,6\}$
  - 枝集合 $E=\{1,2,\ldots,10\}$
  - $|V|=6,\ |E|=10$



接続行列 A

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

辺彩色問題をCPLEXで解く

- 結果(「解」タブ)

```
// solution (optimal) with objective 4
// Quality Incumbent solution:
// MILP objective                                4.0000000000e+00
// MILP solution norm |x| (Total, Max)           1.40000e+01  1.00000e+00
// MILP solution error (Ax=b) (Total, Max)       0.00000e+00  0.00000e+00
// MILP x bound error (Total, Max)               0.00000e+00  0.00000e+00
// MILP x integrality error (Total, Max)         0.00000e+00  0.00000e+00
// MILP slack bound error (Total, Max)           0.00000e+00  0.00000e+00
//

y =  [1 1 1 1 0 0 0 0 0 0];
x = [[0 0 1 0 0 0 0 0 0 0]
     [1 0 0 0 0 0 0 0 0 0]
     [0 1 0 0 0 0 0 0 0 0]
     [0 0 0 1 0 0 0 0 0 0]
     [0 1 0 0 0 0 0 0 0 0]
     [1 0 0 0 0 0 0 0 0 0]
     [0 0 1 0 0 0 0 0 0 0]
     [0 1 0 0 0 0 0 0 0 0]
     [0 0 0 1 0 0 0 0 0 0]
     [1 0 0 0 0 0 0 0 0 0]];
```

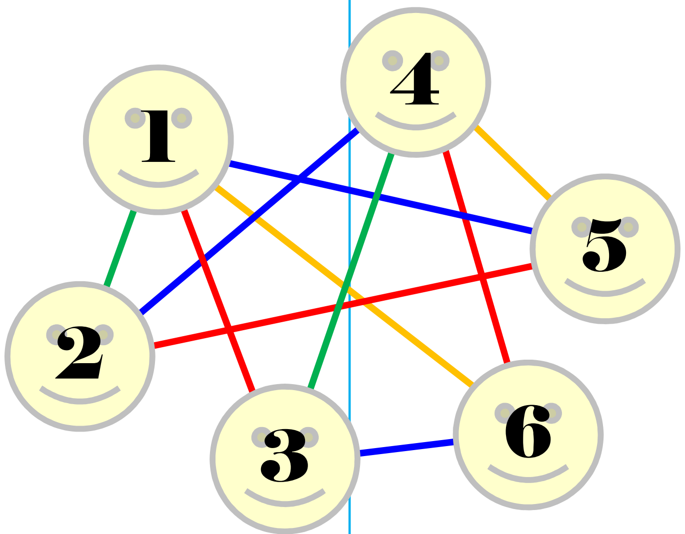
最適値 = 4

4色使う

最適解

色1を枝番2,6,10へ割当
色2を枝番3,5,8へ割当
色3を枝番1,7へ割当
色4を枝番4,9へ割当

実際に枝に
最適解の色を
塗ってみた
↓



- 結果の解釈

枝番号と枝集合の対応表

1	2	3	4	5	6	7	8	9	10
(1,2)	(1,3)	(1,5)	(1,6)	(2,4)	(2,5)	(3,4)	(3,6)	(4,5)	(4,6)

色1 = 枝2(1,3), 6(2,5), 10(4,6)
色2 = 枝3(1,5), 5(2,4), 8(3,6)
色3 = 枝1(1,2), 7(3,4)
色4 = 枝4(1,6), 9(4,5)

1セット目練習ペア群 = (1,3), (2,5), (4,6)
2セット目練習ペア群 = (1,5), (2,4), (3,6)
3セット目練習ペア群 = (1,2), (3,4)　※5,6は休憩
4セット目練習ペア群 = (1,6), (4,5)　※2,3は休憩

