

# Exploration de la Recherche avec Tabous

## Métaheuristiques

---

Pierre BOURGEY, Paul BOUTET, Florian GIURGIU

9 mai 2025

Télécom Saint-Etienne

# Plan

Introduction

Principe Fondamental

Extensions

Avantages - Inconvénients

Applications

# Introduction

---

1. Développement des métaheuristiques dans les années 1980
2. Introduction de la recherche tabou par Fred Glover en 1986
3. Applications dans divers domaines : optimisation, planification, etc.

## Inspiration

Inspirée du comportement de recherche humain.

## Mécanisme

Un "tabou" similaire aux processus cognitifs.

## Stratégie

Évitement des mouvements déjà explorées.

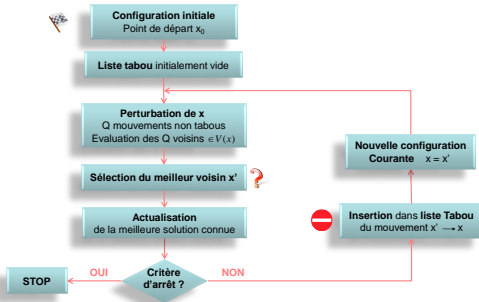
Cette approche s'inspire de la manière dont les humains explorent et évitent les erreurs.

# Principe Fondamental

---

1. **Voisinage** : Ensemble de solutions accessibles par un mouvement.
2. **Mouvement** : Opération qui modifie une solution pour en obtenir une nouvelle. On peut utiliser un vecteur de déplacement, une permutation etc...
3. **Tabou** : Mouvements interdits pour éviter les cycles locaux.
4. **Temps en mémoire courte / taille mémoire** : Durée pendant laquelle un mouvement est considéré tabou (souvent fixé proche de 10 itérations).

# Principe Fondamental



## Exploration

Exploration systématique de l'espace de recherche.

## Mémoire

Mécanisme de "mémoire courte" pour éviter les répétitions.

## Évasion

Capacité de sortir des minima locaux.



# Extensions

---

# Critère d'aspiration

**Définition :** Le critère d'aspiration permet de contourner les mouvements tabous si une solution est suffisamment prometteuse. Cela évite de rester bloqué dans des minima locaux.

## Exemple :

1. Considérons des permutations où les mouvements sont des inversions de deux éléments.
2. Départ : **[1,2,3,4,5]** (coût : 3).
3. Permutation : **[2,1,3,4,5]** (coût : 3), interdit **1**  $\leftrightarrow$  **2**.
4. Permutation : **[2,1,4,3,5]** (coût : 3), interdit **3**  $\leftrightarrow$  **4**.
5. Permutation : **[2,1,4,5,3]** (coût : 3), interdit **3**  $\leftrightarrow$  **5**.
6. Une meilleure solution **[1,2,4,5,3]** (coût : 2) existe, mais le mouvement **1**  $\leftrightarrow$  **2** est interdit.
7. On effectue ce mouvement interdit car il améliore le coût global.

**Définition :** La mémoire à long terme permet de garder une trace des mouvements qui ont été bénéfiques ou nuisibles dans le passé. Cela aide à éviter les mouvements qui ont conduit à mauvaises solutions.

## Exemple :

1. Considérons un problème d'ordonnancement où certaines tâches sont plus difficiles que d'autres.
2. Un mouvement qui a conduit à une solution de mauvaise qualité dans le passé sera évité dans le futur.

## Avantages :

1. Amélioration de la qualité des solutions trouvées.
2. Réduction du temps de calcul en évitant les mouvements nuisibles.

# Pénalisation des mouvements récurrents

**Définition :** La pénalisation des mouvements récurrents consiste à attribuer un coût plus élevé aux mouvements qui ont été effectués plusieurs fois dans le passé. Cela aide à éviter les cycles et à encourager l'exploration de nouvelles solutions.

