4TC-IAT, Optimisation combinatoire

Tristan Roussillon

INSA Lyon, TC

2023

Problème d'optimisation combinatoire

$$(\mathrm{P}) \left\{ \begin{array}{c} \min \ f(s) \text{ tel que :} \\ s \in S \text{ et } A(s) \end{array} \right.$$

- S est un ensemble discret fini de solutions,
- $f : S \mapsto \mathbb{R}$ est la fonction objectif,
- $A : S \mapsto \{\text{vrai}, \text{faux}\}$ est le critère d'acceptation d'une solution.

Ensemble des solutions acceptables

L'ensemble des solutions acceptables n'est généralement pas décrit par une liste exhaustive, car

- il peut-être difficile ou long de décider si une solution est acceptable ou non,
- et la liste pourrait être si longue qu'elle soit impossible à compléter en un temps raisonnable.

Elle est plus souvent décrite implicitement par des propriétés.

De nombreux problèmes se ramènent à un problème d'optimisation combinatoire

- problème d'affectation (assignment problem)
- problème de coloration (graph coloring)
- problème des surveillants de musée (art gallery problem)
- problème du sac à dos (knapsack problem)
- problème du voyageur de commerce (travelling salesman problem)
- problème de séquençage de tâches (job sequencing)
- ...

Ex. 1 : problème d'affectation

- 4 ouvriers $\{1, 2, 3, 4\}$ et 4 tâches $\{1, 2, 3, 4\}$
- 1 matrice de coût $C := \begin{pmatrix} 8 & 3 & 1 & 5 \\ 11 & 7 & 1 & 6 \\ 7 & 8 & 6 & 8 \\ 11 & 6 & 4 & 9 \end{pmatrix}$
  par ex. affecter l'ouvrier 4 à la tâche 3 coûte $c_{4,3} = 4$
- Trouver l'affectation de coût minimal
  par ex. l'identité a un coût de $8 + 7 + 6 + 9 = 30$.

Ex. 1 : problème d'affectation

$$(\mathrm{P}) \left\{ \begin{array}{c} \min \;\; f(s) \text{ tel que :} \\ s \in S \end{array} \right.$$

- S est l'ensemble des permutations de $(1, 2, 3, 4)$.
- $f : S \mapsto \mathbb{R}$ est définie telle que, $\forall (t_1, t_2, t_3, t_4) \in S$, $f((t_1, t_2, t_3, t_4)) = \sum_{i=1}^{4} c_{k,t_k}$.

C'est un problème facile car

- toutes les solutions sont acceptables,
- il n'y a que $4! = 24$ solutions possibles.

Ex. 1 : problème d'affectation

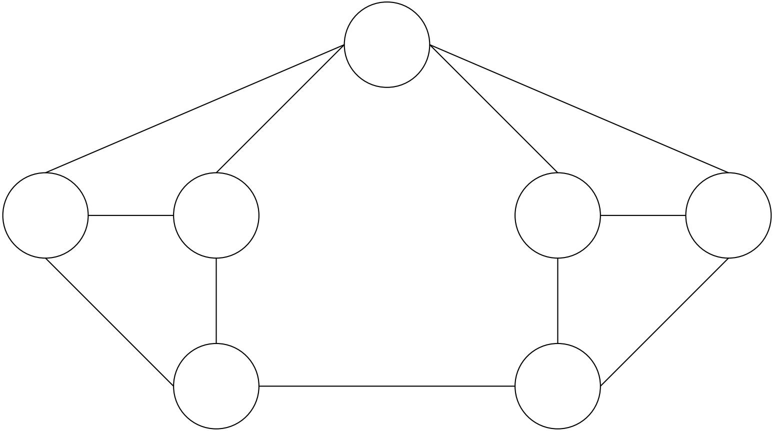
S'il y a n ouvriers et n tâches, il y a $n!$ solutions possibles.

Par ex. si $n = 23$, $n! \approx 5 \cdot 10^{22}$, alors qu'une année ne compte qu'environ $3,16 \cdot 10^{13}$ microsecondes.