辺彩色問題をCPLEXで解く

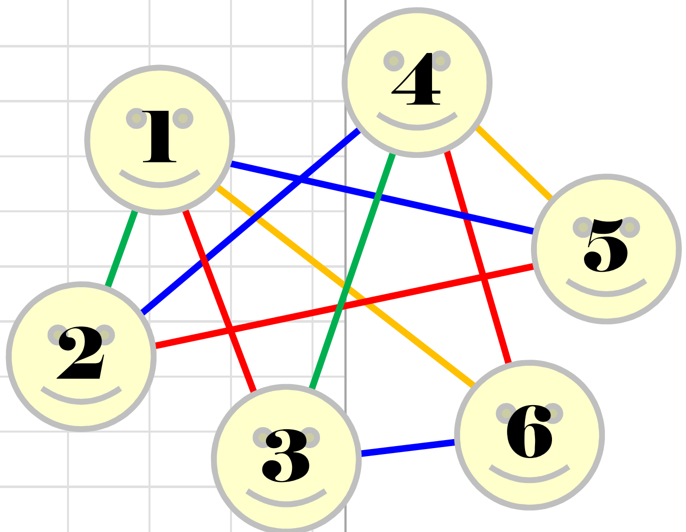
➢ 結果（Excelファイル[ec.xlsx]）

[4色]使う

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1									k(色)	1	2	3	4	5	6	7	8	9	10
2	v	1	2	3	4	5	6		y	1	1	1	1	0	0	0	0	0	0
3	AT=	1	1	0	0	0	0	(1, 2)	x	0	0	1	0	0	0	0	0	0	0
4		1	0	1	0	0	0	(1, 3)		1	0	0	0	0	0	0	0	0	0
5		1	0	0	0	1	0	(1, 5)		0	1	0	0	0	0	0	0	0	0
6		1	0	0	0	0	1	(1, 6)		0	0	0	1	0	0	0	0	0	0
7		0	1	0	1	0	0	(2, 4)		0	1	0	0	0	0	0	0	0	0
8		0	1	0	0	1	0	(2, 5)		1	0	0	0	0	0	0	0	0	0
9		0	0	1	1	0	0	(3, 4)		0	0	1	0	0	0	0	0	0	0
10		0	0	1	0	0	1	(3, 6)		0	1	0	0	0	0	0	0	0	0
11		0	0	0	1	1	0	(4, 5)		0	0	0	1	0	0	0	0	0	0
12		0	0	0	1	0	1	(4, 6)		1	0	0	0	0	0	0	0	0	0
13																			
14		1	12	12	12	12	12												
15		1	1	13	13	13	13												
16		1	1	1	1	15	15												
17		1	1	1	1	1	16												
18			2	2	24	24	24												
19			2	2	2	25	25												
20				3	34	34	34												
21				3	3	3	36												
22					4	45	45												
23					4	4	46												

枝と対応色

実際に枝に
最適解の色を
塗ってみた
↓



辺彩色問題を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 [ec.mod], データファイル [ecex1.dat]
    → 生成する lpファイル [ecex1.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 ecex1.lp ec.mod ecex1.dat   [Enter]
```

  - この結果，モデルファイル [ec.mod] とデータファイル [ecex1.dat] と同じフォルダ内に，lpファイル [ecex1.lp] が出来る(※確認すること)

辺彩色問題を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(ecex1.lp)を読み込み，model へセット

```
model = read("ecex1.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("ecex1.sol")   [Enter]
```

辺彩色問題を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

辺彩色問題をgurobiで解く(2)

- 問題(ex1)を python & gurobi で記述(ec.py)

①

```
# coding: Shift_JIS
from gurobipy import *

# ####### 例題設定 #######
def make_data_ex1():
	V = [1,2,3,4,5,6]
	E = [(1,2),(1,3),(1,5),(1,6),(2,4),(2,5),(3,4),
(3,6),(4,5),(4,6)]
	K = [1,2,3,4,5,6,7,8,9,10]
	return V,E,K
```

1つのファイル「ec.py」に
①②③の順に記述して保存

③

```
# ####### 実行 #######
if __name__=="__main__":
	V,E,K = make_data_ex1()		# データの
	mod = ec(V,E,K)		# モデルの
	mod.write(“ecex1.lp")		# lpファイ
	mod.optimize()		# 最適化実
	print("¥n optimal value = ", mod.ObjVal)
	mod.printAttr('X')		# 最適解の
	mod.write(“ecex1.sol")		# 最適解を
```

②

```
# ####### 定式化 #######
def ec(V,E,K):
	mod = Model(“edge coloring problem")

	# 変数設定
	x,y = {},{}
	for k in K:
	   y[k] = mod.addVar(vtype="B", name="y(%s)" % k)
	   for (I,j) in E:
	      x[i,j,k] = mod.addVar(vtype="B", name="x[(%s,%s),%s]" % (i,j,k))
	mod.update()

	# 制約条件の設定
	for (i,j) in E:
	   mod.addConstr(quicksum(x[i,j,k] for k in K) == 1)
	for v in V:
	   for k in K:
	      mod.addConstr(quicksum(x[i,j,k] for (i,j) in E if i==v or j==v) <= y[k])
	for k in K[0:len(K)-1]:
	   mod.addConstr(y[k] >= y[k+1])

