

## SIO - Simulation et optimisation

## Heuristiques et approximation

13 octobre 2025

## Table des matières

<b>1</b>	<b>Heuristiques</b>	<b>1</b>
1.1	Optimisation combinatoire	2
1.1.1	Type de problèmes	2
1.1.2	Exemples	2
1.2	Algorithmes de résolution	2
1.2.1	Algorithmes exacts	2
1.2.2	Algorithmes approximatifs	2
1.2.3	Algorithmes heuristiques	2
1.2.3.1	Heuristiques constructives	3
1.2.3.2	Heuristiques d'amélioration	3
<b>2</b>	<b>Heuristiques de coloration</b>	<b>3</b>
2.1	Coloration d'un graphe	4
<b>3</b>	<b>Heuristiques pour le TSP</b>	<b>4</b>
<b>4</b>	<b>Heuristiques d'échanges</b>	<b>5</b>
<b>5</b>	<b>Métaheuristiques</b>	<b>6</b>

# 1 Heuristiques

## 1.1 Optimisation combinatoire

L'optimisation dite combinatoire est caractérisée par un ensemble de solutions finies mais souvent très grand. Nous aurons donc les caractéristiques suivantes:

- un ensemble **fini**  $S$  de solutions admissibles ou réalisables
- une fonction objectif  $f : S \rightarrow \mathbb{R}$  associant une valeur à chaque solution

### i Info

On cherchera donc souvent à résoudre le problème d'optimisation suivant:

$$\min_{x \in S} f(x)$$

### 1.1.1 Type de problèmes

- problèmes **faciles** (polynomiaux): on peut trouver une solution optimale en temps polynomial
- problèmes **difficiles** (NP-difficiles): on ne connaît pas d'algorithme polynomial pour trouver une solution optimale

### 1.1.2 Exemples

- **problème de l'arbre recouvrant de poids minimal** → problème polynomial
- **problème du plus court chemin de  $s$  à  $t$**  → problème polynomial grâce à l'algorithme de Dijkstra ou Bellman-Ford
- **problème du voyageur de commerce (TSP)** → problème NP-difficile
- **problème du stable de cardinal maximum** → problème NP-difficile
- **problème d'optimisation linéaire**
  - si les variables sont continues → problème polynomial
  - si les variables sont entières → problème NP-difficile

## 1.2 Algorithmes de résolution

### 1.2.1 Algorithmes exacts

- un algorithme **exact** fournit toujours une solution optimale
- pour les problèmes faciles, un bon algorithme **exact** a une complexité polynomiale et demande donc un temps de résolution borné par un polynôme en la taille du problème
- pour les problèmes difficiles, un algorithme **exact** a souvent une complexité exponentielle et demande donc un temps de résolution très long

### 1.2.2 Algorithmes approximatifs

- un algorithme **approximatif** fournit une solution réalisable mais pas forcément optimale
- elle assure la qualité minimale de la solution produite
- elle est de complexité raisonnable (polynomiale)

### 1.2.3 Algorithmes heuristiques

- une heuristique fournit une solution sous-optimale en général
- aucune garantie sur la qualité de la solution
- elle est efficace en pratique (temps de calcul raisonnable)
- on constate empiriquement qu'elle fournit souvent de bonnes solutions

### Hint

Une bonne heuristique doit posséder les qualités suivantes:

- complexité raisonnable et relativement simple à implémenter
- produire de bonnes solutions en pratique proche de l'optimal tout en minimisant les cas de mauvaises solutions

**1.2.3.1 Heuristiques constructives**

Les heuristiques constructives construisent une solution réalisable pas à pas en ajoutant des éléments à une solution partielle jusqu'à obtenir une solution complète. Ces méthodes ne disposent pas d'une solution complète du problème mais parfois d'une solution partielle.

**1.2.3.2 Heuristiques d'amélioration**

Les heuristiques d'amélioration ou dite d'échange, partent d'une solution admissible et cherchent à l'améliorer en explorant son voisinage. Ces modifications, souvent appelés échanges, sont enchaînées tant que des améliorations sont possibles.

## 2 Heuristiques de coloration

### 2.1 Coloration d'un graphe

Soit  $G = (V, E)$  un graphe non orienté. Une **coloration** de  $G$  est une affectation de couleurs aux sommets de  $G$  telle que deux sommets adjacents n'ont pas la même couleur. Le but est de minimiser le nombre de couleurs utilisées.

### **3 Heuristiques pour le TSP**

## 4 Heuristiques d'échanges

## 5 Métaheuristiques