Heureusement, il existe des algorithmes plus efficaces que la recherche exhaustive :

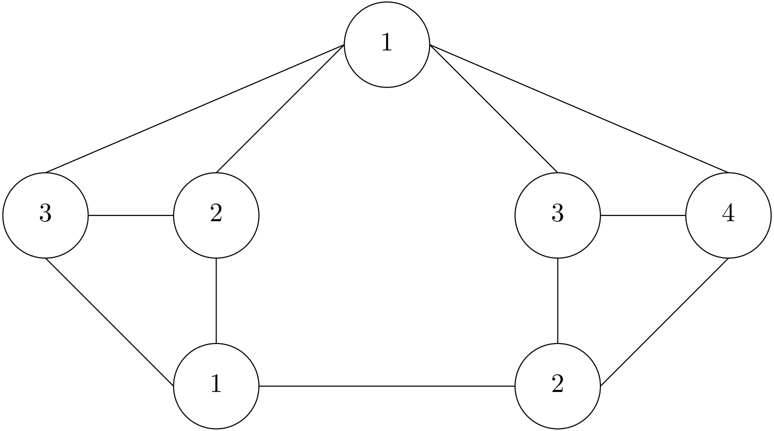
- algorithme hongrois en $O(n^3)$,
- algorithme des enchères,
- push-relabel, preflow-push, ...

Ex. 2 : coloration de graphe



- Soit le graphe ci-dessus (graphe de Moser).
- Trouver la coloration avec le plus petit nombre de couleurs.

Ex. 2 : coloration de graphe



- Soit le graphe ci-dessus (graphe de Moser).
- Trouver la coloration avec le plus petit nombre de couleurs.

Ex. 2 : coloration de graphe

$$(\mathrm{P}) \left\{ \begin{array}{c} \min \ f(s) \text{ tel que :} \\ s \in S \text{ et } A(s) \end{array} \right.$$

- S est l'ensemble des colorations du graphe,
- $\forall s \in S$, $A(s)$ si pour toute arête, les couleurs des noeuds liés par l'arête sont différentes,
- $f : S \mapsto \mathbb{R}$ est définie telle que, $\forall s \in S$, $f(s)$ est le nombre de couleurs présentes dans s.

Il est très difficile d'énumérer l'ensemble des colorations acceptables.

Complexité du problème de coloration

On note n le nombre de sommets.
Il y a des problèmes associé :

- décision : existe-t-il une coloration en k couleurs ?
- recherche : exhiber une telle coloration.

Le problème de décision est NP-complet pour $k > 2$.

Cas d'école

Sommaire

Méthodes de résolution

- recherche exhaustive (très-très petits problèmes)
- *Branch and Bound* (séparation-évaluation)
- relaxation continue dans certains cas particuliers
- méthode des coupes (intégrales, mixtes)
- *Branch and Bound* + coupes = *Branch and cut*
- programmation par contraintes
- méta-heuristiques :
  glouton, meilleur voisin, tabou, recuit-simulé, algorithme génétique, colonie de fourmis, ...

Sommaire

Différentes classes de problèmes et d'algorithmes

$$(\text{P}) \begin{cases} \min\ f(x) \text{ tel que :} \\ g_i(x) \leq 0,\ \ i \in I = \{1, \ldots, m\} \\ x \in S \subseteq \mathbb{R}^n \end{cases}$$

- Optimisation continue ($S = \mathbb{R}^n$)
  - sans contrainte ($I = \emptyset$)
    descente de gradient
  - avec contraintes ($I \neq \emptyset$)
    - non-linéaires
    - linéaires
      simplexe, points intérieurs
- Optimisation combinatoire ($S = \mathbb{Z}^n$)
  branch and bound, (meta)-heuristiques

Pour aller plus loin

- Théorie de la dualité de Lagrange
- Livre de référence sur les aspects théoriques de l'optimisation

Michel Minoux, *Programmation mathématique, Théorie et algorithmes*, Lavoisier, 2eme édition, 2008, 711 pp.