

# Résoudre des problèmes pratiques grâce à la théorie des graphes

Le problème du rock and roll

Nathalie Bertrand

chargée de recherche Inria Rennes

[nathalie.bertrand@inria.fr](mailto:nathalie.bertrand@inria.fr)

# Le problème du rock and roll

**Données** : des filles, des garçons



# Le problème du rock and roll

**Données** : des filles, des garçons



**Question** : qui danse avec qui pour maximiser le nombre de danseurs?



# Le problème du rock and roll

**Données** : des filles, des garçons



**Question** : qui danse avec qui pour maximiser le nombre de danseurs?

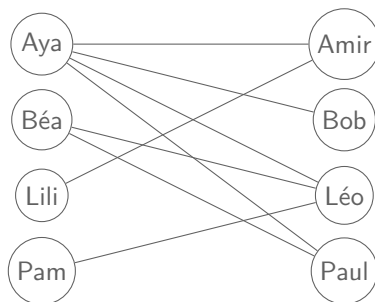


**Contraintes**: Lili veut danser avec Léo ou Paul; Amir avec Pam; Paul avec Lili, Pam ou Aya, etc.

# Modélisation du problème du rock and roll

Un **graphe** pour représenter la situation

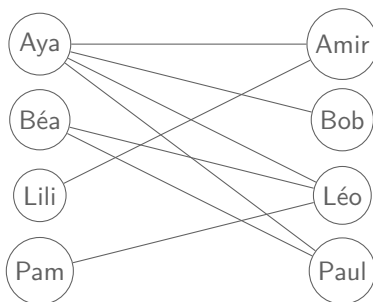
- ▶ **sommets** = danseurs
- ▶ **arcs** = couples possibles



# Modélisation du problème du rock and roll

Un **graphe** pour représenter la situation

- ▶ **sommets** = danseurs
- ▶ **arcs** = couples possibles

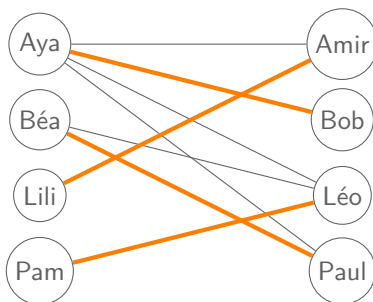


**Objectif** : calculer un ensemble d'arcs de cardinal le plus grand possible tel que chaque danseuse ou danseur apparaît au plus une fois.

# Modélisation du problème du rock and roll

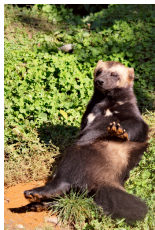
Un **graphe** pour représenter la situation

- ▶ **sommets** = danseurs
- ▶ **arcs** = couples possibles



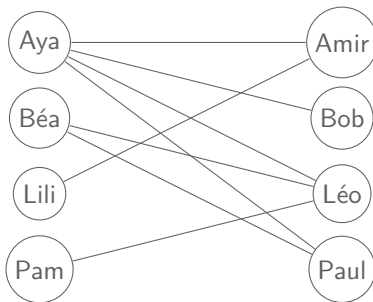
**Objectif** : calculer un ensemble d'arcs de cardinal le plus grand possible tel que chaque danseuse ou danseur apparaît au plus une fois.

# Résolution du problème du rock and roll



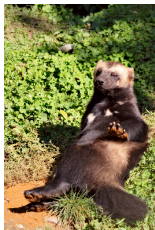
## Algorithme glouton

*Pour chaque fille de Aya à Pam  
sélectionner de Amir à Paul le premier prétendant libre*



