

Présentation Projet OPT

Hybridation des métaheuristiques pour le
problème du 'Bin Packing'

Groupe 1 / Equipe 2

Table de matière

01

Introduction

Bin Packing

02

Problématique

Pourquoi hybrider?

03

État de l'art

- Métaheuristiques
- Hybridation
- Métaheuristiques hybrides

04

Solution proposée

Schéma d'hybridation
Composants

05

Étude expérimentale

- Environnement et benchmarks
- Etude empirique des paramètres
- Etude des performances de la solution
- Etude comparative

06

Synthèse & Conclusion

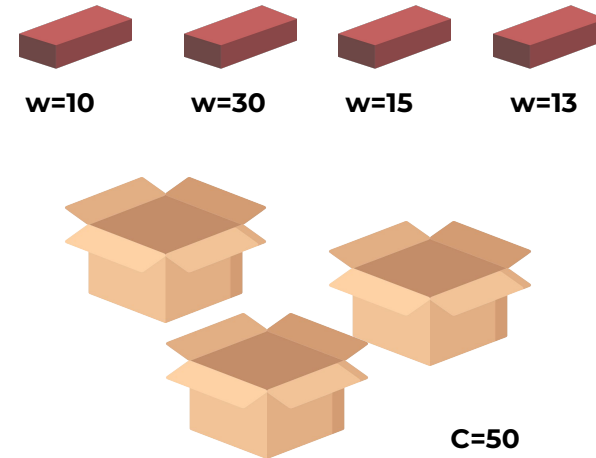
01

Introduction

Problème du Bin Packing

Un problème d'optimisation classique où un ensemble d'articles de différentes tailles doit être emballé dans un nombre minimum de conteneurs de taille fixe appelés bacs (bins).

- Introduit par Keller en 1969.
- Problème NP-Difficile.
- 3 formes du problème: 1D, 2D, 3D.
- Problème de minimisation.



Contexte

Pratiquement: utilisé dans de nombreuses situations pratiques de la vie réelle:

- Emballage de valises et optimisation de stockage.
- Optimisation des ressources à l'aide de l'allocation efficace des ressources.

Théoriquement : Nous pouvons formuler et coder le problème "Bin Packing" comme suit:

$$\left\{ \begin{array}{l} \text{Ensemble des objets : } I = \{1, 2, \dots, n\} \\ \text{Ensemble des bacs : } B = \{1, 2, \dots, m\} \\ \text{Poids de l'objet } i : w_i, \forall i \in I \\ \text{Capacité maximale de chaque bac : } Cy_j \end{array} \right. \left| \begin{array}{l} \left\{ \begin{array}{l} x_{ij} \in 0, 1 \quad \forall i \in I, \forall j \in B \\ y_j \in 0, 1 \quad \forall j \in B \end{array} \right. \end{array} \right. \left| \begin{array}{l} \text{Fonction objectif : Minimiser } \sum y_j, \text{ avec } j \in B \\ \text{Contraintes : } \left\{ \begin{array}{l} \sum_{j \in B} x_{ij} = 1 \\ \sum_{i \in I} w_i x_{ij} \leq Cy_j \quad \forall j \in B \end{array} \right. \end{array} \right.$$

02

Problématique

Pourquoi aller vers l'hybridation des métaheuristiques pour le problème d'optimisation combinatoire **“Bin Packing”** ?

Pourquoi aller vers l'hybridation ?

- Un problème **NP-difficile**.
- Les approches exactes sont inefficaces pour de grandes instances.
- Les Heuristiques offrent des solutions acceptables mais non optimales.
- Les métaheuristiques offrent des techniques efficaces mais limitées.
- **Solution:** L'hybridation des métaheuristiques est une approche qui **vise à combiner les avantages de différentes métaheuristiques afin d'améliorer les performances** de résolution du problème.





