



Fusion de polyèdres pour les jeux vidéo et logiciels 3D

FONTAINE Antoine
DUBOIS Jérôme
LEPAGNOT

Projet M1 IM
M.



Sommaire

1. Introduction
2. Modélisation de polyèdres
3. Test de convexité
4. Algorithme de fusion
5. Graphe des fusions convexes
6. Algorithme brute-force
7. Méta-heuristiques
 - Principe et évaluation
 - Recuit simulé
 - Algo génétique
8. Points d'améliorations
9. Conclusion



Introduction

- Polyèdre: solide dont les faces sont des polygones.

Problématique :

- Un nombre élevé d'objets 3D implique divers problèmes de performances :
 - Occupation de mémoire
 - Vitesse d'affichage



Source : diapositives du sujet



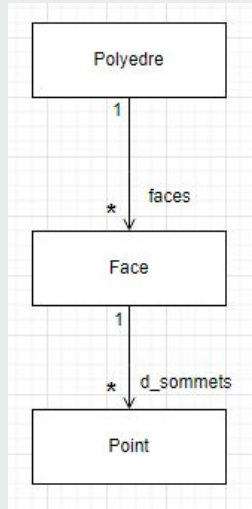
Objectif :

- Développer un algorithme de fusion de polyèdres sur des polyèdres convexes élémentaires
- Rechercher et implémenter des métaheuristiques pour résoudre le problème



Création de polyèdres

Implémentation des polyèdres



Polyèdre :

- Liste de faces
- Est convexe ou non

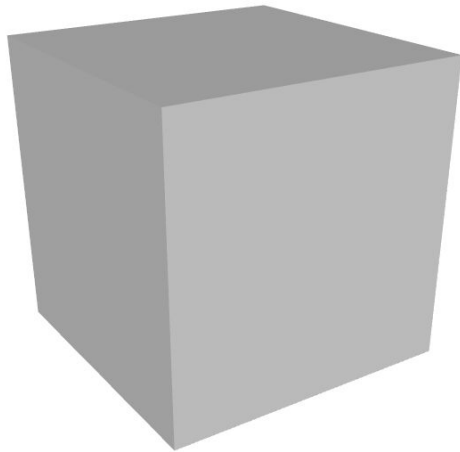
Face :

- Liste ordonnée de points

Point :

- Coordonnées X, Y, Z

Fichier d'objet 3D *.obj



- Description d'une géométrie 3D
- Syntaxe clé :
 - Sommet : "v <x> <y> <z>"
 - Objet : "o <nom_obj>"
 - Face : "f <v1> <v2> <v3>"

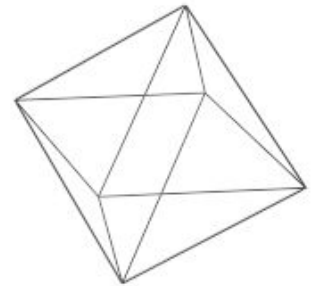
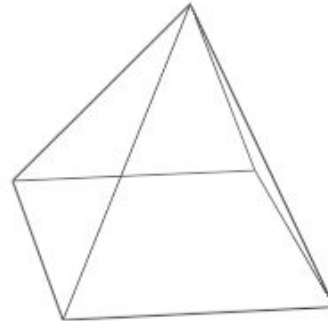
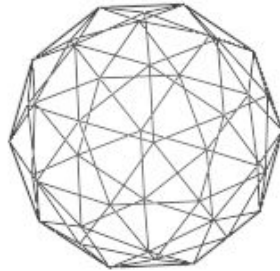
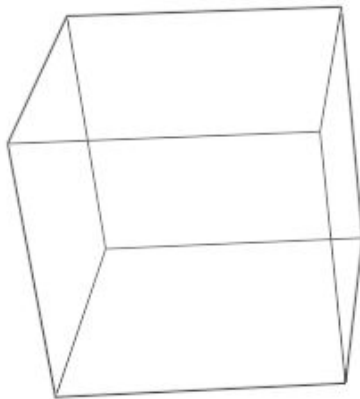
```
1      v 1 1 -1
2      v 1 -1 -1
3      v 1 1 1
4      v 1 -1 1
5      v -1 1 -1
6      v -1 -1 -1
7      v -1 1 1
8      v -1 -1 1
9      o Object1
10     f 1 5 7 3
11     f 4 3 7 8
12     f 8 7 5 6
13     f 6 2 4 8
14     f 2 1 3 4
15     f 6 5 1 2
```



