

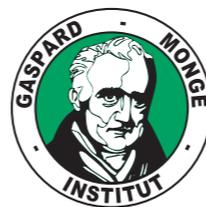
Méthodes et modélisation pour l'optimisation

M1 informatique, 2017–2018
05 — Modélisation SAT

UP

EM

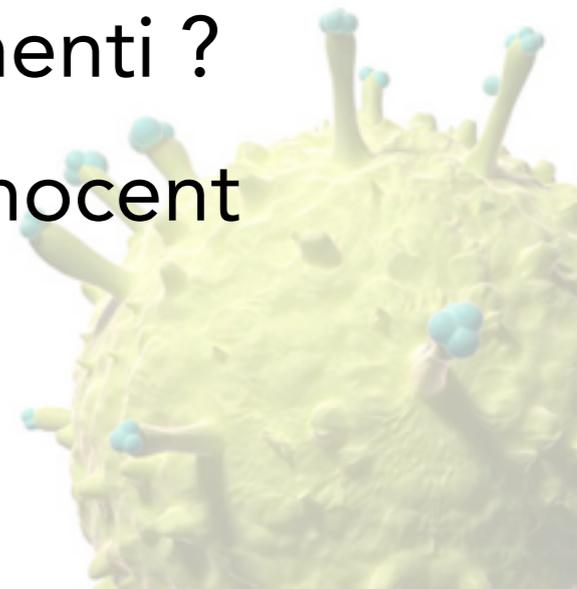
UNIVERSITÉ PARIS-EST
MARNE-LA-VALLÉE



INSTITUT D'ÉLECTRONIQUE
ET D'INFORMATIQUE
GASPARD-MONGE

Mystère

- ▶ Trois étudiants **Jean**, **Paulette** et **Nicolas** sont accusés d'avoir introduit un virus dans la salle informatique. Pendant leur audition, ils prétendent ceci :
 - ▶ Jean: "C'est **Paulette** qui l'a fait et **Nicolas** est innocent."
 - ▶ Paulette: "Si **Jean** est coupable, alors **Nicolas** l'est aussi."
 - ▶ Nicolas: "Ce n'est pas **moi**. C'est un des **deux autres** ou peut-être les deux."
1. Est-ce que les trois déclarations sont contradictoires ?
 2. Supposons qu'ils soient tous coupables, qui a menti ?
 3. Supposons que personne n'ait menti, qui est innocent et qui est coupable ?



Vocabulaire

- ▶ **Variable booléenne** : x prend la valeur 1 (vrai) ou 0 (faux)
- ▶ **Formule propositionnelle** : expression constituée de
 - ▶ variables x
 - ▶ négations $\neg x$ "NON x "
 - ▶ conjonctions $x \wedge y$ " x ET y "
 - ▶ disjonctions $x \vee y$ " x OU y "
 - ▶ implications $x \rightarrow y$ "SI x , ALORS y "
- ▶ Une **affectation satisfaisante** d'une formule est une affectation de variables aux valeurs (0 ou 1) qui rend la formule vraie
- ▶ Une formule est **satisfaisable** si elle a une affectation satisfaisante

Forme normale conjonctive

- ▶ **Littéral** : $x, \neg x$
- ▶ **Clause** (disjonction de **littéraux**) : $x \vee y \vee \neg z$
- ▶ Formule en **forme normale conjonctive** (CNF)
(conjonction de **clauses**) : $(x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z)$

Exercice : quelles formules sont en CNF ?

A. $(\neg x \vee \neg y) \wedge (\neg z \vee \neg w)$

B. $x \rightarrow y$

C. $\neg x \wedge y$

D. $(x \wedge y) \vee (z \wedge w)$

E. $(\neg(x \vee y)) \wedge (z \vee w)$

Mise en forme CNF

- ▶ Distributivité : $x \vee (y \wedge z) \Leftrightarrow (x \vee y) \wedge (x \vee z)$
- ▶ de Morgan : $\neg(x \vee y) \Leftrightarrow \neg x \wedge \neg y$
- ▶ Implication : $(x \rightarrow y) \Leftrightarrow \neg x \vee y$
- ▶ Associativité : $x \vee (y \vee z) \Leftrightarrow (x \vee y) \vee z \Leftrightarrow x \vee y \vee z$

Exercice : mettez les formules suivantes en CNF

A. $(x \wedge y) \vee (z \wedge w)$

B. $\neg(x \rightarrow y)$

C. $(\neg(x \vee y)) \wedge (z \rightarrow w)$

Le problème SAT

- ▶ Le problème de trouver une affectation satisfaisante à une formule CNF ou de démontrer qu'il n'y en a aucune s'appelle **SAT** (satisfaisabilité booléenne)
- ▶ SAT est **NP-difficile** ; tout problème dans NP peut être exprimé comme une formule en CNF
- ▶ On décrit souvent l'entrée à SAT par un ensemble de clauses : $(x_1 \vee \neg x_2)$, $(\neg x_1 \vee x_3)$, $(\neg x_2 \vee \neg x_3)$ au lieu de la formule $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3)$
- ▶ Alors, SAT est le problème de trouver une affectation de variables qui rend toutes les clauses vraies simultanément

Mystère en CNF

"C'est *Paulette* qui l'a fait et *Nicolas* est innocent."

$$X_P, \neg X_N$$

"Si *Jean* est coupable, alors *Nicolas* l'est aussi."

$$\neg X_J \vee X_N$$

"Ce n'est pas *moi*. C'est un des *deux autres* ou peut-être les deux."