## Exemple

1. Considérons un problème de voyageur de commerce où certaines villes sont visitées plusieurs fois.
2. Un mouvement qui revient à une ville déjà visitée sera pénalisé.

## Avantages :

1. Encouragement de l'exploration de nouvelles solutions.
2. Évitement des cycles et des solutions sous-optimales.

# Extension des voisinages

**Définition :** L'extension des voisinages consiste à élargir l'ensemble des solutions candidates en considérant des mouvements plus complexes. Cela permet d'explorer de nouvelles régions de l'espace de recherche.

## Exemple

1. Considérons un problème d'optimisation où les mouvements sont des permutations de plusieurs éléments.
2. Un mouvement qui échange plusieurs éléments sera considéré comme un voisinage.

## Avantages :

1. Exploration de nouvelles solutions potentiellement meilleures.
2. Évitement des minima locaux en diversifiant les mouvements.

**Définition :** L'hybridation consiste à combiner la recherche tabou avec d'autres méthodes d'optimisation pour améliorer les performances. Cela permet de tirer parti des forces de chaque méthode.

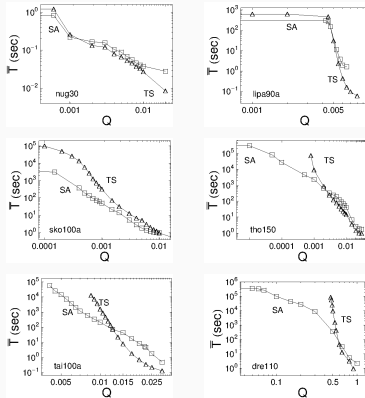
Pour plus d'informations, voir la référence [1].

## Avantages - Inconvénients

---

1. Rapidité d'exécution.
2. Résultats de qualité acceptable.
3. Paramétrage simple (peu de paramètres : taille mémoire, itérations max).





**Figure 1:**  $\bar{T}$  versus  $Q$  for various problem instances for SA (squares) and TS (triangles). For plots which achieve the lowest known cost for an instance ( $Q = 0$ ), we extend the line connecting the plot points to the left edge of the panel.

Mesures de performance :

$$Q = \frac{C - C_{best}}{C_{best}}$$

$C_{best}$  = meilleur coût connu

$C$  = coût de la solution courante

$\bar{T}$  = temps moyen pour atteindre une qualité  $Q$ .

1. Complexité de définition des mouvements et voisinages.
2. Sensibilité aux paramètres (ex : taille de la mémoire, durée de la recherche).

# Applications

---

# Problème d'Affectation Quadratique (QAP)

## Problématique centrale

Affecter  $n$  objets à  $n$  emplacements en minimisant :

$$\sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{p(i)p(j)}$$

où :

1.  $f_{ij}$  : Flux entre l'objet  $i$  et  $j$  (non symétrique)
2.  $d_{rs}$  : Distance entre l'emplacement  $r$  et  $s$  (symétrique)
3.  $p(i)$  : Permutation donnant l'emplacement de l'objet  $i$

## Défi algorithmique

1. NP-difficile : Pas de solution exacte pour  $n > 20$
2. Espace de solutions :  $n!$  permutations possibles
3. Coût de calcul :  $\mathcal{O}(n^2)$  par évaluation

# Problème d'Affectation Quadratique (QAP)

## Applications réelles

- |  |   |
|--|---|
| 1. Planification d'usine                 | 1. Répartition de fichiers              |
| 2. Placement de composants électroniques | 2. Affectation de portes aéroportuaires |
| 3. Optimisation de clavier               | 3. Agencement hospitalier               |

# Exemple concret : Répartition de bâtiments

## Configuration du problème

1. 5 bâtiments à placer sur 5 sites géographiques
2. Objectif : Minimiser les déplacements entre bâtiments
3. Deux matrices clés :

**Matrice des distances  $D$**   
**(symétrique)**

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 1 \\ 1 & 0 & 2 & 1 & 1 \\ 2 & 2 & 0 & 1 & 2 \\ 3 & 1 & 1 & 0 & 1 \\ 1 & 1 & 2 & 1 & 0 \end{bmatrix}$$