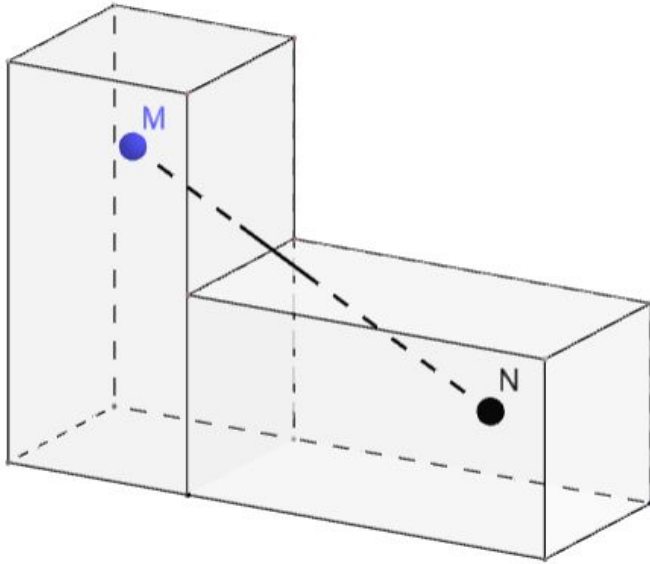

Tests de convexité

Polyèdres convexes

Définition : Un polyèdre est convexe si, **pour toute paire de points situés à l'intérieur du polyèdre** (ou sur une des faces), le **segment reliant ces deux points est entièrement contenu à l'intérieur** ou sur la surface du polyèdre.

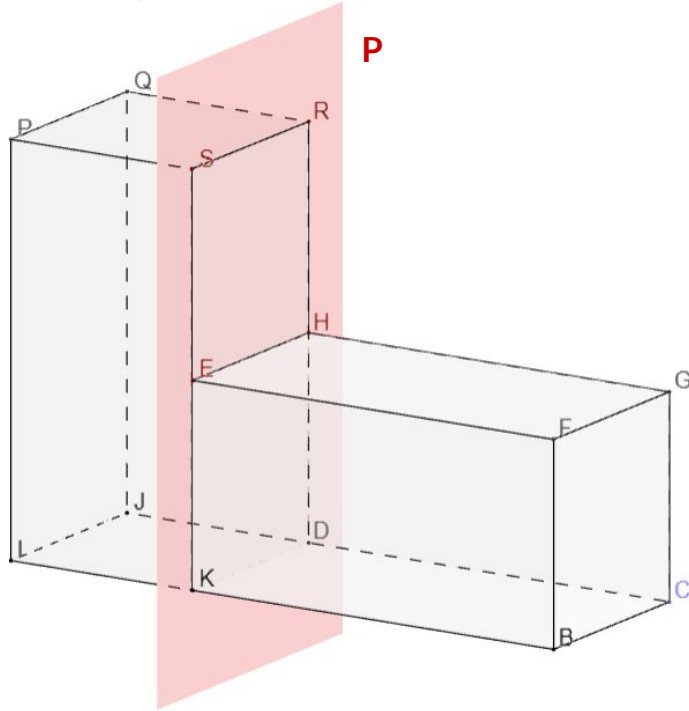


Polyèdre non convexe



- Soit M et N deux points contenus dans le polyèdre
- Le segment $[MN]$ n'est pas contenu dans le polyèdre

Polyèdre non convexe (en pratique)