03

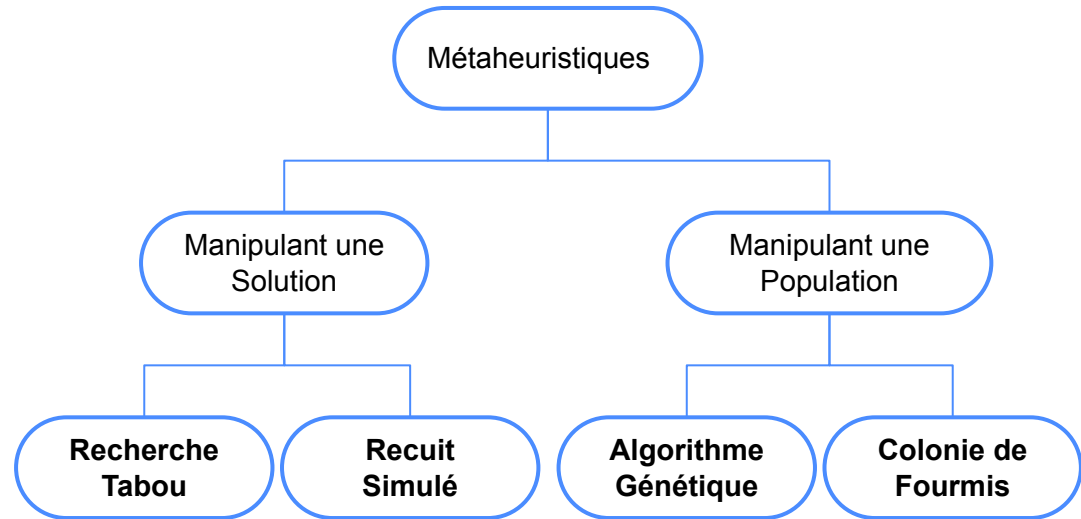
Etat de l'art

Métaheuristiques pour le Bin Packing

- Les paramètres pour adapter les **AG au bin packing** incluses :
la représentation binaire des solutions, opérateurs de croisement et de mutation spécifiques, taille population, taux de croisement et de mutation.
- Les paramètres pour adapter la **RT au bin packing** incluses :
la taille et la structure de la liste tabou, les critères de voisinage et les mouvements interdits, les stratégies d'intensification et de diversification, ainsi que les critères de sortie.

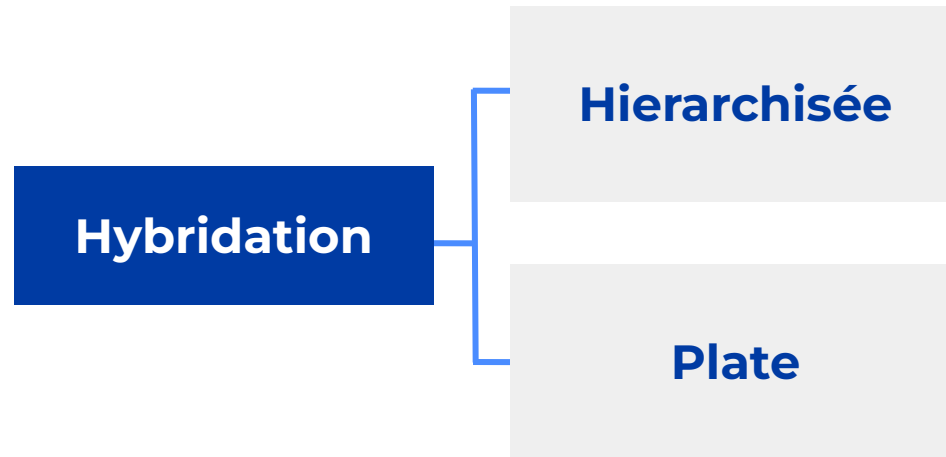
Métaheuristiques pour le Bin Packing

Les métaheuristiques abordées au niveau du cours sont toutes bien adaptées à ce problème.



Taxonomie d'Hybridation -Talbi-

Dans notre cas, on a choisi la taxonomie d'hybridation proposée par Talbi qui propose une classification des méthodes hybrides à deux niveaux.



Métaheuristiques Hybrides

AG avec RS

L'AG pour explorer l'espace de recherche et le Recuit simulé pour affiner les solutions.

ACO avec RT

L'ACO pour guider la recherche, et la recherche tabou pour intensifier la recherche.

PSO avec AG

L'AG pour l'initialisation, la reproduction et l'évolution de la population, le PSO est utilisé pour explorer l'espace de solution.

RT avec RS

La recherche tabou pour exploiter les régions prometteuses et échapper aux optima locaux, et le recuit simulé est utilisé pour explorer l'espace de recherche.

Métaheuristiques Hybrides

AG avec les CNN

Les NC sont utilisés pour extraire les caractéristiques et faire des prédictions. L'AG pour faire évoluer une population de modèles CNN.