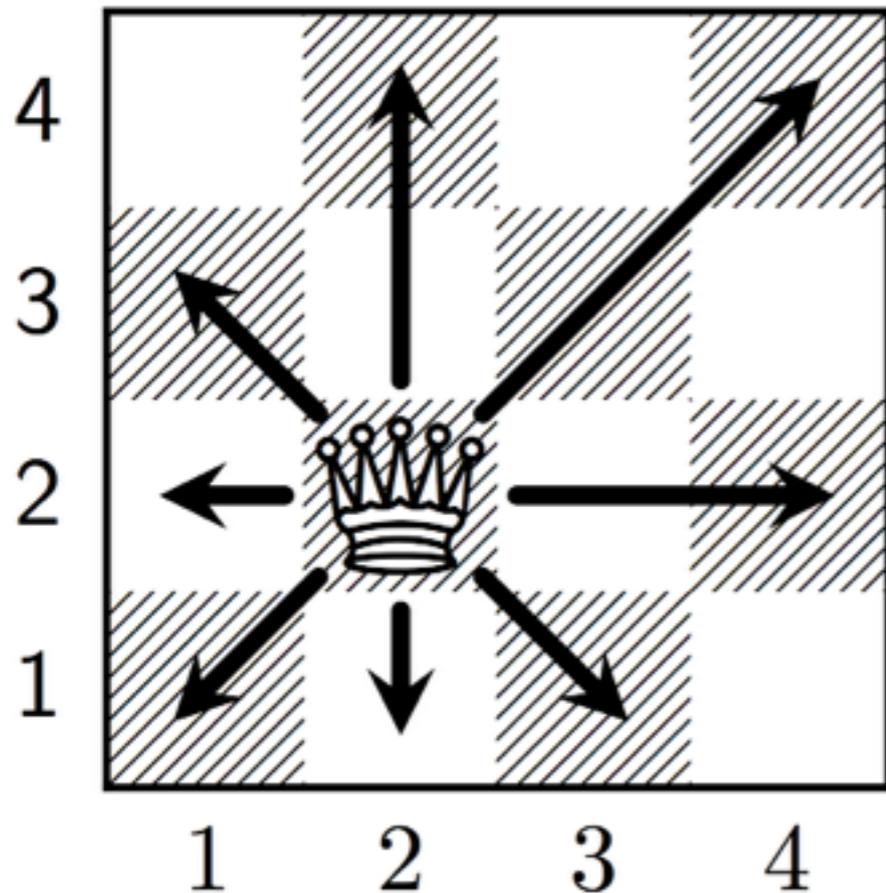
$$\neg X_N, X_P \vee X_J$$

La CNF a 4 clauses

$$X_P, \neg X_N, \neg X_J \vee X_N,$$

$$X_P \vee X_J$$

4 reines



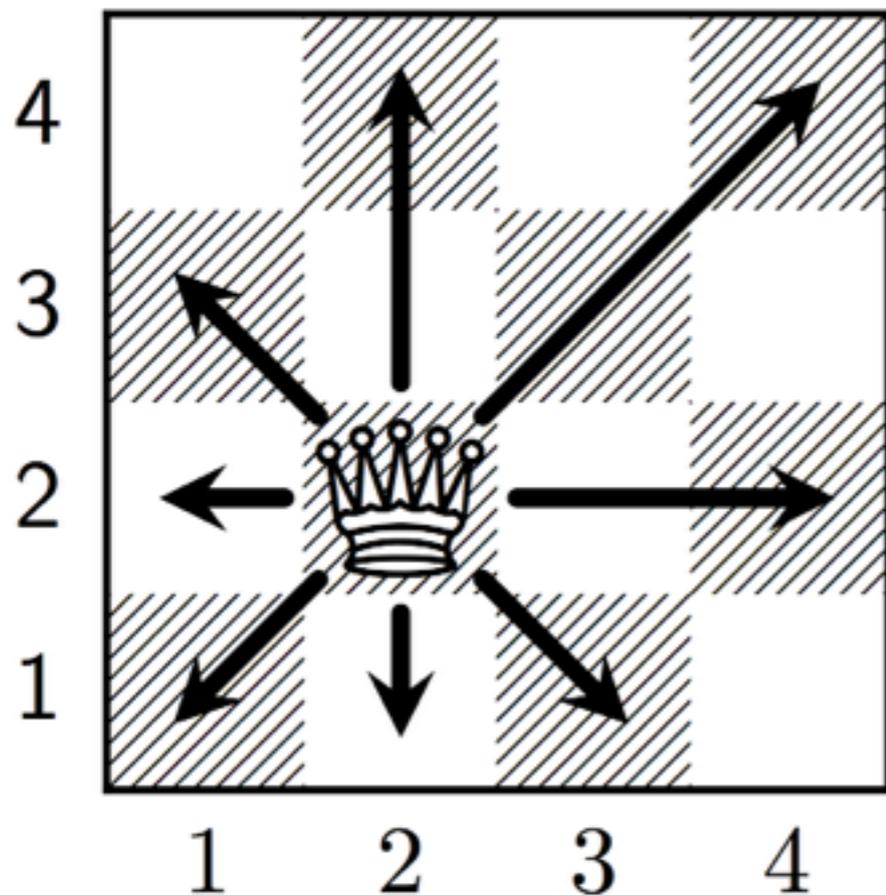
Est-ce qu'il est possible de placer les 4 reines de manière à ce qu'aucune d'entre elles ne soit en prise ?

- ▶ Modéliser ce problème par une formule CNF
- ▶ C-à-d, trouver une formule CNF telle que chaque solution satisfaisante correspond à un placement valide de 4 reines

4 reines

► Variables x_{ij}

intention : $x_{ij} = 1$ si la reine i est en rang i , colonne j



1. Au moins une reine dans chaque colonne

$$(x_{1j} \vee x_{2j} \vee x_{3j} \vee x_{4j})$$
$$j = 1, \dots, 4 \quad (4 \text{ clauses})$$

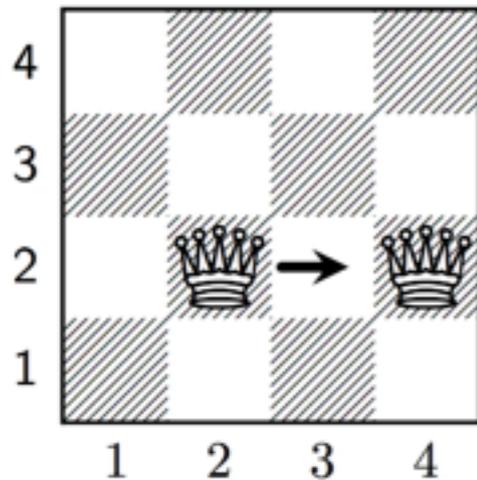
2. Il n'y a pas deux reines à (a,b) et à (c,d) si une reine à (a,b) peut attaquer (c,d)

$$(\neg x_{ab} \vee \neg x_{cd})$$

(beaucoup de clauses)