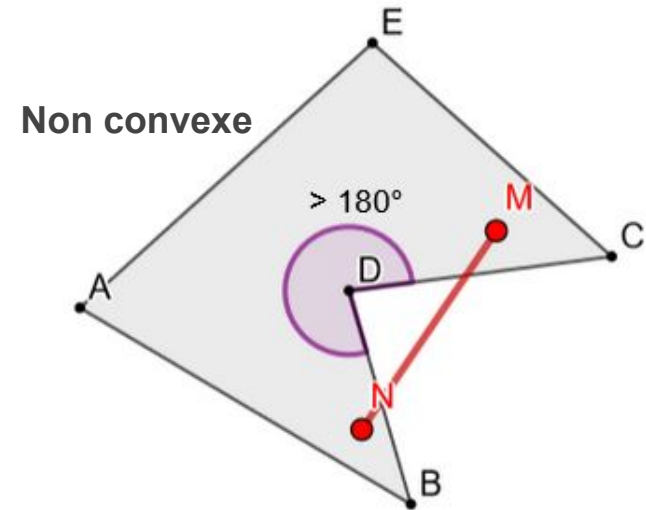
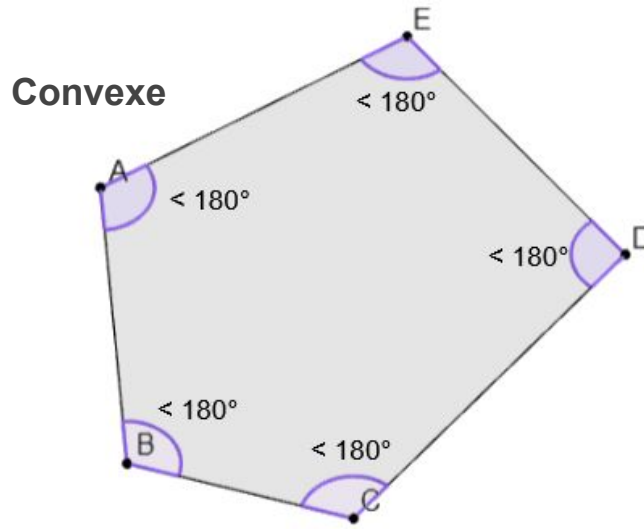


Trouver au moins une face qui vérifie :

- Plan **P** qui prolonge la face **SRHE**
- Les points **S, R, H, E, D, K** sont **sur le plan**
- Les points **P, Q, L, J** sont à **gauche** du plan
- Les points **F, G, B, C** sont à **droite** du plan
- **Au moins deux points ne sont pas du même côté**
⇒ Polyèdre non convexe

Convexité de polygones

- Même principe que pour les polyèdres
- Tous les angles intérieurs sont inférieurs à 180°





Algorithme de fusion



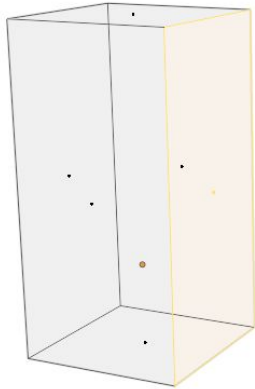
Fusion de deux polyèdres convexes

Critères pour faire une fusion convexe :