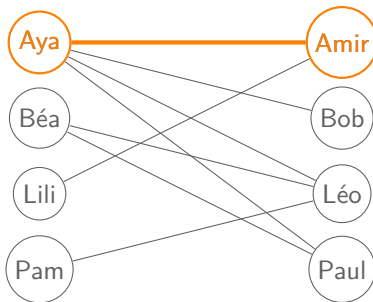


# Résolution du problème du rock and roll

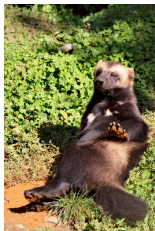


## Algorithme glouton

*Pour chaque fille de Aya à Pam  
sélectionner de Amir à Paul le premier prétendant libre*

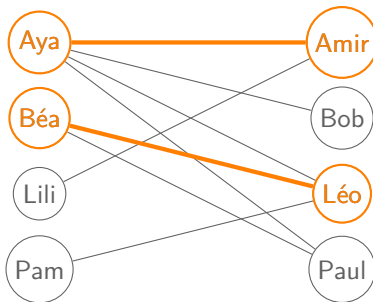


# Résolution du problème du rock and roll

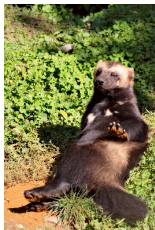


## Algorithme glouton

*Pour chaque fille de Aya à Pam  
sélectionner de Amir à Paul le premier prétendant libre*

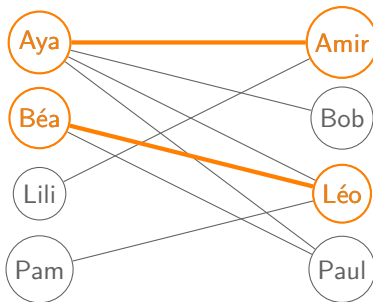


# Résolution du problème du rock and roll



## Algorithme glouton

*Pour chaque fille de Aya à Pam  
sélectionner de Amir à Paul le premier prétendant libre*



L'algorithme glouton ne calcule pas une solution optimale.

# Résolution du problème du rock and roll



## Algorithme de Hopcroft et Karp

*Initialiser* avec une affectation arbitraire

*Tant qu'* il existe un chemin améliorant

*faire* le ou exclusif de ce chemin avec l'affectation courante



# Résolution du problème du rock and roll

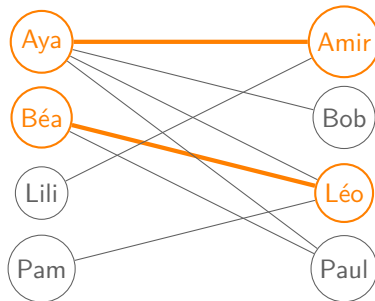


## Algorithme de Hopcroft et Karp

Initialiser avec une affectation arbitraire

Tant qu'il existe un chemin améliorant

faire le ou exclusif de ce chemin avec l'affectation courante



# Résolution du problème du rock and roll



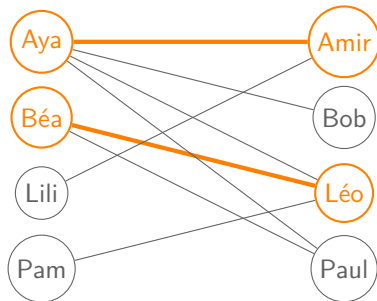
## Algorithme de Hopcroft et Karp



Initialiser avec une affectation arbitraire  
Tant qu'il existe un chemin améliorant  
faire le ou exclusif de ce chemin avec l'affectation courante

### Chemin améliorant

- ▶ commence et termine par un sommet libre ;
- ▶ alterne arcs libres et choisis ;
- ▶ voit chaque sommet au plus une fois.



# Résolution du problème du rock and roll



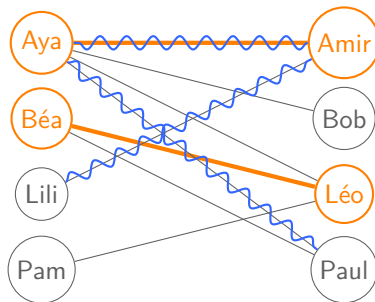
## Algorithme de Hopcroft et Karp



Initialiser avec une affectation arbitraire  
Tant qu'il existe un chemin améliorant  
faire le ou exclusif de ce chemin avec l'affectation courante

### Chemin améliorant

- ▶ commence et termine par un sommet libre ;
- ▶ alterne arcs libres et choisis ;
- ▶ voit chaque sommet au plus une fois.



