



# Algorithme de recherche avec tabou

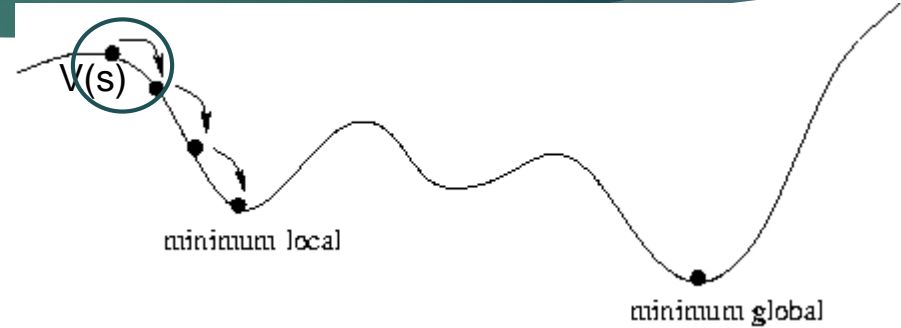
OPTIMISATION COMBINATOIRE

# Plan

1. Introduction
2. Définition de la RT
3. L'algorithme de la RT en générale
4. Critère d'arrêt dans la RT
5. Comportement de l'algorithme tabou
6. Diverses améliorations de la RT
7. Avantages et inconvénients de la RT
8. Domaines concernés par la RT
9. étude de cas PVC
10. Conclusion
11. Bibliographie

# Introduction : La Méthode de descente

A chaque itération, la meilleure solution  $s'$  du Voisinage de  $s$  ( telle que  $f(s') < f(s)$  ) est sélectionnée.



## Carences et solutions

- Taille du voisinage
- Blocage en optima locaux



exploration partielle



autorisation de l'acceptation de mouvements de dégradation



**La recherche tabou est une correction de la méthode de descente mise en place pour sortir des minima locaux**

# La Méthode de recherche tabou : Idée de base.

**Acceptation de solutions moins bonnes que la solution courante**



**Risque de cyclage**



**Liste de tabou : solutions interdites pour un nombre d'itérations**



**Critères d'aspiration : conditions permettant de lever le statut tabou d'une solution**

# Définition de base de la R.T

## Définition :

Développée dans un cadre particulier par Glover en 1986 (et indépendamment par Hansen en 1986) , c'est une méthode heuristique de recherche locale utilisée pour résoudre des problèmes complexes et/ou de très grande taille (souvent NP-durs).

## Principe de base :

- poursuivre la recherche de solutions même lorsqu'un optimum local est rencontré et ce,
  - en permettant des déplacements qui n'améliorent pas la solution
  - en utilisant le principe de mémoire pour éviter les retours en arrière (mouvements cycliques)

# Définition de base de la R.T

## Mémoire :

- elle est représentée par une liste taboue qui contient des mouvements ou des solutions qui sont temporairement interdits

## 2 alternatives :

- Une liste contient les solutions interdites (coûteux en place mémoire).
- Une liste des mouvements interdits (qui ramènent vers ces solutions déjà visitées ).
  - Avantages : prend moins de place mémoire.
  - élimine plus de solutions que celles visitées effectivement

Généralement les listes sont gérées en FIFO (first in first out).