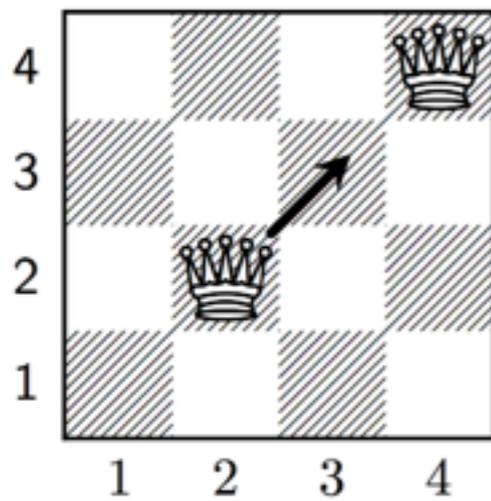
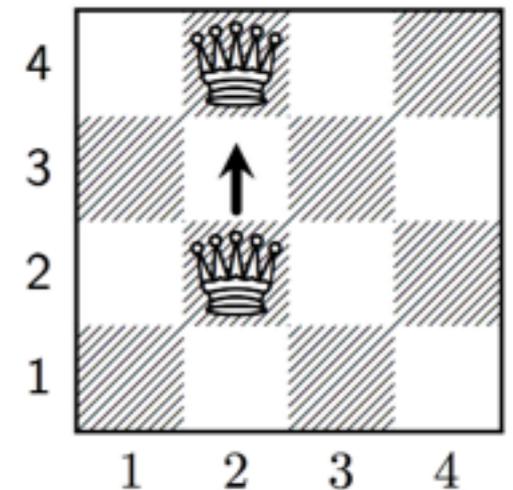
4 reines

- ▶ Quand est-ce qu'une reine à (a,b) peut attaquer (c,d) ?



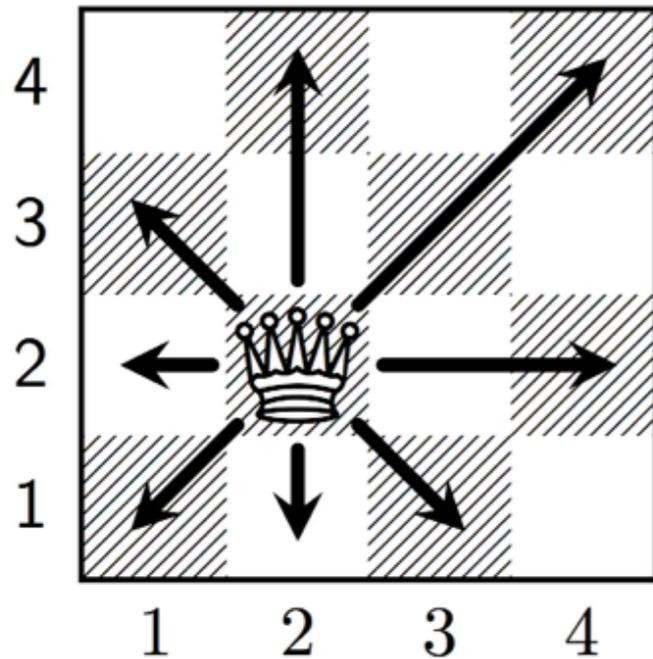
Attaque horizontale : $a = c, b \neq d$

Attaque verticale : $a \neq c, b = d$



Attaque diagonale : $|a - c| = |b - d|, (a,b) \neq (c,d)$

4 reines en Python



1. Il y a au moins une reine dans chaque colonne

```
for j in range(1,5):  
    clause = [var(i,j) for i in range(1,5)]  
    formula.append(clause)
```

2. Il n'y a pas deux reines à (a,b) et à (c,d) si une reine à (a,b) peut attaquer (c,d)

```
for a in range(1,5):  
    for b in range(1,5):  
        for c in range(1,5):  
            for d in range(1,5):  
                if (a,b) < (c,d): # ordre lexicographique  
                    if ((a == c) or # horizontale  
                        (b == d) or # verticale  
                        (abs(a-c) == abs(b-d))): # diagonale  
                        clause = [neg(var(a,b)), neg(var(c,d))]  
                        formula.append(clause)
```

Macros

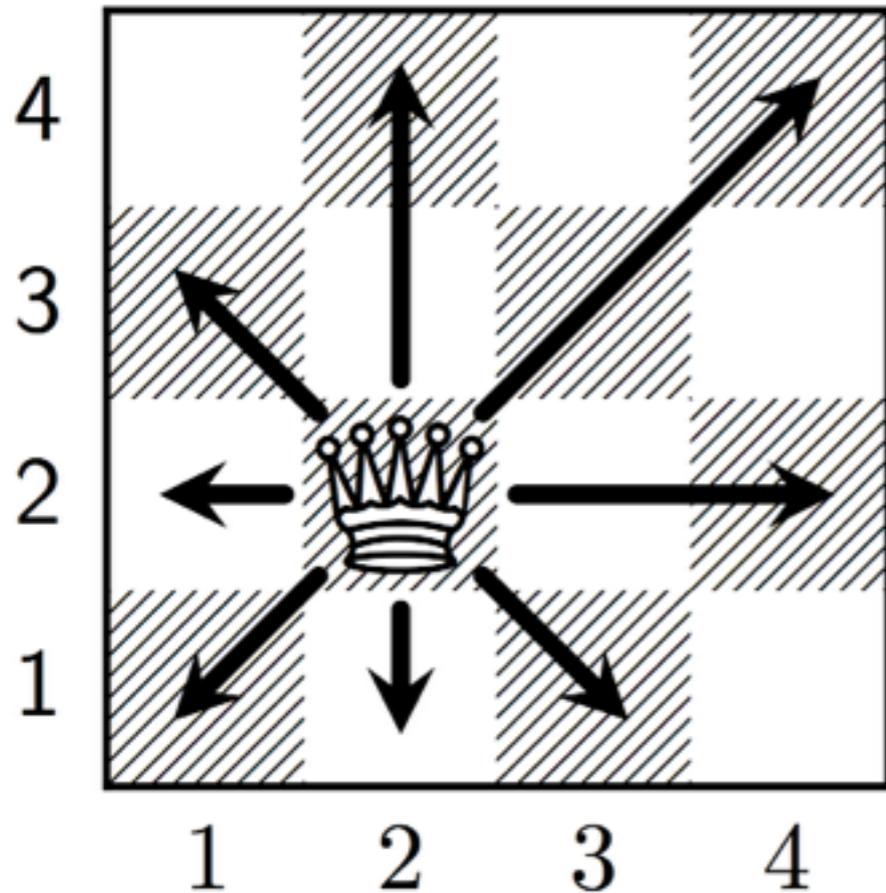
Il y a **au moins une** variable parmi x_1, x_2, \dots, x_n qui est vraie

$$\text{AtLeastOne}(x_1, x_2, \dots, x_n)$$
$$\Leftrightarrow$$
$$(x_1 \vee x_2 \vee \dots \vee x_n)$$