# Résolution du problème du rock and roll



## Algorithme de Hopcroft et Karp



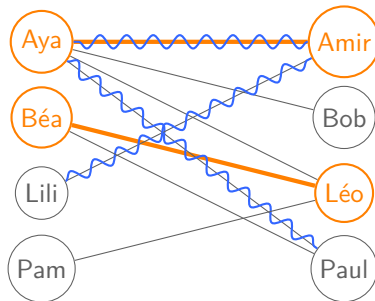
Initialiser avec une affectation arbitraire  
Tant qu'il existe un chemin améliorant  
faire le ou exclusif de ce chemin avec l'affectation courante

### Chemin améliorant

- ▶ commence et termine par un sommet libre ;
- ▶ alterne arcs libres et choisis ;
- ▶ voit chaque sommet au plus une fois.

### Ou exclusif (XOR)

— ou ~  
mais pas ~





# Résolution du problème du rock and roll



## Algorithme de Hopcroft et Karp



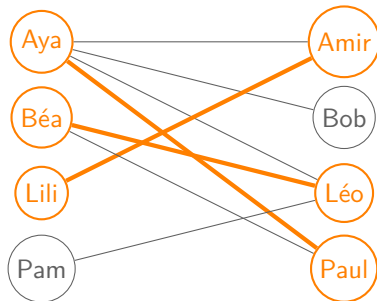
Initialiser avec une affectation arbitraire  
Tant qu'il existe un chemin améliorant  
faire le ou exclusif de ce chemin avec l'affectation courante

### Chemin améliorant

- ▶ commence et termine par un sommet libre ;
- ▶ alterne arcs libres et choisis ;
- ▶ voit chaque sommet au plus une fois.

### Ou exclusif (XOR)

— ou ~  
mais pas ~—



# Résolution du problème du rock and roll



## Algorithme de Hopcroft et Karp



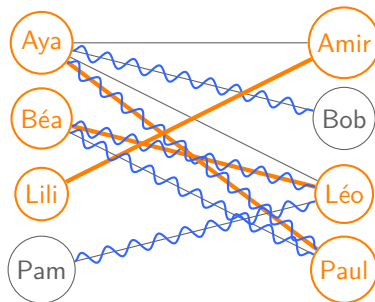
Initialiser avec une affectation arbitraire  
Tant qu'il existe un chemin améliorant  
faire le ou exclusif de ce chemin avec l'affectation courante

### Chemin améliorant

- ▶ commence et termine par un sommet libre ;
- ▶ alterne arcs libres et choisis ;
- ▶ voit chaque sommet au plus une fois.

### Ou exclusif (XOR)

— ou ~  
mais pas ~—



# Résolution du problème du rock and roll



## Algorithme de Hopcroft et Karp



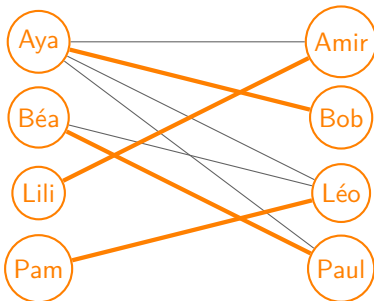
Initialiser avec une affectation arbitraire  
Tant qu'il existe un chemin améliorant  
faire le ou exclusif de ce chemin avec l'affectation courante

### Chemin améliorant

- ▶ commence et termine par un sommet libre ;
- ▶ alterne arcs libres et choisis ;
- ▶ voit chaque sommet au plus une fois.