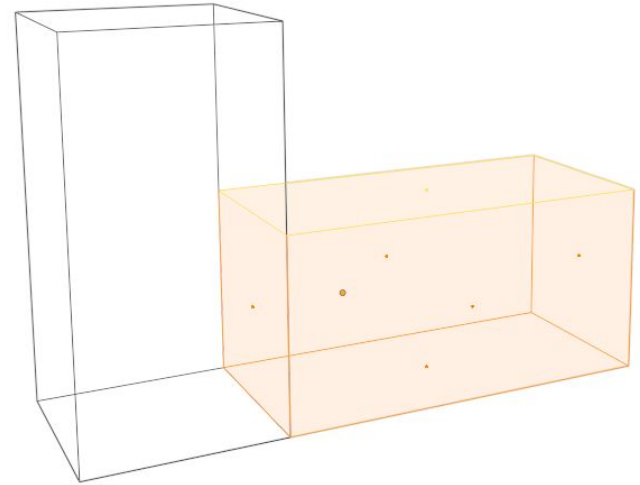
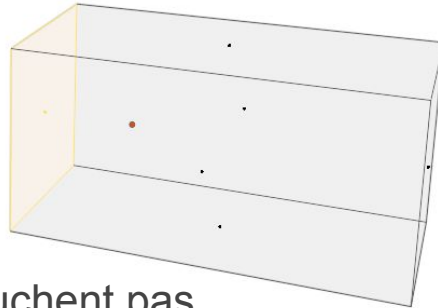
- Les deux polyèdres sont **convexes**
- Ils ont **au moins une face commune**,
c'est-à-dire une face avec les mêmes sommets
- La **fusion** est **convexe**

Fusion de deux polyèdres (exemples)

- Pas de faces communes



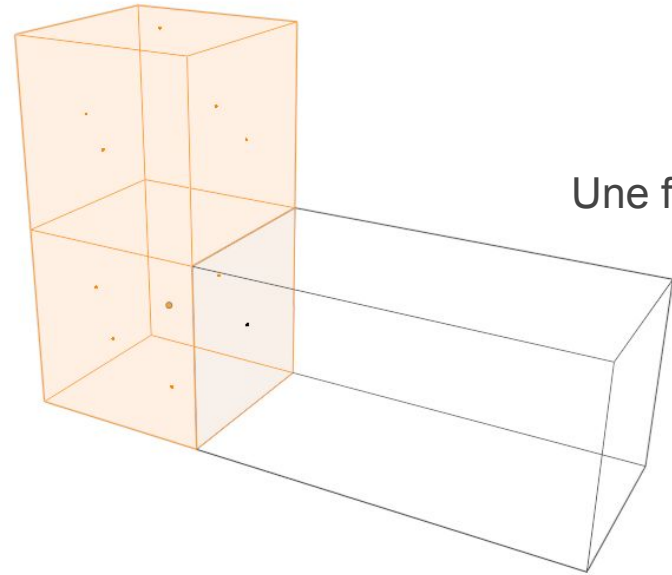
Les faces ne se touchent pas



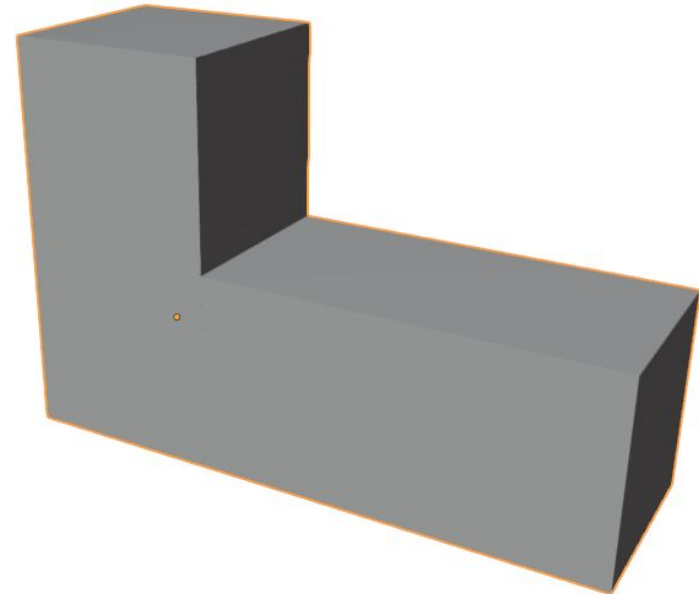
Deux faces se touchent, mais
elles ne partagent pas les
mêmes sommets

Fusion de deux polyèdres (exemples)