Quantum Computing avec AG

Le quantum computing est utilisé pour améliorer certains aspects de l'AG, tels que la représentation des solutions, l'exploration de l'espace de recherche et la manipulation des solutions candidates.

'New Metaheuristics'

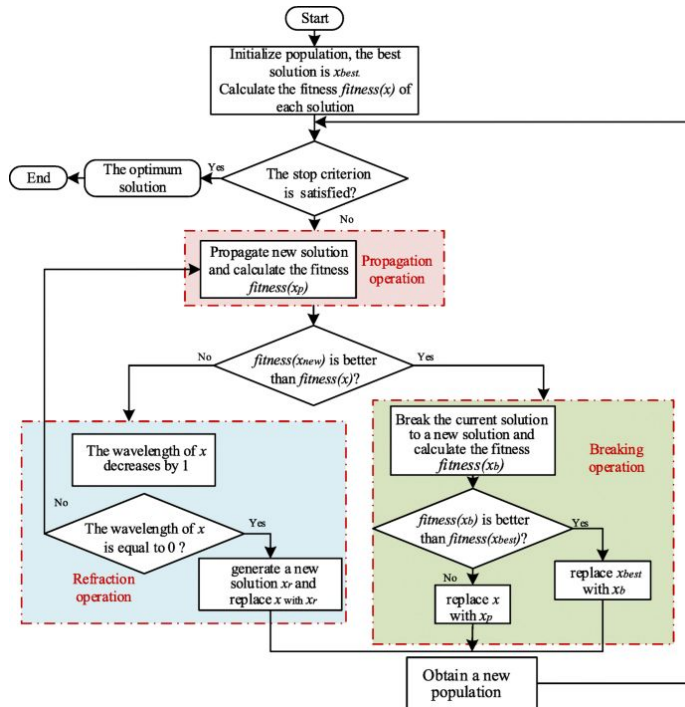
N°	Métaheuristique	Avantages	Inconvénients
1	Moth-Flame Optimization (MFO)	Facile à paralléliser. Simple à comprendre et à mettre en œuvre.	Sensible au choix des paramètres.
2	Prey-Predator Algorithm (PPA)	Optimiseur global. Robuste au bruit. Mise en œuvre simple.	Coûteux en termes de calcul. Sensible à la population initiale. Difficile de trouver des paramètres optimaux.
3	Water Wave optimization (WWO)	Peu de paramètres à régler (bon choix pour les praticiens). Peut être utilisé pour les problèmes de minimisation et de maximisation.	Sensible à la population initiale. Lent à converger pour certains problèmes.
4	The Whale Optimization Algorithm (WOA)	Convergence rapide.	Convergence vers les optima locaux. Évolutivité limitée.
5	Shark Smell Optimization (SSO)	Facile à paralléliser. Mise en œuvre simple.	Coûteux en calcul pour les grands problèmes.
6	World Cup Competitions (WCO)	Trouver de bonnes solutions. Mise en œuvre simple.	Coûteux en calcul, en particulier pour les problèmes importants.
7	Dragonfly Algorithm (DA)	Polyvalence.	Convergence prématurée. Évolutivité limitée. Manque de fondements théoriques.
8	Grey Wolf Optimizer (GWO)	Efficace en temps de calcul. Simple à comprendre et à mettre en œuvre.	Sensible aux conditions initiales. Peut être piégé dans des optimums locaux. Coûteux en calcul pour les problèmes à grande échelle.