Distances en kilomètres

**Matrice des flots  $F$  (non symétrique)**

$$F = \begin{bmatrix} 0 & 8 & 1 & 2 & 3 \\ 5 & 0 & 2 & 1 & 2 \\ 1 & 2 & 0 & 1 & 2 \\ 2 & 1 & 1 & 0 & 6 \\ 3 & 2 & 2 & 7 & 0 \end{bmatrix}$$

Nombre de déplacements/jour

## Configuration initiale

1. Permutation initiale :  $P = (2, 4, 1, 5, 3)$
2. Coût initial : 72
3. Matrice de mémoire  $T$  vide

## Matrice d'interdiction initiale $T$

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

# Itération 1

Mouvement	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
$\Delta$	2	-12	-12	2	0	-10	-12	4	8	6

1. 3 mouvements optimaux :  $\Delta = -12$
2. Choix aléatoire : (1,3)

Mise à jour de  $T$

**Nouvelle solution**

$P = (1, 4, 2, 5, 3)$

Coût : 60

$t = 9$  tiré aléatoirement

$$T = \begin{pmatrix} 0 & 0 & 10 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



## Itération 2

Mouvement	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
$\Delta$	14	X	10	0	10	8	12	12	6	-

1. Mouvement (1,3) interdit ( $t_{13} = 10$ )
2. Meilleur mouvement : (1,4) ( $\Delta = -8$ )

Mise à jour de  $T$

**Nouvelle solution**

$P = (5, 4, 2, 1, 3)$

Coût : 52

$t = 6$  tiré aléatoirement

$$T = \begin{pmatrix} 8 & 0 & 10 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \end{pmatrix}$$

## Itération 3

Mouvement	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
$\Delta$	10	24	10	0	22	20	8	8	14	-

1. Aucun  $\Delta < 0$  (minimum local)
2. Choix du moins mauvais : (2, 3) ( $\Delta = +8$ )

Mise à jour de  $T$

**Nouvelle solution**

$P = (5, 2, 4, 1, 3)$

Coût : 52

$t = 8$  tiré aléatoirement

$$T = \begin{pmatrix} 8 & 0 & 10 & 0 & 0 \\ 10 & 0 & 11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 11 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \end{pmatrix}$$

## Itération 4

Mouvement	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
$\Delta$	24	10	10	8	8	22	20	14	-	-

1. Mouvement (2,4) choisi ( $\Delta = +8$ )

Mise à jour de  $T$

**Nouvelle solution**

$P = (5, 1, 4, 2, 3)$

Coût : 60

$t = 5$  tiré aléatoirement

$$T = \begin{pmatrix} 8 & 0 & 10 & 9 & 0 \\ 10 & 9 & 11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 11 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \end{pmatrix}$$

## Itération 5

Mouvement	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
$\Delta$	12	<b>-10</b>	X	10	X	X	4	14	20	10

1. 1 mouvement optimal : (1, 3) ( $\Delta = -10$ )
2. Mouvements (1, 4), (2, 3), (2, 4) interdits

### Mise à jour de $T$

#### Nouvelle solution



$P = (4, 1, 5, 2, 3)$

Coût : 50

$t = 6$  tiré aléatoirement

$$T = \begin{pmatrix} 8 & 0 & 10 & 9 & 0 \\ 10 & 9 & 11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 11 & 11 & 0 & 0 \\ 11 & 0 & 0 & 8 & 0 \end{pmatrix}$$

1. Interdiction : (4, 3) et (5, 1)  
jusqu'à itération 11

-  Jebari H. - Rahali El Azzouzi S. - Samadi H. (2016). *Hybridation des métaheuristiques pour la résolution de problème d'ordonnancement multi-objectif dans un atelier flow-shop.*
-  Gerald Paul, (2010). *Comparative Performance of Tabu Search and Simulated Annealing Heuristics for the Quadratic Assignment Problem.*