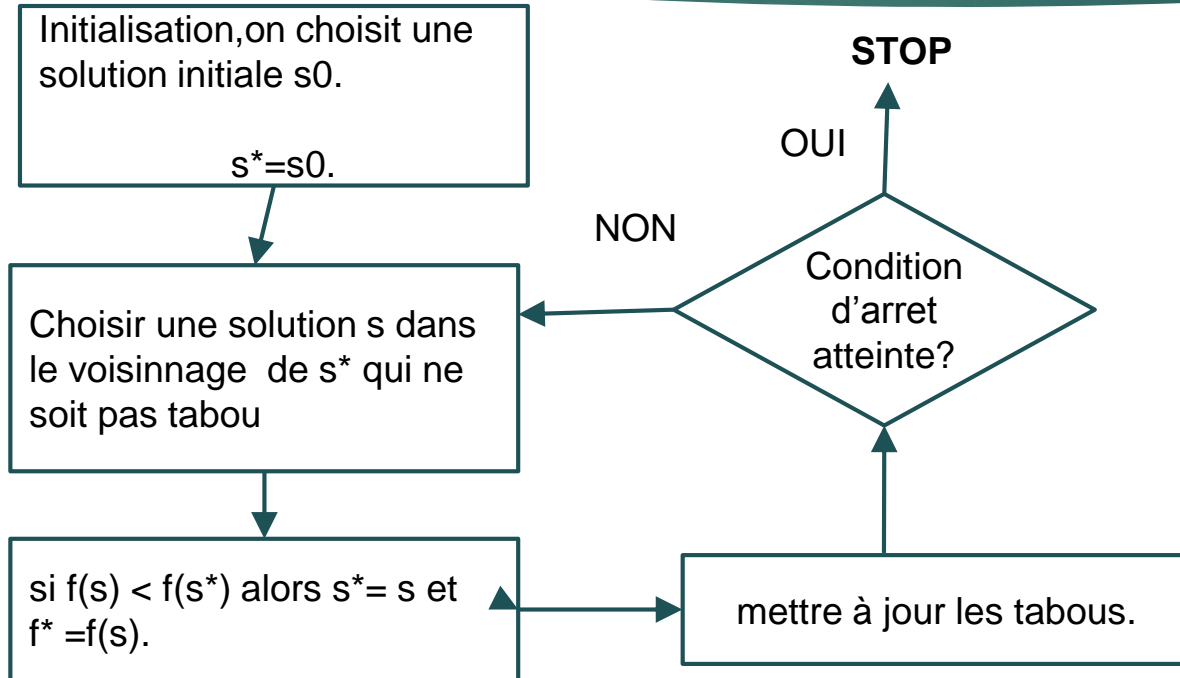
# Définition de base de la R.T

- Liste tabou des mouvements interdits élimine plus de solutions que celles visitées effectivement .
- plus efficace que la liste des solutions taboues mais élimine éventuellement de très bonnes solutions.

Idée:  une fonction d'aspiration qui permet de lever le status tabou.( l'expection au interdiction).

- Il est possible de violer une interdiction lorsqu'un mouvement interdit permet d'obtenir la meilleure solution enregistrée jusqu'à maintenant.

# L'algorithme de la RT en générale



·  $s_0$  : solution initiale

·  $s^*$  : meilleure solution jusqu'à présent.

$s$  : nouvelles solutions du voisinage de  $s^*$

·  $f(x)$  : fonction objectif à minimiser.

$f(s^*)$  : valeur de la meilleure solution



# Critère d'arrêt

- On peut arrêter la recherche à tout moment.
  - Contrairement au recuit simulé...
- Des critères d'arrêt possibles sont :
  - Si une solution prouvée optimale a été trouvée.
  - Si une limite a été atteinte en ce qui concerne
    - Le nombre d'itérations ;
    - Le temps de calcul.
  - Si la recherche semble stagner : nombre d'itérations sans amélioration de la meilleure configuration trouvée.

# Comportement de l'algorithme tabou

- **Si la liste taboue est courte :**

- Il y a moins d'interdictions (mouvements tabous).
- La recherche épouse mieux les optima locaux rencontrés.
- L'algorithme tend à parcourir de moins grandes distances dans l'espace de recherche. Il explore moins l'espace de recherche.
- Le risque de cycles est plus grand.

# Comportement de l'algorithme tabou

- **Si la liste taboue est longue :**
  - Il y a avantage d'interdictions (mouvements tabous).
  - La recherche risque de manquer de nombreux optima locaux sur son chemin.
  - L'algorithme tend à parcourir de plus grandes distances dans l'espace de recherche. Il explore davantage l'espace de recherche.
  - Le risque de cycles est réduit.

# Comportement de l'algorithme tabou

- **Le comportement de l'algorithme dépend :**
  - De la longueur de la liste taboue.
  - la taille de la liste de candidats (si on en utilise une)

Plus la liste de candidats est petite, moins la liste taboue a besoin d'être grande.

# Diverses améliorations de la RT

- La liste taboue peut s'avérer trop contraignante lors de la recherche d'une solution. Le mécanisme d'aspiration permet de lever ponctuellement le statut "tabou" afin d'atteindre des solutions inédites.
- **L'intensification** : est l'une des stratégies qui permet de mémoriser les meilleures solutions rencontrées (ou leur configuration) et les utilise afin d'améliorer la recherche.
  - **La diversification** : cherche à utiliser des mouvements encore jamais réalisés afin d'explorer des régions nouvelles de l'espace de recherche en mémorisant bien sûr les solutions les plus visitées.