04

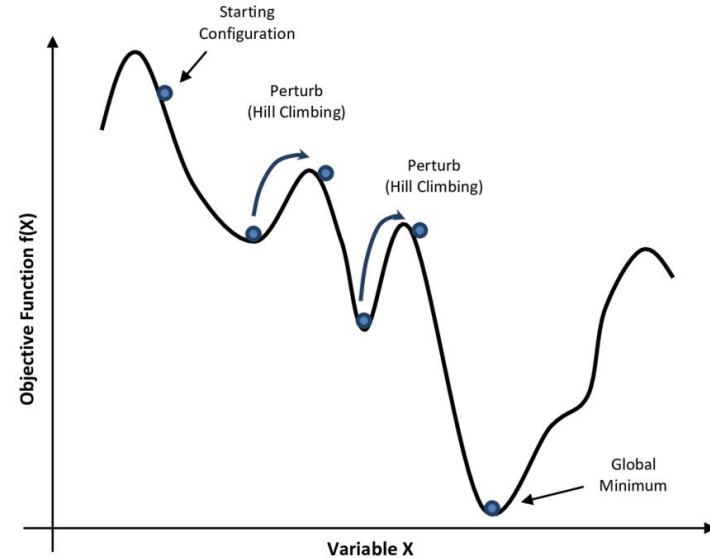
Solution proposée

Algorithmes d'optimisation

Water wave optimizer



Recuit simulé



Caractéristique nécessaire

Water wave optimizer

- Propagation de la vague (recherche locale)

$$x'(d) = x(d) + \text{rand}(-1, 1) \cdot \lambda L(d)$$

- Refraction de la vague.

$$x'(d) = N\left(\frac{x^*(d) + x(d)}{2}, \frac{|x^*(d) - x(d)|}{2}\right)$$

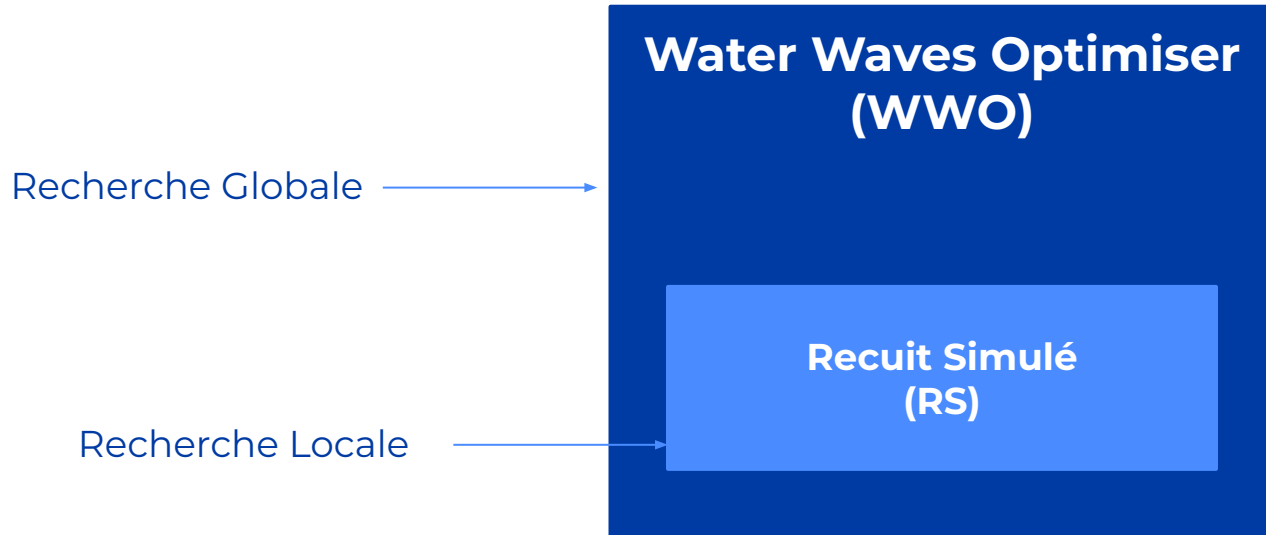
- Break de la vague.

$$x'(d) = x(d) + N(0, 1) \cdot \beta L(d)$$

Recuit simulé

- Recherche intensive (voisinage).
- Diversification dans certains cas (équation de la chaleur).
- Converge vers la solution optimale si nombre d'itération tend vers + l'infini.