	# 目的関数の設定
	mod.setObjective(quicksum(y[k] for k in K), GRB.MINIMIZE)
	mod.update()
	mod.__data = x,y
	return mod
```

辺彩色問題をgurobiで解く(2)

- Pythonファイル(ec.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(“ec.py”).read() )  [Enter]
```

※python3系の場合

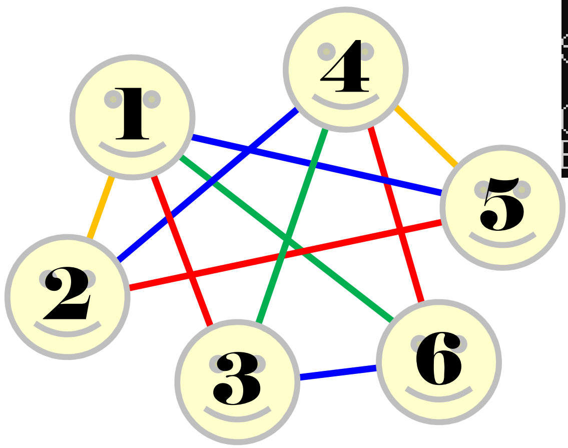
※python2系の場合の命令文は以下

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

辺彩色問題をgurobiで解く(2)

- 実行結果

```
optimal value =   4.0

   Variable            X
-------------------------
       y(1)            1
 x[(1,6),1]            1
 x[(3,4),1]            1
       y(2)            1
 x[(1,3),2]            1
 x[(2,5),2]            1
 x[(4,6),2]            1
       y(3)            1
 x[(1,5),3]            1
 x[(2,4),3]            1
 x[(3,6),3]            1
       y(4)            1
 x[(1,2),4]            1
 x[(4,5),4]            1
gurobi>
```

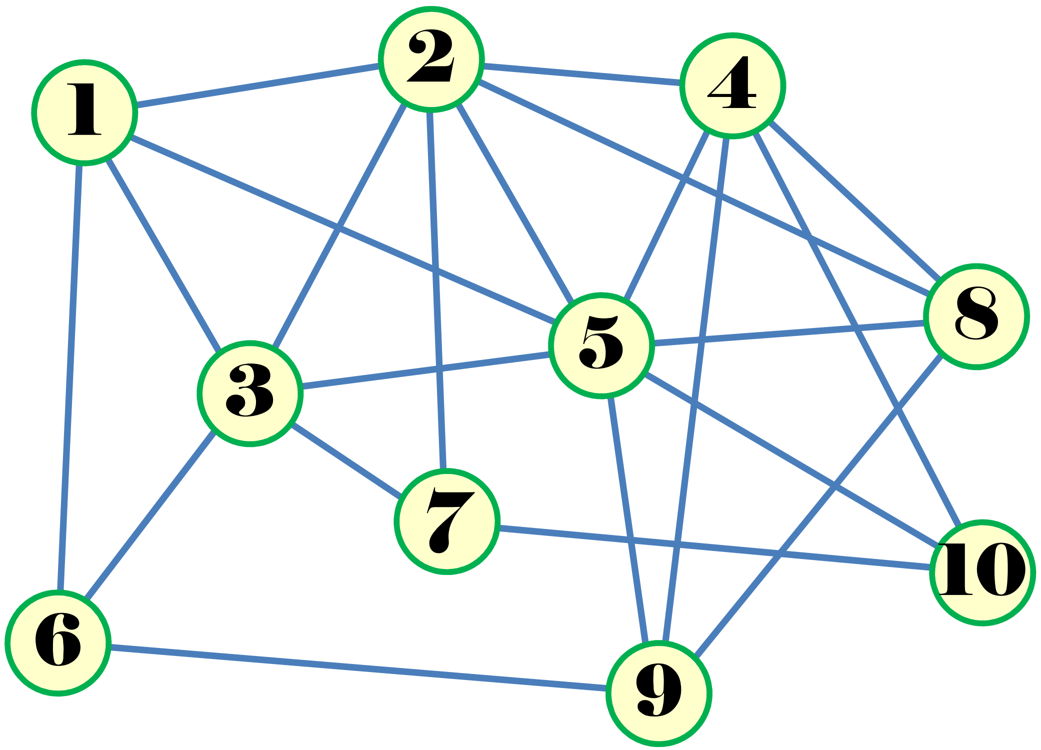



```
gurobi> exec(open("ec.py").read())
Gurobi Optimizer version 9.5.2 build v9.5.2rc0 (win64)
Thread count: 10 physical cores, 20 logical processors, using up to 20 threads
Optimize a model with 79 rows, 110 columns and 378 nonzeros
Model fingerprint: 0xeb1990f6
Variable types: 0 continuous, 110 integer (110 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 10.0000000
Presolve removed 2 rows and 2 columns
Presolve time: 0.00s
Presolved: 77 rows, 108 columns, 362 nonzeros
Variable types: 0 continuous, 108 integer (108 binary)

Root relaxation: objective 4.000000e+00, 83 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

H    0     0                       4.0000000    2.00000  50.0%     -    0s
     0     0          -    0         4.00000    4.00000  0.00%     -    0s

Explored 1 nodes (151 simplex iterations) in 0.01 seconds (0.00 work units)
Thread count was 20 (of 20 available processors)

Solution count 2: 4 10

Optimal solution found (tolerance 1.00e-04)
Best objective 4.000000000000e+00, best bound 4.000000000000e+00, gap 0.0000%
```