### Ou exclusif (XOR)

— ou ~  
mais pas ~—



# Résolution du problème du rock and roll



## Algorithme de Hopcroft et Karp



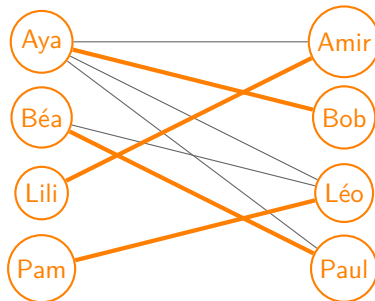
*Initialiser* avec une affectation arbitraire  
*Tant qu'* il existe un chemin améliorant  
*faire* le ou exclusif de ce chemin avec l'affectation courante

### Chemin améliorant

- ▶ commence et termine par un sommet libre ;
- ▶ alterne arcs libres et choisis ;
- ▶ voit chaque sommet au plus une fois.

### Ou exclusif (XOR)

— ou ~~~~~  
mais pas ~~~~~—



L'algorithme de Hopcroft et Karp calcule une solution optimale.

# Complexité du problème du rock and roll

Comment quantifier la **efficacité d'un algorithme** ?

indépendamment de l'ordinateur, du langage de programmation, du compilateur, etc.

# Complexité du problème du rock and roll

## Comment quantifier la **efficacité d'un algorithme** ?

indépendamment de l'ordinateur, du langage de programmation, du compilateur, etc.

**Complexité** : nombre d'**opérations** nécessaires pour résoudre le problème en fonction du nombre de sommets

- ▶ inspection de paires de sommets
- ▶ recherche de chemin augmentant (parcours du graphe)

# Complexité du problème du rock and roll

## Comment quantifier la **efficacité d'un algorithme** ?

indépendamment de l'ordinateur, du langage de programmation, du compilateur, etc.

**Complexité** : nombre d'**opérations** nécessaires pour résoudre le problème en fonction du nombre de sommets

- ▶ inspection de paires de sommets
- ▶ recherche de chemin augmentant (parcours du graphe)

L'algorithme de Hopcroft et Karp nécessite moins de  $n^3$  opérations pour un graphe avec  $n$  sommets ↪ complexité polynomiale

# Complexité du problème du rock and roll

## Comment quantifier la **efficacité d'un algorithme** ?

indépendamment de l'ordinateur, du langage de programmation, du compilateur, etc.

**Complexité** : nombre d'**opérations** nécessaires pour résoudre le problème en fonction du nombre de sommets

- ▶ inspection de paires de sommets
- ▶ recherche de chemin augmentant (parcours du graphe)

L'algorithme de Hopcroft et Karp nécessite moins de  $n^3$  opérations pour un graphe avec  $n$  sommets ↪ complexité polynomiale

**Ce qu'il faut en retenir** : Le problème du rock and roll est simple.



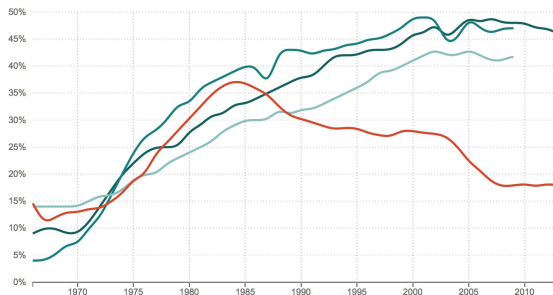
# Rock and roll & Informatique

De façon générale, répartition inégale femmes/hommes en informatique.

## What Happened To Women In Computer Science?

% Of Women Majors, By Field

■ Medical School ■ Law School ■ Physical Sciences ■ Computer science



Source: National Science Foundation, American Bar Association, American Association of Medical Colleges  
Credit: Quoctrung Bui/NPR

années 80 : apparition du PC

confusion entre informatique et nouvelles technologies

# Rock and roll pour tous !

Et si



veut aussi danser avec



?

# Rock and roll pour tous !

Et si

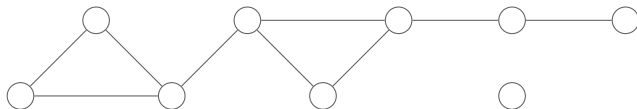


veut aussi danser avec



?