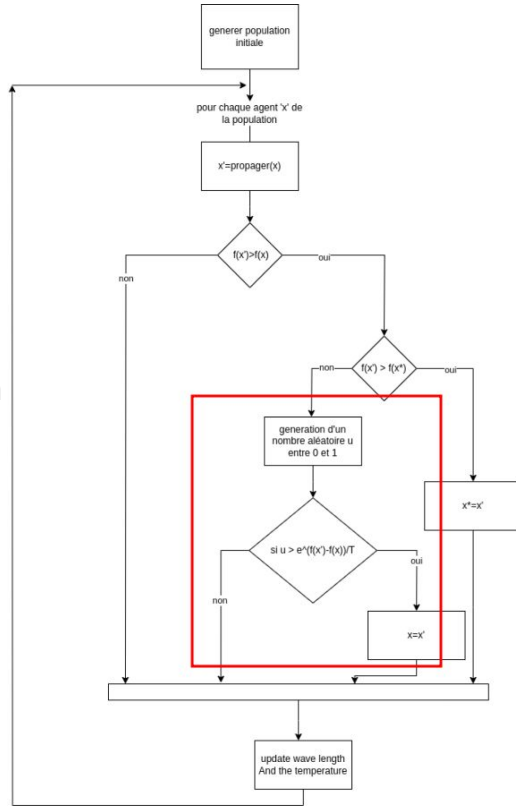
Schéma d'hybridation



Notation: LRH (WWO(RS))(HET, P, S)

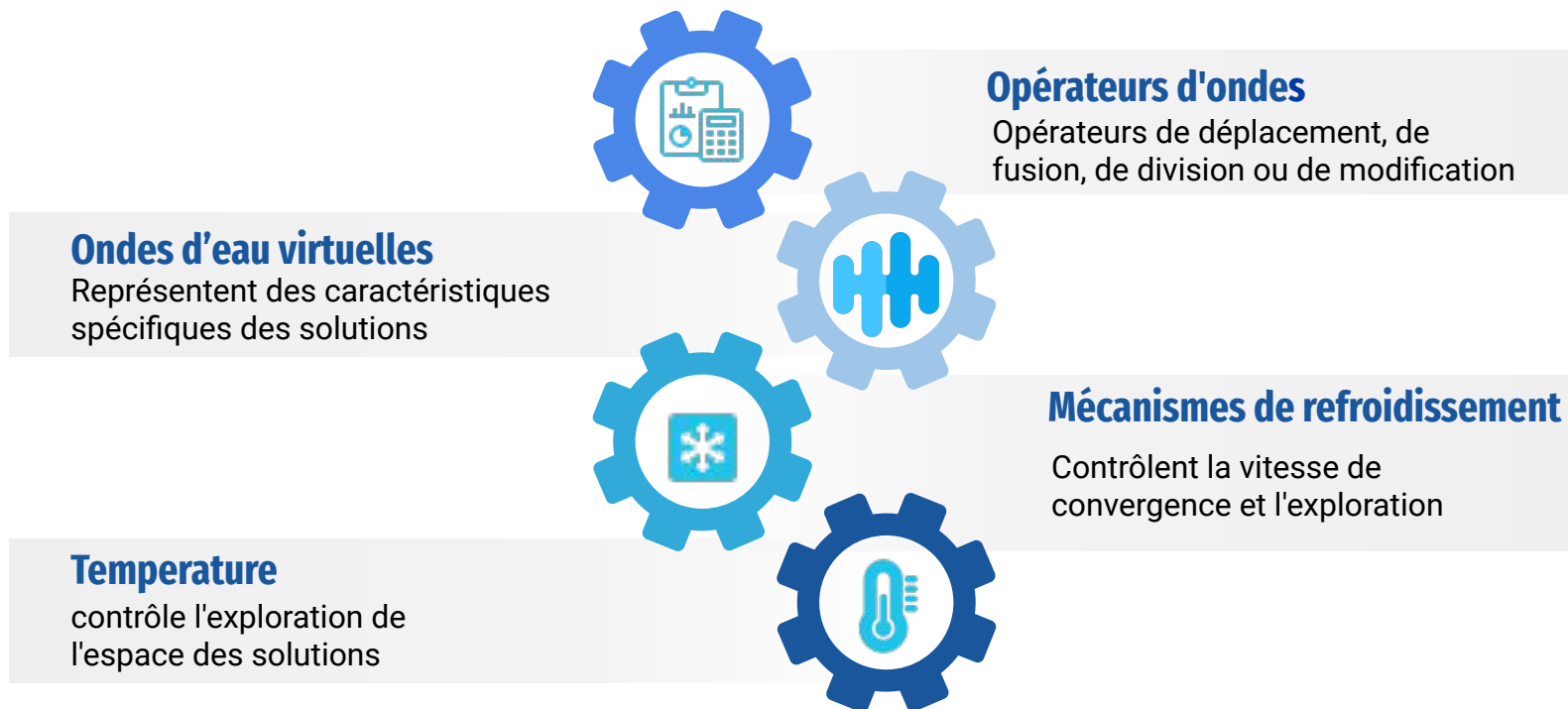
Schéma de l'hybridation

Algorithme d'hybridation



- Remplacement du break wave par une affectation directe donc prendre immédiatement la solution qui améliore la solution optimale
- Remplacement le phénomène de la réfraction par l'équation de la chaleur utilisé dans le recuit simulé