Il y a **au plus une** variable parmi x_1, x_2, \dots, x_n qui est vraie

$$\text{AtMostOne}(x_1, x_2, \dots, x_n)$$
$$\Leftrightarrow$$
$$(\neg x_1 \vee \neg x_2), (\neg x_1 \vee \neg x_3), \dots, (\neg x_{n-1} \vee \neg x_n)$$

4 reines



1. Au moins une reine dans chaque colonne

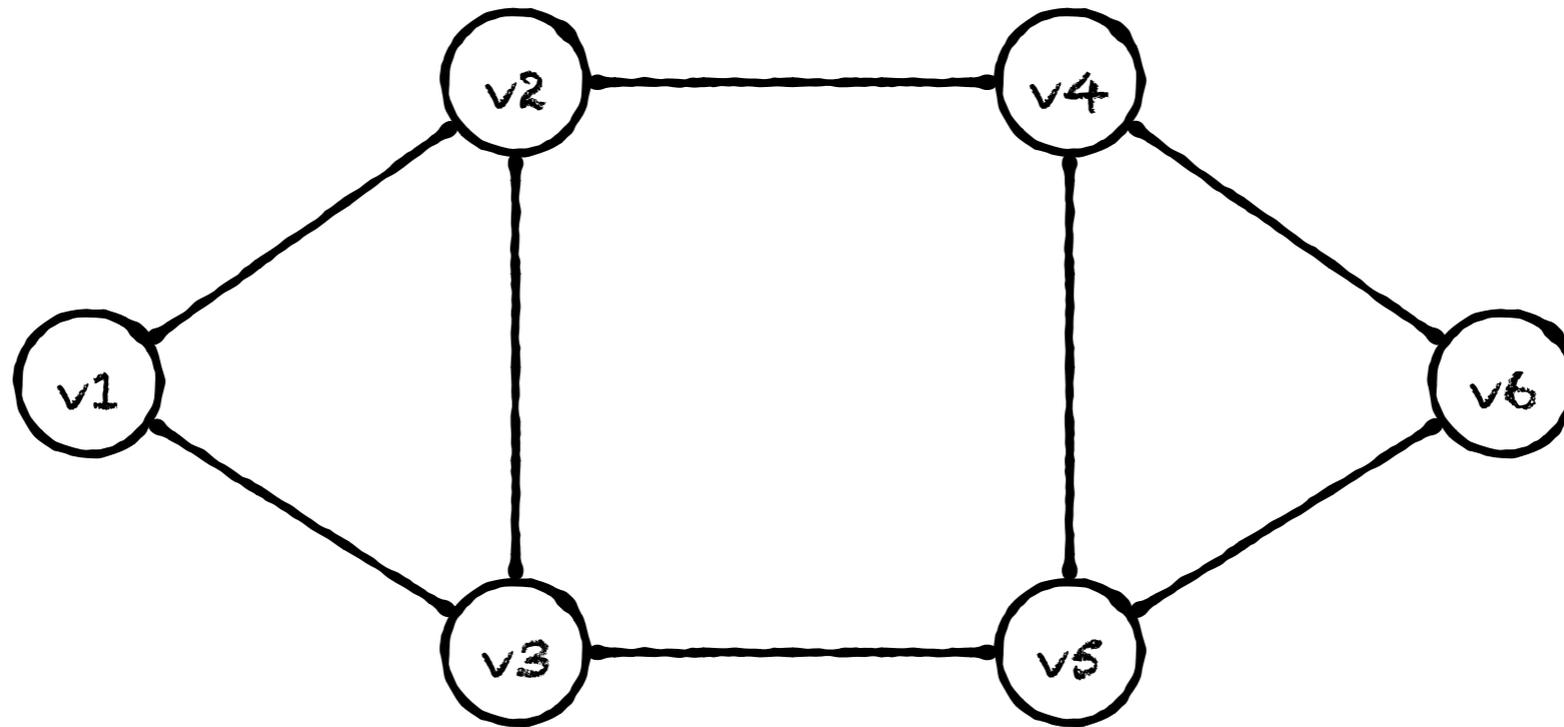
$\text{AtLeastOne}(x_{1j}, x_{2j}, x_{3j}, x_{4j})$

2. Il n'y a pas deux reines à (a,b) et à (c,d) si une reine à (a,b) peut attaquer (c,d)

$\text{AtMostOne}(x_{ab}, x_{cd})$

On se permet d'utiliser les macros AtLeastOne et AtMostOne au td et à l'examen.

Coloration de graphe

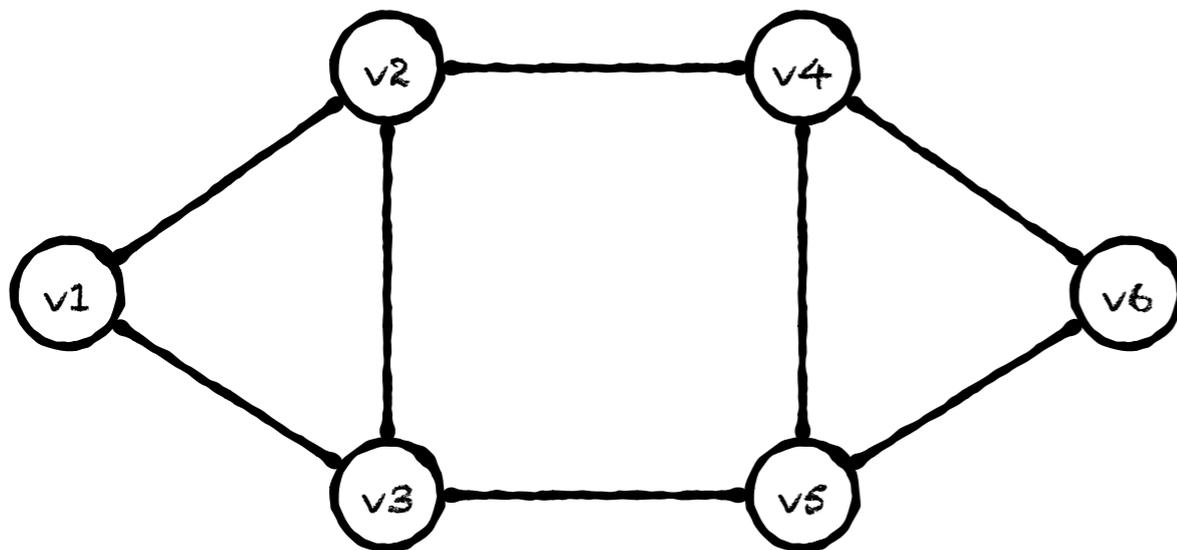


- ▶ Le graphe a-t-il une coloration avec trois couleurs ?
- ▶ Couleurs : 1, 2, 3
- ▶ Introduire une variable x_{ic} pour toute $i = 1, \dots, 6$ et $c = 1, 2, 3$
- ▶ La variable $x_{ic} = 1$ indique que le sommet v_i prend la couleur c