【演習】辺彩色問題を解く

- ex2) グラフ $G = (V, E)$
  - 点集合 $V = \{1,2,\ldots,10\}$,
  - 枝集合 $E = \{1,2,\ldots,22\}$
    $=\{(1,2),(1,3),(1,5),(1,6),(2,3),(2,4),(2,5),(2,7),(2,8),(3,5),(3,6),$
    $(3,7),(4,5),(4,8),(4,9),(4,10),(5,8),(5,9),(5,10),(6,9),(7,10),(8,9)\}$

（$|V|=n=10$）
（$|E|=m=22$）

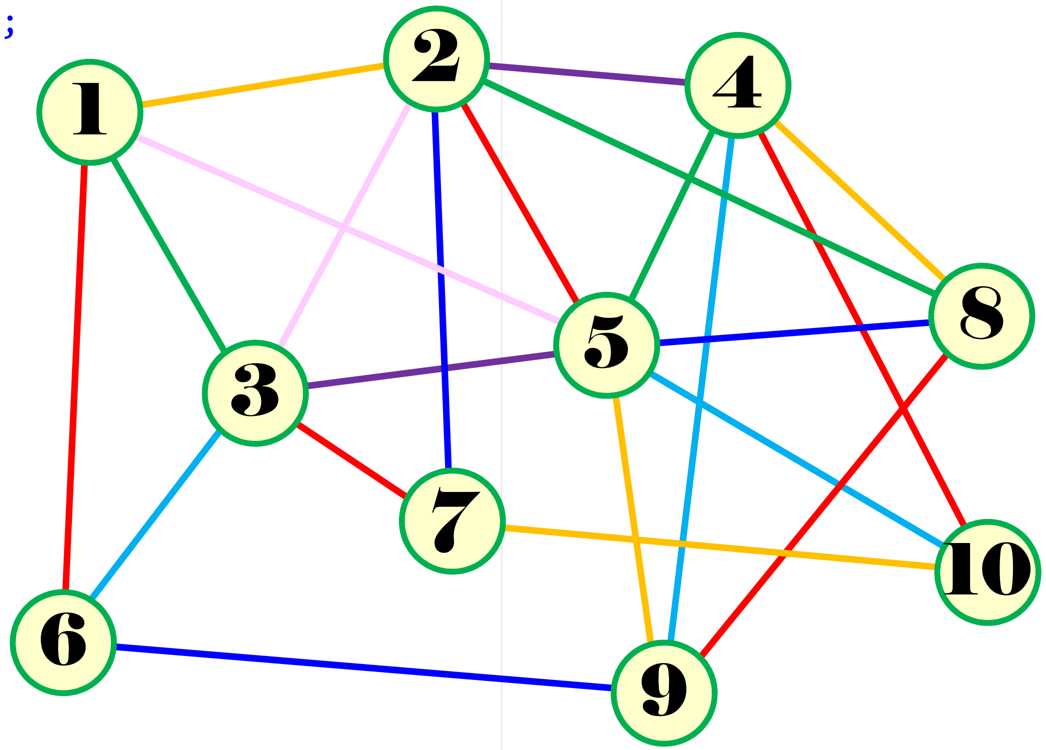


- $d_{max}=7$, $|E|=22$ より，使用色数 $k \in [7, 22]$

辺彩色問題をCPLEXで解く

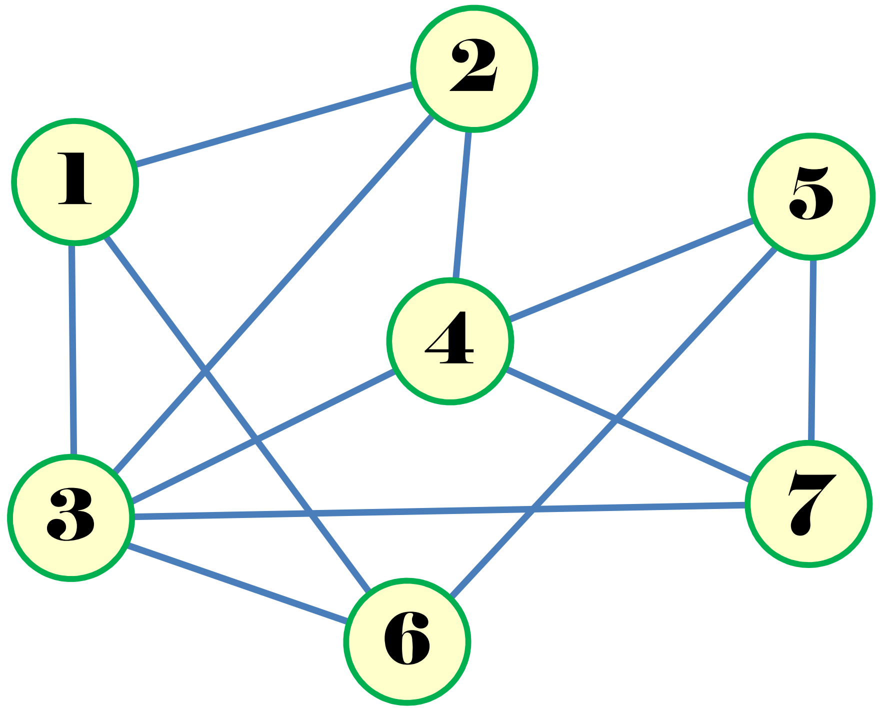
➢ 結果([解]タブ)

```
// solution (optimal) with objective 7
// Quality Incumbent solution:
// MILP objective                                   7.0000000000e+00
// MILP solution norm |x| (Total, Max)              2.90000e+01  1.00000e+00
// MILP solution error (Ax=b) (Total, Max)          0.00000e+00  0.00000e+00
// MILP x bound error (Total, Max)                  0.00000e+00  0.00000e+00
// MILP x integrality error (Total, Max)            0.00000e+00  0.00000e+00
// MILP slack bound error (Total, Max)              0.00000e+00  0.00000e+00
//
y =  [1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
x = [[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 1 0 0 0 0 0   最適解    0 0 0 0 0 0 0 0 0]
     [0 0 1 0 0 0 0             0 0 0 0 0 0 0 0 0]
     [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]];
```



E={(1,2),(1,3),(1,5),(1,6),(2,3),(2,4),(2,5),(2,7),
(2,8),(3,5),(3,6),(3,7),(4,5),(4,8),(4,9),(4,10),
(5,8),(5,9),(5,10),(6,9),(7,10),(8,9)}