Composants



Justification du choix

1. Exploration et exploitation améliorées

- Le WWO est connu pour son efficacité dans l'**exploration de l'espace** de recherche, tandis que le recuit simulé est efficace pour l'**exploitation des zones locales** ainsi que sa **tolérance de choisir des solutions qui n'améliorent pas l'état actuel pour échapper au optimum locaux** [c'est un algorithme qui converge vers la solution optimal].
- En combinant ces deux métaheuristiques, on a pu bénéficier de leur capacité combinée à rechercher des solutions prometteuses dans différentes parties de l'espace de recherche.

2. Diversification de la recherche

- Le WWO a tendance à générer **une grande diversité de solutions**, tandis que le recuit simulé peut aider à **éviter de rester bloqué** dans des optima locaux sous-optimaux.
- En combinant ces deux approches, vous pouvez favoriser une recherche plus diversifiée, augmentant ainsi les chances de trouver de meilleures solutions.

Explication visuelle

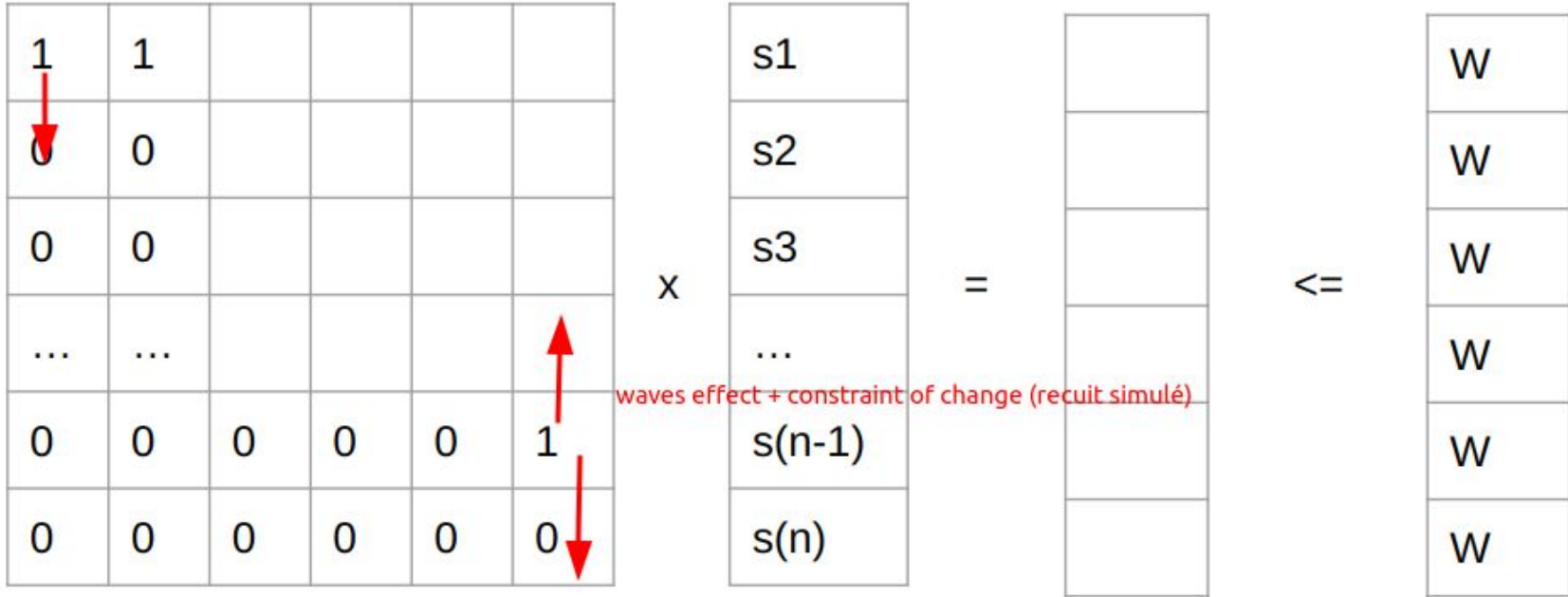
matrice :