Coloration de graphe

► Variables x_{ic}

intention : $x_{ic} = 1$ si le sommet v_i prend la couleur c



3. Les deux sommets d'une arête (v_i, v_j) ne prennent pas la même couleur c

$\text{AtMostOne}(x_{ic}, x_{jc})$

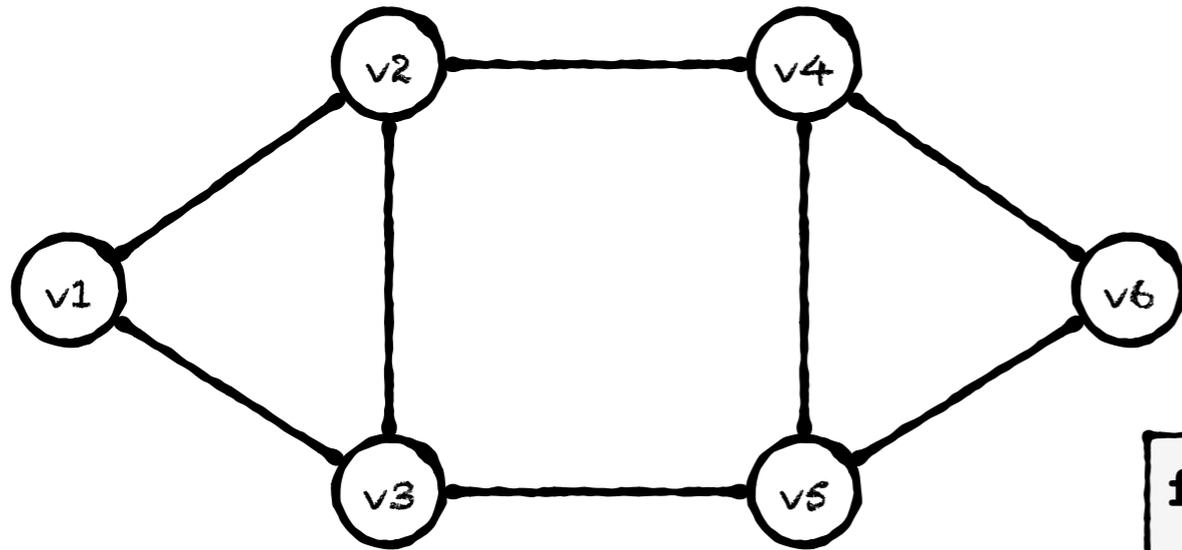
1. Chaque sommet $i = 1, \dots, 6$ prend **au moins une** couleur

$\text{AtLeastOne}(x_{i1}, x_{i2}, x_{i3})$

2. Chaque sommet $i = 1, \dots, 6$ prend **au plus une** couleur

$\text{AtMostOne}(x_{i1}, x_{i2}, x_{i3})$

Coloration de graphe en Python



1. Chaque sommet $i = 1, \dots, 6$ prend **au moins une** couleur
2. Chaque sommet $i = 1, \dots, 6$ prend **au plus une** couleur

```
for i in nodes:  
    vars = [var(i,0), var(i,1), var(i,2)]  
    clauses = at_least_one(vars)  
    formula.extend(clauses)  
    clauses = at_most_one(vars)  
    formula.extend(clauses)
```

3. Les deux sommets d'une arête (v_i, v_j) ne prennent pas la même couleur c

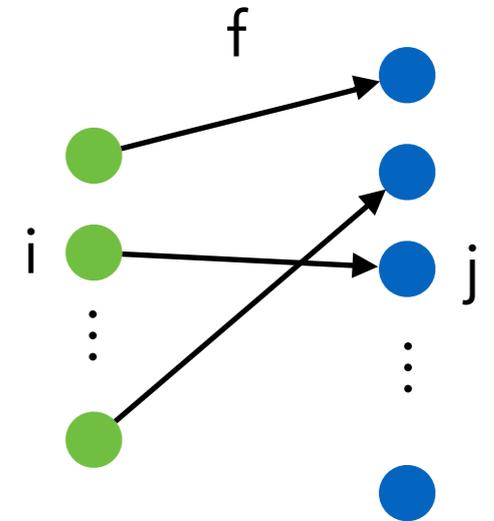
```
for e in edges:  
    i, j = e  
    for c in range(3):  
        clauses = at_most_one([var(i,c), var(j,c)])  
        formula.extend(clauses)
```

Formaliser des affectations

- ▶ On a souvent besoin de formaliser une affectation

$$f : \{1, \dots, n\} \rightarrow \{1, \dots, m\}$$

- ▶ affecter les **reines** aux **colonnes**
- ▶ affecter les **sommets** aux **couleurs**



- ▶ Variables de décision *intention* : $x_{ij} = 1$ si $(i,j) \in f$
- ▶ Pour que f soit une affectation (et non pas seulement une relation binaire), on introduit des contraintes qui disent

“pour tout i , il y a exactement un j t.q. $(i,j) \in f$ ”

AtMostOne($x_{i1}, x_{i2}, \dots, x_{im}$)