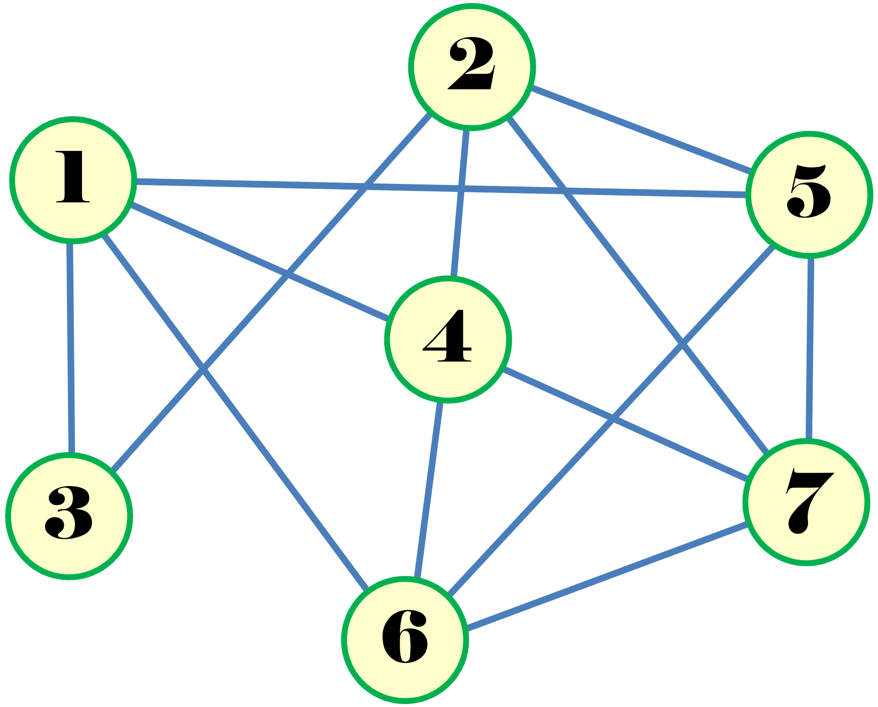
【演習】辺彩色問題を解く



- ex3) グラフ $G = (V, E)$
  - 点集合 $V = \{1,2,3,4,5,6,7\}$,
  - 枝集合 $E = \{1,2,\ldots\} = \{(1,2), (1,3), (1,6), (2,3), (2,4), (3,4), (3,6), (3,7), (4,5), (4,7), (5,6), (5,7)\}$

- 問
  1. $|V| = ?$ $|E| = ?$
  2. 接続行列A をつくれ
  3. 例1と同様に変数を設定し，定式化せよ
  4. 整数計画ソルバー（cplex）を用いて，辺彩色をせよ
  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，辺彩色をせよ
  7. 整数計画ソルバー（gurobi）とpython で解き，辺彩色をせよ
  8. 結果を networkx でグラフ描画せよ

【演習】辺彩色問題を解く



- ex4) グラフ $G = (V, E)$
  - 点集合 $V = \{1,2,3,4,5,6,7\}$,
  - 枝集合 $E = \{1,2,\ldots\} = \{(1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,7), (4,6), (4,7), (5,6), (5,7), (6,7)\}$

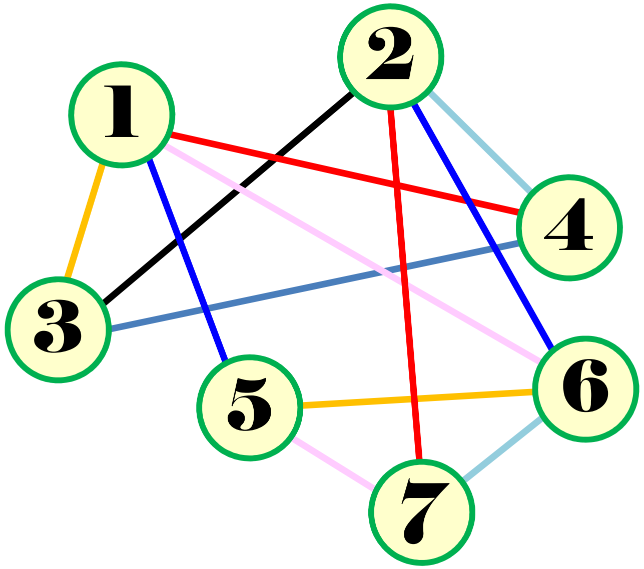
- 問
  1. $|V| = ?$ $|E| = ?$
  2. 接続行列A をつくれ
  3. 例1と同様に変数を設定し，定式化せよ
  4. 整数計画ソルバー（cplex）を用いて，辺彩色をせよ
  5. oplrun を使って，mod file / dat file から lp file を作れ
  6. 整数計画ソルバー（gurobi）で5のlp file を解き，辺彩色をせよ
  7. 整数計画ソルバー（gurobi）とpython で解き，辺彩色をせよ
  8. 結果を networkx でグラフ描画せよ

辺彩色問題となる事例

- **辺彩色問題 edge coloring problem**
  - 無向グラフ $G = (V, E)$
    - 点集合 $V = \{1, 2, \ldots, n\}$，枝集合 $E=\{1,2,\ldots,m\}$，$|V|=n$, $|E|=m$
  - 辺彩色 = 隣接枝に異なる色を割り当てるとき何色必要か？

- 枝数 $|E|=m$ より，自明解 m色（全枝に異なる色を割当）
- 「点の最大次数＝最小必要色数」だとわかる（最大次数=d_{max} としよう）
- 目的 = 色数k が最小の割当を求める