n colonnes: les objets

n lignes: nombre
de bins

1	1				
0	0				
0	0				
...	...				
0	0	0	0	0	1
0	0	0	0	0	0

Explication visuelle





05

Etude Expérimentale

Environnement

Environnement et Langage de programmation



Python

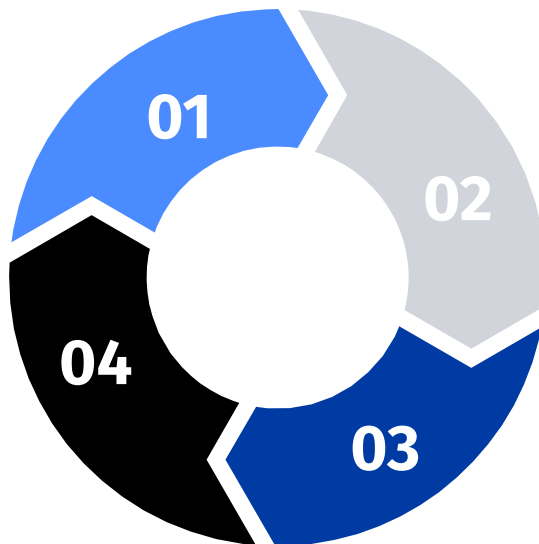


VS code

Système d'exploitation



Windows



Bibliothèques ou frameworks



NumPy



Pandas

Matériel



RAM : 16GO

CPU : i7 10eme generation

Benchmarks

Benchmarks	Nombre d'objets	Capacité du conteneur
BPP_50_50_0.1_0.7_6	50	50
BPP_50_150_0.1_0.8_8	50	150
BPP_750_200_0.1_0.7_0	750	200

Etude Empirique des Paramètres

Paramètres WWO (Water wave optimizer) :

- Instance du type de problème
 - Nombre d'objets
 - Liste d'objets
 - Capacité du conteneur (Bin)
- La largeur maximale d'onde H_{max}
- Le coefficient de réduction de la longueur d'onde α
- Le coefficient de rupture β

Paramètres SA (Simulated Annealing) :

- Température initiale T_0 .
- Le coefficient de réduction de la température α .
- La fonction de réduction de la température (géométrique).

Choix des paramètres

Paramètre	choix
Largeur maximale d'onde	0,5
Coefficient de réduction de la longueur d'onde α	1.01
Largeur maximale d'onde	0,5
Coefficient de réduction de la longueur d'onde α	1.01
Coefficient de rupture β	1.01