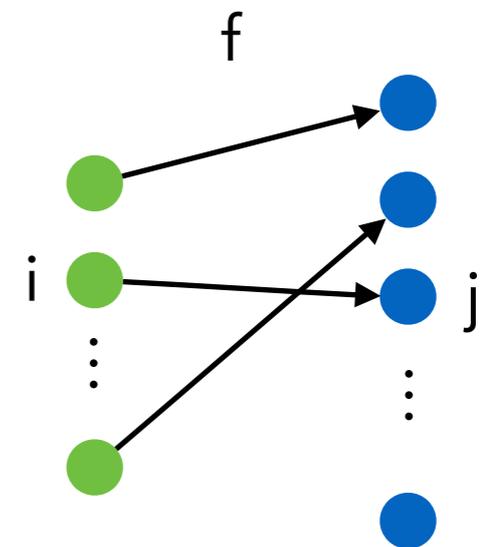
AtLeastOne($x_{i1}, x_{i2}, \dots, x_{im}$)

Formaliser des injections

- ▶ Parfois, l'affectation $f : \{1, \dots, n\} \rightarrow \{1, \dots, m\}$ est injective
 - ▶ pas deux reines dans une même colonne
- ▶ Pour imposer cette condition supplémentaire sur f , on introduit des contraintes qui disent

“pour tout j , il y a au plus un i t.q. $(i,j) \in f$ ”

AtMostOne($x_{1j}, x_{2j}, \dots, x_{nj}$)



Emploi du temps

| | Cours | Enseignant |
|------------|----------------|------------|
| i=1 | MMPO CM | Thapper |
| 2 | Complexité CM | Nicaud |
| 3 | MMPO TD1 | Hubard |
| | Compression CM | Nicaud |
| | ... | ... |

| | Lun | Mar | Mer | Jeu | Ven |
|----------------|------------|-----|-----|-----|-----|
| 08h30 10h30 | j=1 | | | | |
| 10h45 12h45 | 2 | | | | |
| 14h00 16h00 | 3 | | | | |
| 16h15 18h15 | | | | | |

| Dis | séances $i = 1, \dots, n$ | |
|--------|---------------------------|--|
| T | er 10h45-16h00 | |
| Hubard | lun, ven | |
| ... | .. | |

créneaux $j = 1, \dots, m$

Emploi du temps

▶ Variables x_{ij}

intention : $x_{ij} = 1$ si la séance i est affectée au créneau j

On modélise une injection $f : \{ \text{séance} \} \rightarrow \{ \text{créneau} \}$

1. La séance $i = 1, \dots, n$ doit avoir lieu au plus une fois :

$$\text{AtMostOne}(x_{i1}, \dots, x_{im})$$

2. Soit $j(1), \dots, j(d)$ les disponibilités de l'enseignant de la séance $i = 1, \dots, n$. La séance i doit avoir lieu sur un de ces créneaux :

$$\text{AtLeastOne}(x_{ij(1)}, \dots, x_{ij(d)})$$

3. Le créneau $j = 1, \dots, m$ doit concerner au plus une séance :

$$\text{AtMostOne}(x_{1j}, \dots, x_{nj}) \quad (\text{injectivité})$$

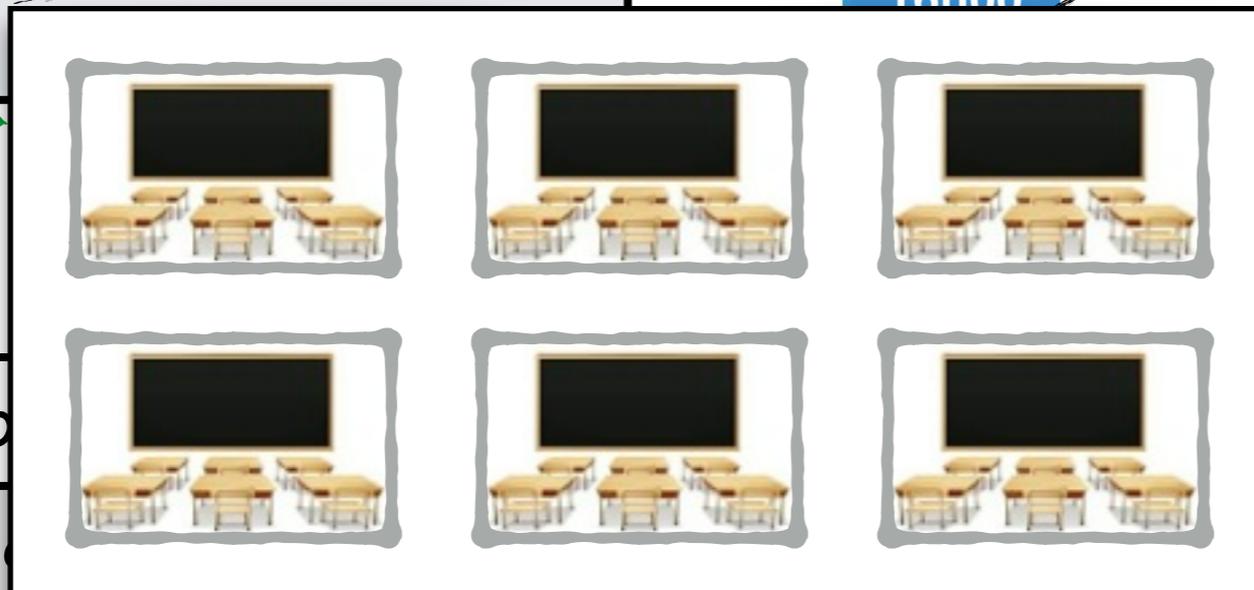
Emploi du temps 2

(créneaux, salle)

$j = 1, \dots, m$

| | |
|-------------|--------|
| Complexité | Nicaud |
| MMPO | Hubard |
| Compression | Nicaud |
| ... | |

| | Lun | Mar | Mer | Jeu | Ven |
|-------|-----|-----|-----|-----|-----|
| 08h30 | | | | | |
| 10h30 | | | | | |
| 10h45 | | | | | |
| 12h45 | | | | | |
| 14h00 | | | | | |
| 16h00 | | | | | |



| | | | | |
|--------|--|--|----------|------|
| Dispo | | | | |
| Th | | | | 6h00 |
| Hubard | | | lun, ven | |
| ... | | | ... | |

Emploi du temps 2

▶ Variables x_{ij}

intention : $x_{ij} = 1$ si la séance i est affectée **au couple** (créneau,salle) j