# Diverses améliorations de la RT

Sélection du meilleur voisin :

- **Best Fit** : le voisinage est exploré en entier.
- **First Fit** : une partie du voisinage est explorée.
- **Utilisation d'une table de calculs** : pour éviter de calculer entièrement le coût de chaque voisin, à chaque itération on mémorise dans une table les modifications au coût de la solution courante associées à chacun des mouvements possibles.

# Avantages et inconvénients de la RT

## **Avantage :**

- Grande efficacité
- Fonctionnement simple à comprendre.

## **Inconvénients :**

- Paramètres peu intuitifs.
- Demande en ressources importantes si la liste des tabous est trop imposante.
- Aucune démonstration de la convergence.

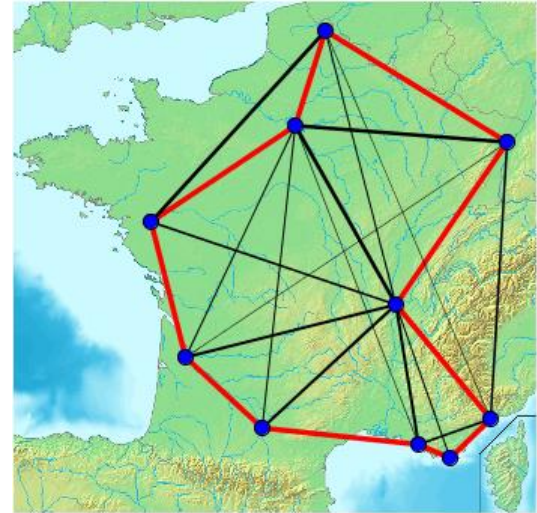
# Domaines concernés par la RT

- Problèmes de transport.
- Planification et ordonnancement.
- Optimisation de graphes.
- Télécommunications.
- Logique et intelligence artificielle.
- Création d'horaires.
- Optimisation de structures.
- Techniques spécialisées.



# Etude de cas : Problème de Voyageur de Commerce

- Un voyageur de commerce doit visiter un certain nombre de villes, et chaque ville une et une seule fois. Etant donné des distances entre chaque paire de villes, il doit minimiser la distance totale parcourue.



# Etude de cas : Problème de Voyageur de Commerce

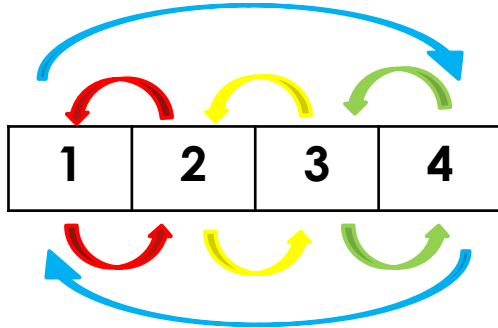
- ▶ On peut représenter ce problème par un graphe : chaque ville correspond à un sommet et chaque arête à une paire de villes pouvant être visitées l'une à la suite de l'autre.
- ▶ Le problème correspond à trouver un tour complet (circuit Hamiltonien) dans ce graphe qui minimise la somme des distances  $f$ .
- ▶ Le nombre de solutions pour  $n$  villes est de  $(n-1)!/2$ .
  - ▶ Si  $n = 4$ , il y a trois solutions possibles.
  - ▶ Si  $n = 30$ , il y en a **4 420 880 996 869 850 977 271 808 000 000 !**

# Instance du problème

- ▶ Une instance du PVC est un graphe complet de  $n$  sommets dont les arêtes sont pondérées par un coût strictement positif.
- ▶ L'instance sera alors implantée comme une matrice  $M$   $n \times n$  dont les coefficients sont strictement positifs sauf sur la diagonale où ils sont tous nuls.
- ▶  $M$  est appelé matrice de coût. Ainsi la distance entre le sommet  $i$  et le sommet  $j$  est  $M_{ij}$ .