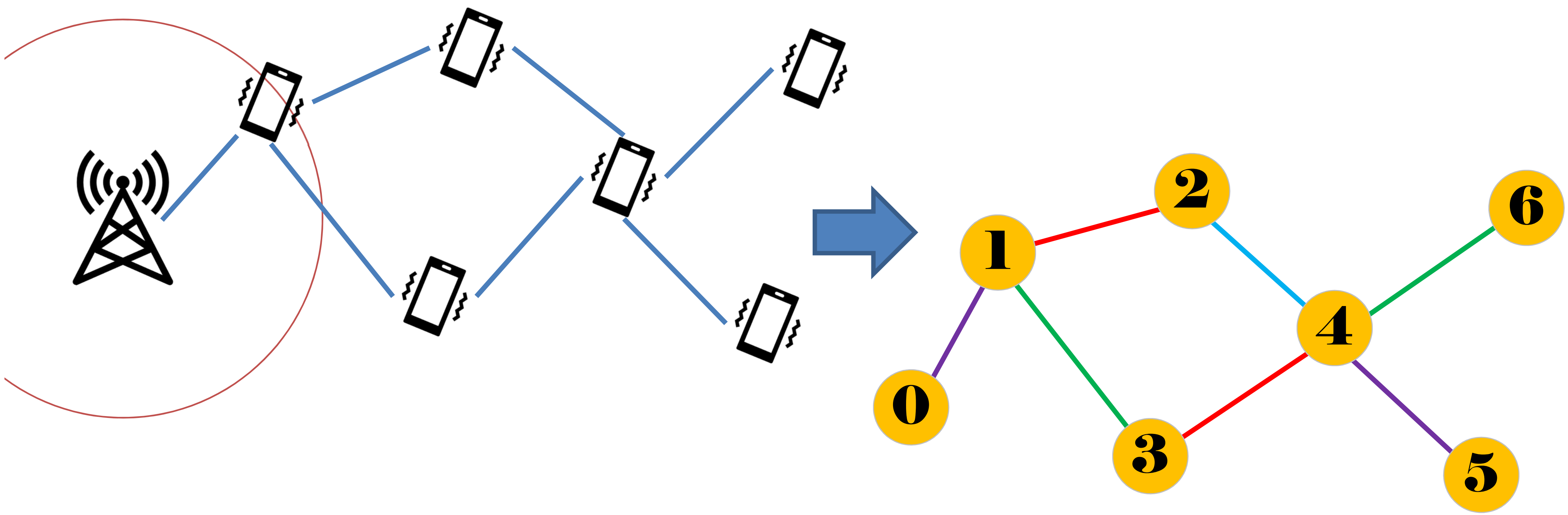
※ $k \in [d_{max}, m]$ （使用色数 k は d_{max} 以上 m 以下）



d_{max}=4, $|E|=m=12$より，この例では最小4色必要で，最大でも12色だと（解く前に）わかる

辺彩色問題となる事例

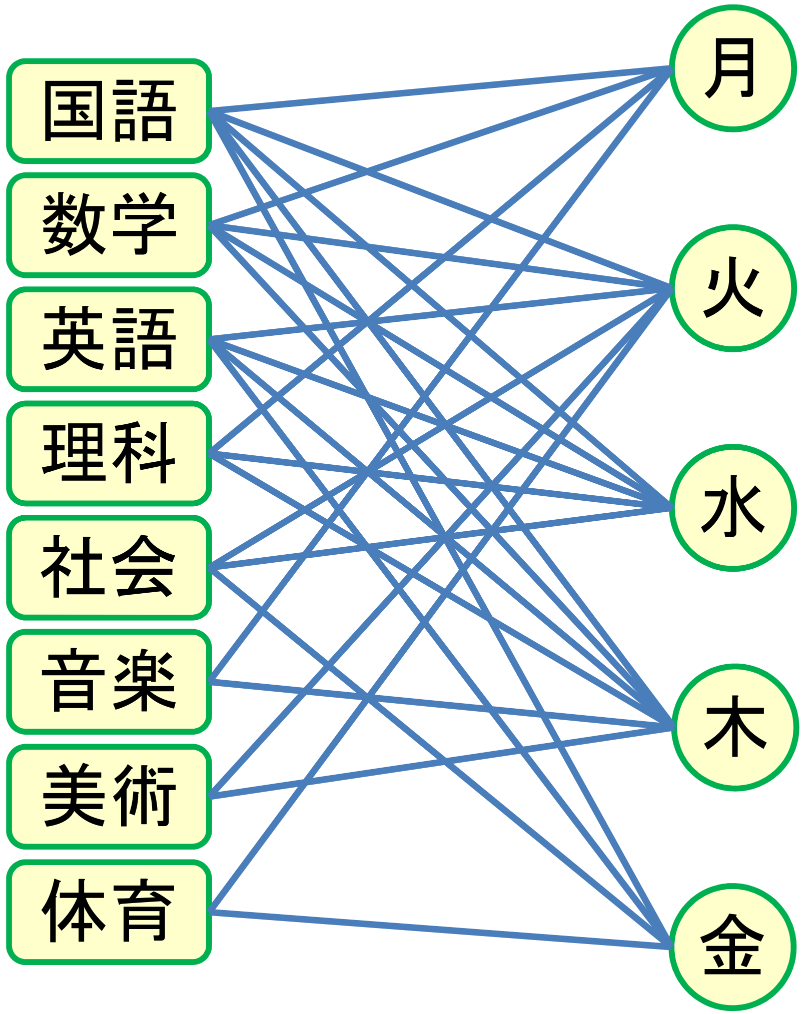
- 例1）マルチホップ方式の無線通信のチャンネル割当
  - 無線通信においては，通常，各端末は基地局の電波範囲内に居なければ通信が出来ない．端末同士の直接通信を可能とすることで，端末を経由し，基地局内にない端末も通信が可能となる．これをマルチホップ方式の無線通信とよぶ．基地局がない山中や僻地での通信を可能とする
  - 各端末へのチャンネル割当を行うが，中継端末は複数の隣接端末のチャンネルを変える必要がある．管理やコスト面からチャンネル数は最小にしたい
  - 各端末を点とし，端末間通信を行う端末間を枝で結んだグラフを考えると，このグラフの辺彩色問題の答えがチャンネル割当になる