1. La séance $i = 1, \dots, n$ doit avoir lieu au plus une fois :

$$\text{AtMostOne}(x_{i1}, \dots, x_{im})$$

2. Soit $j(1), \dots, j(d)$ les disponibilités de l'enseignant de la séance $i = 1, \dots, n$. La séance i doit avoir lieu sur un de ces créneaux :

$$\text{AtLeastOne}(x_{ij(1)}, \dots, x_{ij(d)})$$

3. Le couple (créneau,salle) $j = 1, \dots, m$ doit concerner au plus une séance :

$$\text{AtMostOne}(x_{1j}, \dots, x_{nj}) \quad (\text{injectivité})$$

Emploi du temps 2

- ▶ Variables x_{ij}

intention : $x_{ij} = 1$ si la séance i est affectée **au couple** (créneau, salle) j

- ▶ Deux couples (i_1, j_1) et (i_2, j_2) sont **en conflit** si j_1 et j_2 ont lieu au même créneau (horaires) et, soit les séances i_1 et i_2 sont assurées par le même enseignant, soit elles ont des étudiants en commun (même groupe de td par exemple)

4. Pour tout couple (i_1, j_1) et (i_2, j_2) qui sont en conflit, on introduit

$$\text{AtMostOne}(x_{i_1 j_1}, x_{i_2 j_2}) \Leftrightarrow (\neg x_{i_1 j_1} \vee \neg x_{i_2 j_2})$$

AtMost-k(...), $k > 1$

- ▶ Comment peut-on exprimer qu'au plus 2 d'entre les variables x, y, z, u, v sont vraies ?
- ▶ "Si je sélectionne 3 d'entre elles, au moins 1 est fausse"

$$\text{AtMost-2}(x, y, z, u, v)$$

\Leftrightarrow

$$(\neg x \vee \neg y \vee \neg z), (\neg x \vee \neg y \vee \neg u), (\neg x \vee \neg y \vee \neg v), \dots, (\neg z \vee \neg u \vee \neg v)$$

- ▶ Combien de clauses ?
sélection de 3 parmi 5 = $(5 \times 4 \times 3) / (3 \times 2 \times 1) = 10$
- ▶ Combien de clauses pour n variables ?
sélection de 3 parmi n = $n(n-1)(n-2)/6 = O(n^3)$

AtMost-k(x_1, \dots, x_n), $k > 1$

- ▶ On peut utiliser moins de clauses en introduisant $n \times k$ variables auxiliaires y_{ij} pour $i = 1, \dots, n$ et $j = 1, \dots, k$
- ▶ L'idée est de mettre $x_i \rightarrow y_{i1} \vee y_{i2} \vee \dots \vee y_{ik}$
- ▶ Au plus k parmi les y_{ij} peuvent être vraies
 - ▶ au plus 1 dans chaque colonne

| | | | | | |
|----------|----------|-----|----------|-----|----------|
| y_{11} | y_{12} | ... | y_{1j} | ... | y_{1k} |
| y_{21} | y_{22} | ... | y_{2j} | ... | y_{2k} |
| ... | ... | | ... | | ... |
| y_{i1} | y_{i2} | ... | y_{ij} | ... | y_{ik} |
| y_{n1} | y_{n2} | ... | y_{nj} | ... | y_{nk} |

AtMostOne($y_{1j}, y_{2j}, \dots, y_{nj}$)

AtMost-k(x_1, \dots, x_n), $k > 1$

- ▶ $n \times k$ variables auxiliaires
- ▶ $k \times n(n-1)/2 + n = O(n^2)$ clauses pour k fixe
- ▶ n^2 clauses pour $k = 2$

$$x_i \rightarrow y_{i1} \vee y_{i2} \vee \dots \vee y_{ik}$$

| | | | | | |
|----------|----------|-----|----------|-----|----------|
| y_{11} | y_{12} | ... | y_{1j} | ... | y_{1k} |
| y_{21} | y_{22} | ... | y_{2j} | ... | y_{2k} |
| ... | ... | | ... | | ... |
| y_{i1} | y_{i2} | ... | y_{ij} | ... | y_{ik} |
| y_{n1} | y_{n2} | ... | y_{nj} | ... | y_{nk} |

AtMostOne($y_{1j}, y_{2j}, \dots, y_{nj}$)

ETD 2 solution avec AtMost-k

► Variables x_{ij}

intention : $x_{ij} = 1$ si la séance i est affectée au créneau j

1. La séance $i = 1, \dots, n$ doit avoir lieu au plus une fois :

$$\text{AtMostOne}(x_{i1}, \dots, x_{im})$$

2. Soit $j(1), \dots, j(d)$ les disponibilités de l'enseignant de la séance $i = 1, \dots, n$. La séance i doit avoir lieu sur un de ces créneaux :

$$\text{AtLeastOne}(x_{ij(1)}, \dots, x_{ij(d)})$$

3. Le créneau $j = 1, \dots, m$ doit concerner **au plus k** séances :

$$\text{AtMost-k}(x_{1j}, \dots, x_{nj})$$

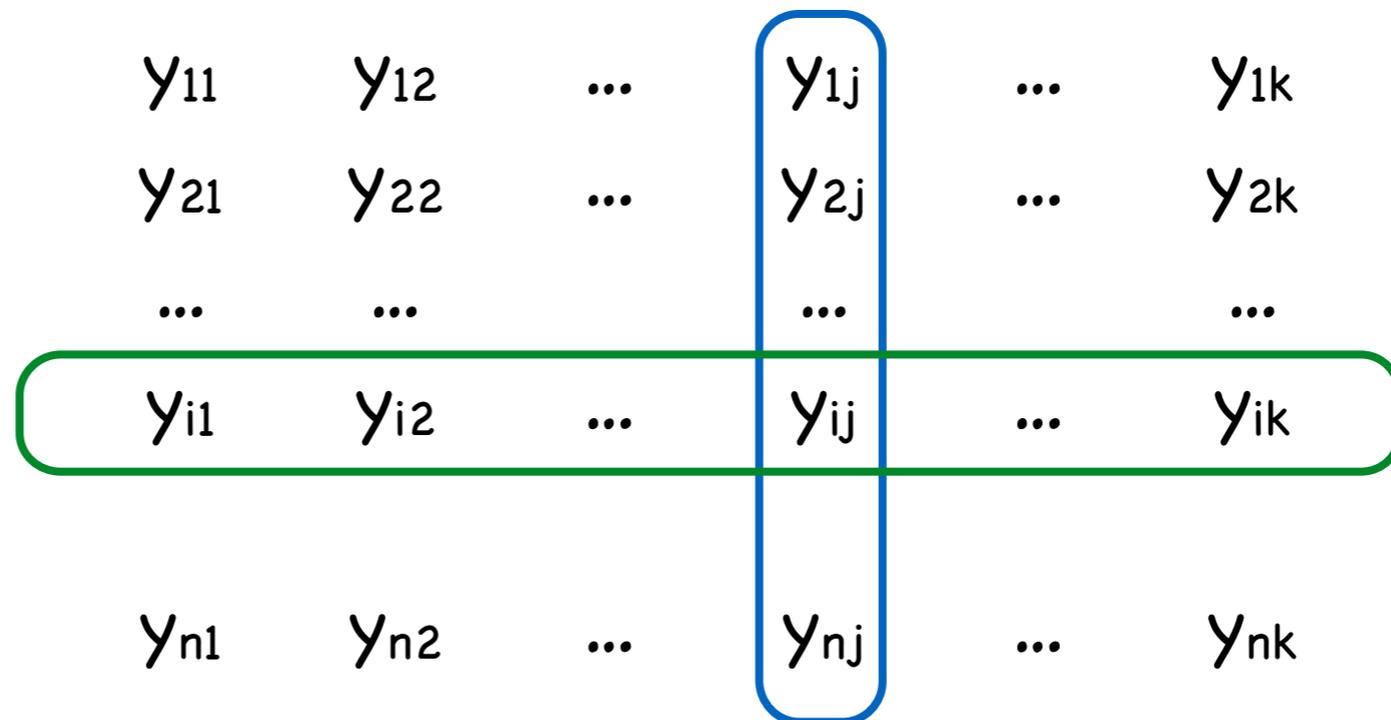
4. Pour tout couple (i_1, j_1) et (i_2, j_2) qui sont en conflit, on introduit