Le graphe est plus riche.



# Rock and roll pour tous !

Et si

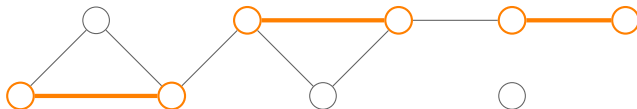


veut aussi danser avec



?

Le graphe est plus riche.



# Rock and roll pour tous !

Et si

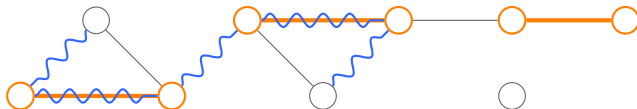


veut aussi danser avec



?

Le graphe est plus riche.



# Rock and roll pour tous !

Et si

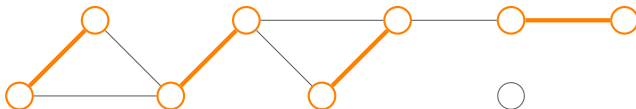


veut aussi danser avec



?

Le graphe est plus riche.



# Rock and roll pour tous !

Et si

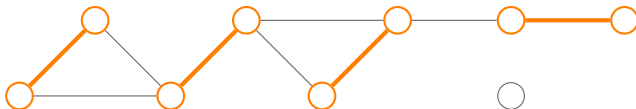


veut aussi danser avec



?

Le graphe est plus riche.



L'algorithme de Hopcroft et Karp fonctionne toujours.

# Rock and roll pour tous !

Et si

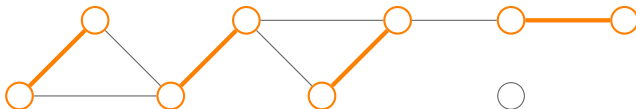


veut aussi danser avec



?

Le graphe est plus riche.



L'algorithme de Hopcroft et Karp fonctionne toujours.



## Algorithme d'Edmonds

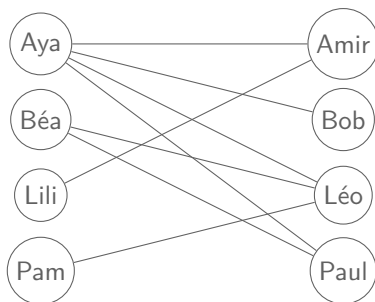
permet de trouver efficacement des chemins améliorants

complexité : toujours polynomiale



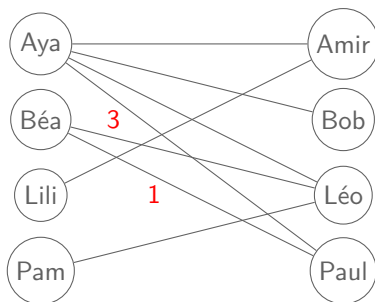
# Niveaux de préférences

Et si Béa préfère Léo à Paul (mais veut bien danser avec les deux) ?



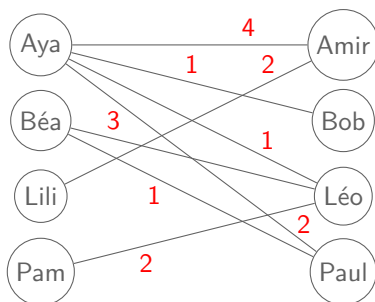
# Niveaux de préférences

Et si Béa préfère Léo à Paul (mais veut bien danser avec les deux) ?



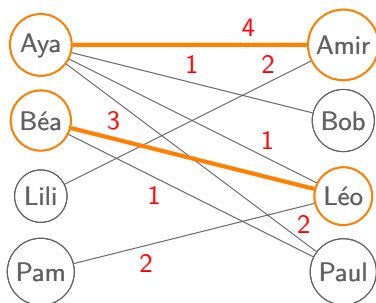
# Niveaux de préférences

Et si Béa préfère Léo à Paul (mais veut bien danser avec les deux) ?