辺彩色問題となる事例

- 例2）時間割の作成
  - ある1クラスの時間割を考える（各科目の担当者は考慮しない）
  - 週5（月～金）で全30授業．各科目実施曜日決定済，同一曜日は各最大1回
    - 週5科目：国語（月～金）
    - 週4科目：数学（月～木），英語（火～金）
    - 週3科目：理科（月水木），社会（火水金）
    - 週2科目：音楽（月木），美術（火木），体育（火金）
    - 週1科目：その他5科目（どこでも配置可）
  - 各曜日6時限目までに全て配置できればベストだが多少ばらついても良い

		月	火	水	木	金
色1→	1					
色2→	2					
色3→	3					
色4→	4					
色5→	5					
色6→	6					
色7→	7					



辺彩色問題となる事例

➢ 例2）時間割の作成　接続行列

枝＼点	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	(国,月)	(国,火)	(国,水)	(国,木)	(国,金)	(数,月)	(数,火)	(数,水)	(数,木)	(英,火)	(英,水)	(英,木)	(英,金)	(理,月)	(理,水)	(理,木)	(社,火)	(社,水)	(社,金)	(音,月)	(音,木)	(美,火)	(美,木)	(体,火)	(体,金)
国	1	1	1	1	1																				
数						1	1	1	1																
英										1	1	1	1												
理														1	1	1									
社																	1	1	1						
音																				1	1				
美																						1	1		
体																								1	1
月	1					1								1						1					
火		1					1			1							1					1		1	
水			1					1			1				1			1							
木				1					1			1				1					1		1		
金					1								1						1						1

```
// solution (optimal) with objective 6
  y = [1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
 1x =[[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  国語
  2   [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
  3   [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
  4   [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
  5   [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
  6   [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  数学
  7   [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
  8   [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
  9   [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 10   [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  英語
 11   [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 12   [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 13   [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 14   [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]  理科
 15   [0 0 0 1 0 0 0 0 0]
 16   [1 0 0 0 0 0 0 0 0]
 17   [0 0 1 0 0 0 0 0 0]  社会
 18   [0 0 0 0 0 1 0 0 0]
 19   [1 0 0 0 0 0 0 0 0]
 20   [0 0 0 0 1 0 0 0 0]  音楽
 21   [0 1 0 0 0 0 0 0 0]
 22   [0 0 0 1 0 0 0 0 0]  美術
 23   [0 0 0 0 1 0 0 0 0]
 24   [0 1 0 0 0 0 0 0 0]  体育
 25   [0 0 0 0 1 0 0 0 0]];
```

	月	火	水	木	金
1	国語	英語	数学	理科	社会
2		体育	英語	音楽	国語
3		社会	国語	数学	英語
4	数学	美術	理科	国語	
5	音楽	数学		美術	体育
6	理科	国語	社会	英語	
7