Etude des performances de la solution

Meilleure exploitation des régions locales de l'espace des solutions

Exploitation

Amélioration de l'exploration de l'espace des solutions

Exploration

Un délai plus raisonnable pour rendre l'approche plus pratique

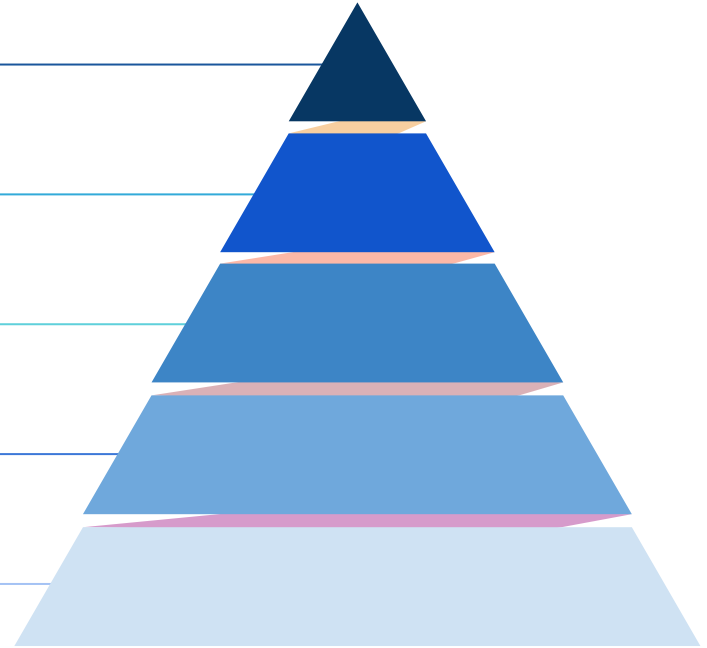
Temps

Minimisation de la fonction de coût

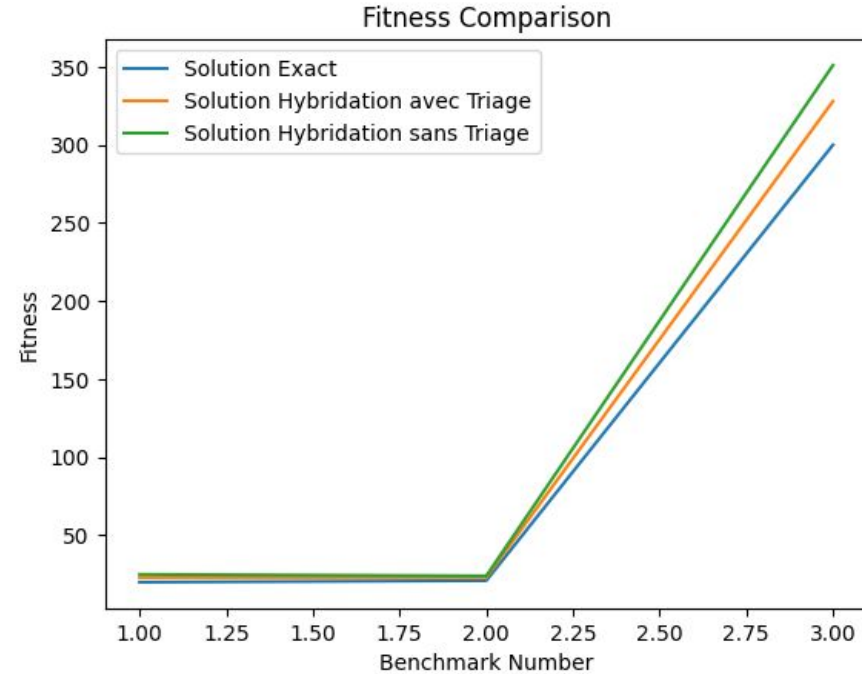
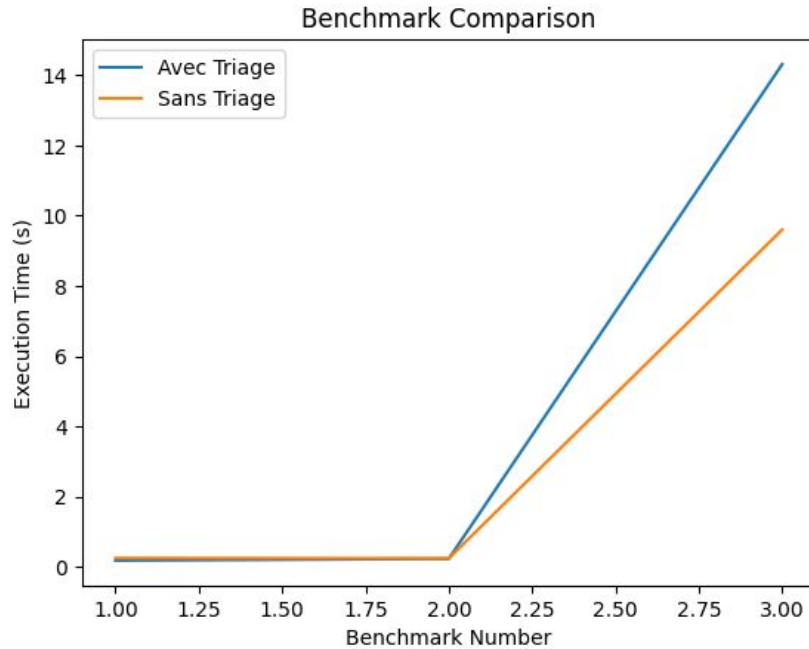
Coût

Accélération de la convergence des algorithmes d'optimisation

Convergence



Etude comparative



06

Synthèse et conclusion



Synthèse et conclusion

“ Le WWO peut aider à explorer efficacement l'espace de recherche et générer une diversité de solutions, tandis que le recuit simulé peut apporter des améliorations locales et aider à éviter les optima locaux. En combinant ces deux méthodes, on a pu profiter de leurs forces respectives et améliorer la performance globale de la recherche. “



Merci