# Niveaux de préférences

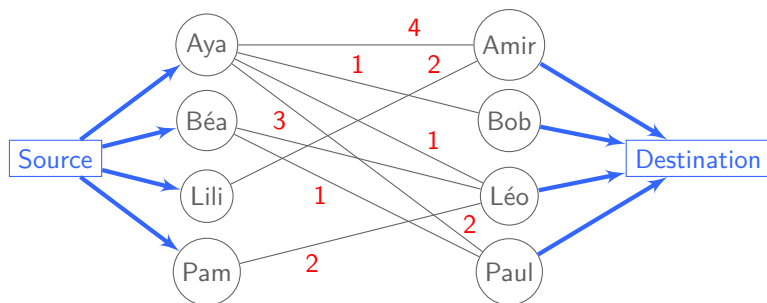
Et si Béa préfère Léo à Paul (mais veut bien danser avec les deux) ?



**Nouvel objectif** : maximiser la somme totale des préférences

# Niveaux de préférences

Et si Béa préfère Léo à Paul (mais veut bien danser avec les deux) ?

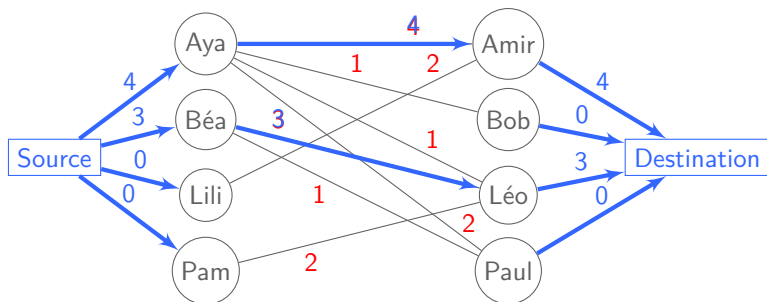


**Nouvel objectif** : maximiser la somme totale des préférences

↪ un problème de **débit d'eau** dans les tuyauteries

# Niveaux de préférences

Et si Béa préfère Léo à Paul (mais veut bien danser avec les deux) ?

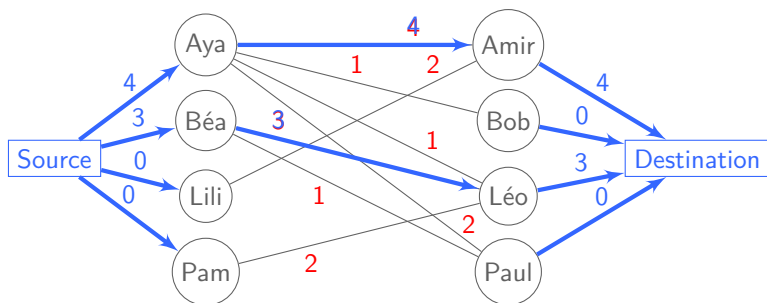


**Nouvel objectif** : maximiser la somme totale des préférences

↪ un problème de **débit d'eau** dans les tuyauteries

# Niveaux de préférences

Et si Béa préfère Léo à Paul (mais veut bien danser avec les deux) ?



**Nouvel objectif** : maximiser la somme totale des préférences

↪ un problème de **débit d'eau** dans les tuyauteries

**complexité** : encore polynomiale

Se généralise aux graphes quelconques (rock and roll pour tous).

# Conclusion

Informatique **théorique** pour résoudre des problèmes **pratiques**



# Conclusion

Informatique **théorique** pour résoudre des problèmes **pratiques**

Quelques mots-clefs pour aller plus loin

- ▶ couplage maximum
- ▶ flot maximal
- ▶ modélisation
  - ▶ théorie des graphes
  - ▶ automates
  - ▶ équations différentielles
  - ▶ etc
- ▶ algorithmique
- ▶ complexité