# Espace de recherche et voisinage

- L'espace de recherche  $S_n$  est l'ensemble des permutations de  $\{1, 2, \dots, n\}$ . Un point de l'espace de recherche est une permutation
- voisinage  $N(x)$  : pour une solution  $X$ ,  $N(x)$  est l'ensemble des permutation par paire de sommets,



2	1	3	4
---	---	---	---

1	3	2	4
---	---	---	---

1	2	4	3
---	---	---	---

4	2	3	1
---	---	---	---

# Fonction d'évaluation

► Pour une permutation  $x$

$$f(x) = \sum_{i=0}^{n-2} Mx_{(i)}x_{(i+1)} + Mx_{(n)}x_{(0)}$$

c'est cette fonction que nous cherchons à minimiser

# Exemple d'instance du PVC

Exemple de 10 villes avec une distance entre 0 et 100

	0	1	2	3	4	5	6	7	8	9
0	0	12	26	11	47	23	46	44	52	35
1	12	0	44	20	49	43	90	92	27	91
2	26	44	0	50	35	21	11	82	1	92
3	11	20	50	0	5	57	69	62	74	91
4	47	49	35	5	0	61	89	26	28	80
5	23	43	21	57	61	0	11	8	73	36
6	46	90	11	69	89	11	0	16	55	61
7	44	92	82	62	26	8	16	0	83	71
8	52	27	1	74	28	73	55	83	0	74
9	35	91	92	91	80	36	61	71	74	0

# Résultat de la recherche Tabou

## ► Configuration :

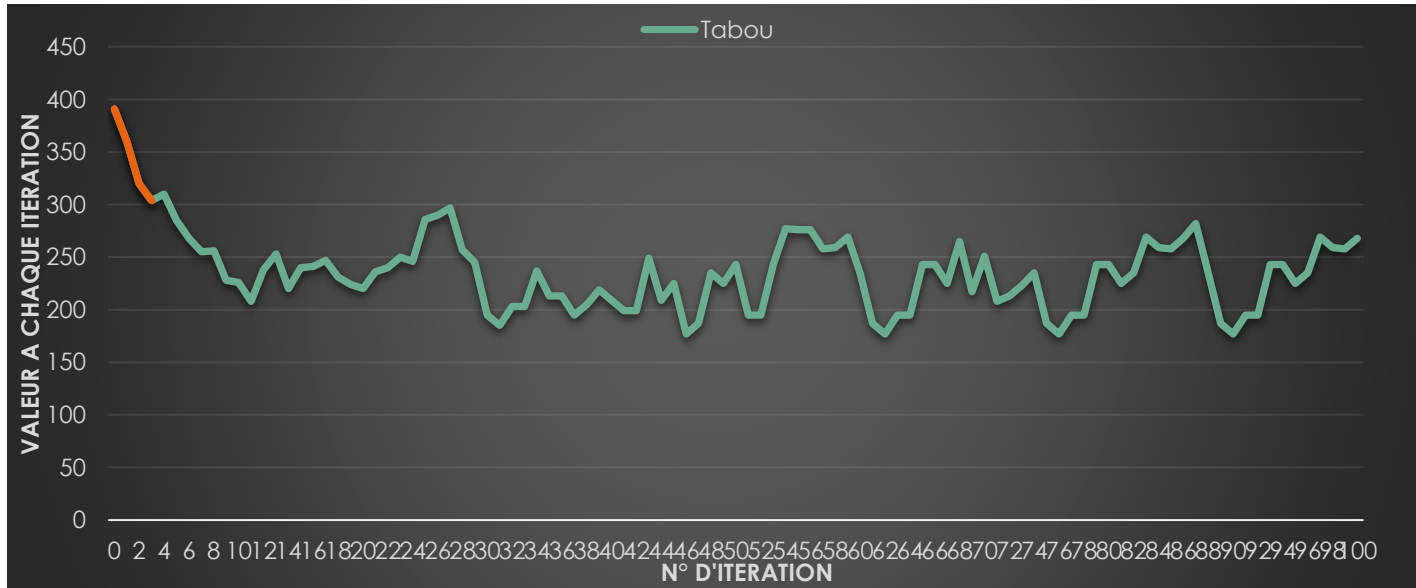
- Liste tabou de taille 10
- Critère d'arrêt : nombre d'itération (100)

Solution initial :  $s = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$ , valeur = 391

Solution Descente :  $s = [0, 2, 1, 3, 4, 7, 5, 6, 8, 9]$ , valeur = 304

Solution Tabou :  $s = [9, 0, 3, 1, 8, 4, 7, 5, 6, 2]$ , valeur = 177

# Résultat de la recherche Tabou





# Bibliographie

- [1] Article Lapetoule de Joseph Ayas & Marc André Viau « la recherche tabou »
- [2] Livre de Kévin « Les algorithmes métaheuristiques » édition Juin 2006