$$\text{AtMostOne}(x_{i_1j_1}, x_{i_2j_2}) \Leftrightarrow (\neg x_{i_1j_1} \vee \neg x_{i_2j_2})$$

AtLeast-k(x_1, \dots, x_n), $k > 1$

- ▶ On introduit $n \times k$ variables auxiliaires y_{ij} pour $i = 1, \dots, n$, $j = 1, \dots, k$ et on met $y_{i1} \vee y_{i2} \vee \dots \vee y_{ik} \rightarrow x_i$
- ▶ Au moins k rangs doivent être non-zéro
 - ▶ **au moins** 1 y_{ij} vraie dans chaque colonne
 - ▶ au plus 1 y_{ij} vraie dans chaque rang

AtMostOne($y_{i1}, y_{i2}, \dots, y_{ik}$)



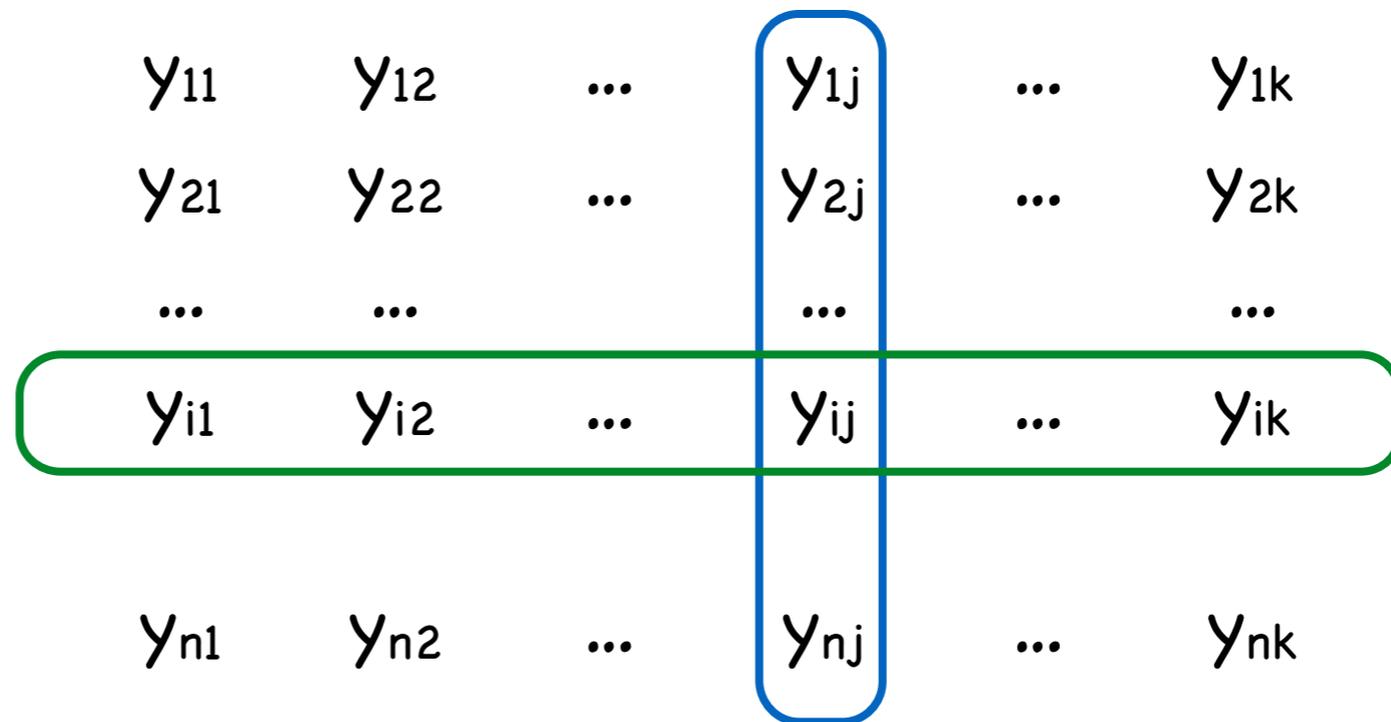
AtLeastOne($y_{1j}, y_{2j}, \dots, y_{nj}$)

AtLeast-k(x_1, \dots, x_n), $k > 1$

- ▶ $n \times k$ variables auxiliaires
- ▶ $k + n \times k(k-1)/2 + n \times k = O(n)$ clauses pour k fixe
- ▶ $3n+2$ clauses pour $k = 2$

$$y_{i1} \vee y_{i2} \vee \dots \vee y_{ik} \rightarrow x_i$$

AtMostOne($y_{i1}, y_{i2}, \dots, y_{ik}$)



AtLeastOne($y_{1j}, y_{2j}, \dots, y_{nj}$)