- Au moins une face commune, mais fusion pas convexe

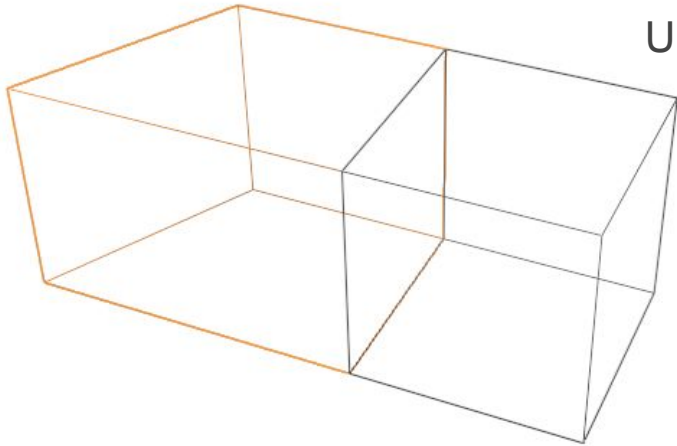


Une face commune \Rightarrow Fusion

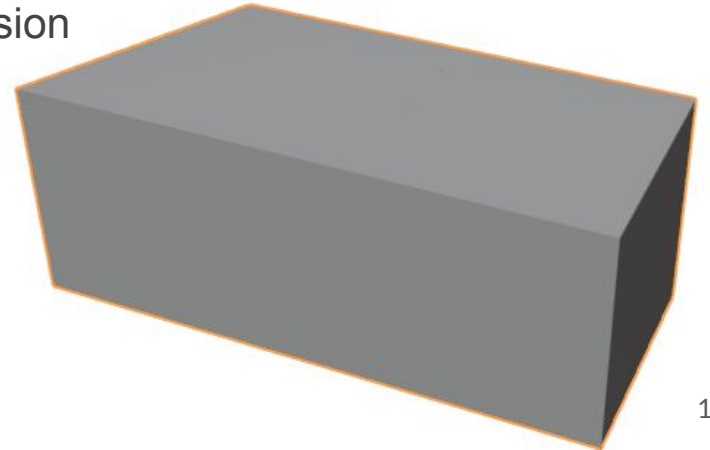


Fusion de deux polyèdres (exemples)

- Au moins une face commune, fusion convexe

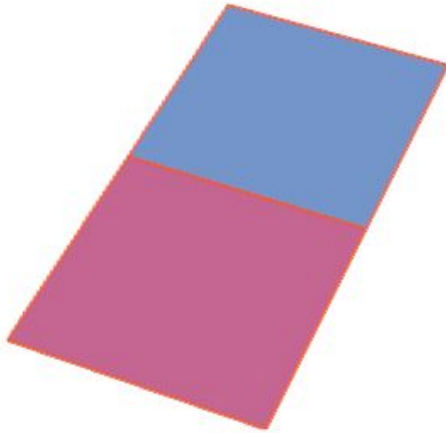


Une face commune \Rightarrow Fusion

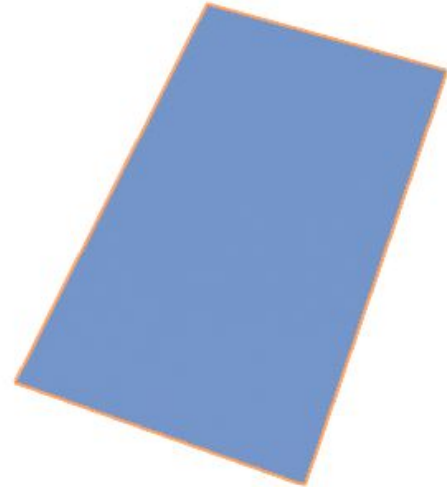


Fusion de deux polyèdres (cas particulier)

- Polyèdres avec un volume nul (une seule face)
 - Exemple avec des polygones convexes



Une arête commune \Rightarrow Fusion





Algorithme de fusion d'un ensemble de polyèdres

Entrée :

- Ensemble de polyèdres **ordonnés** (ou **solution**)

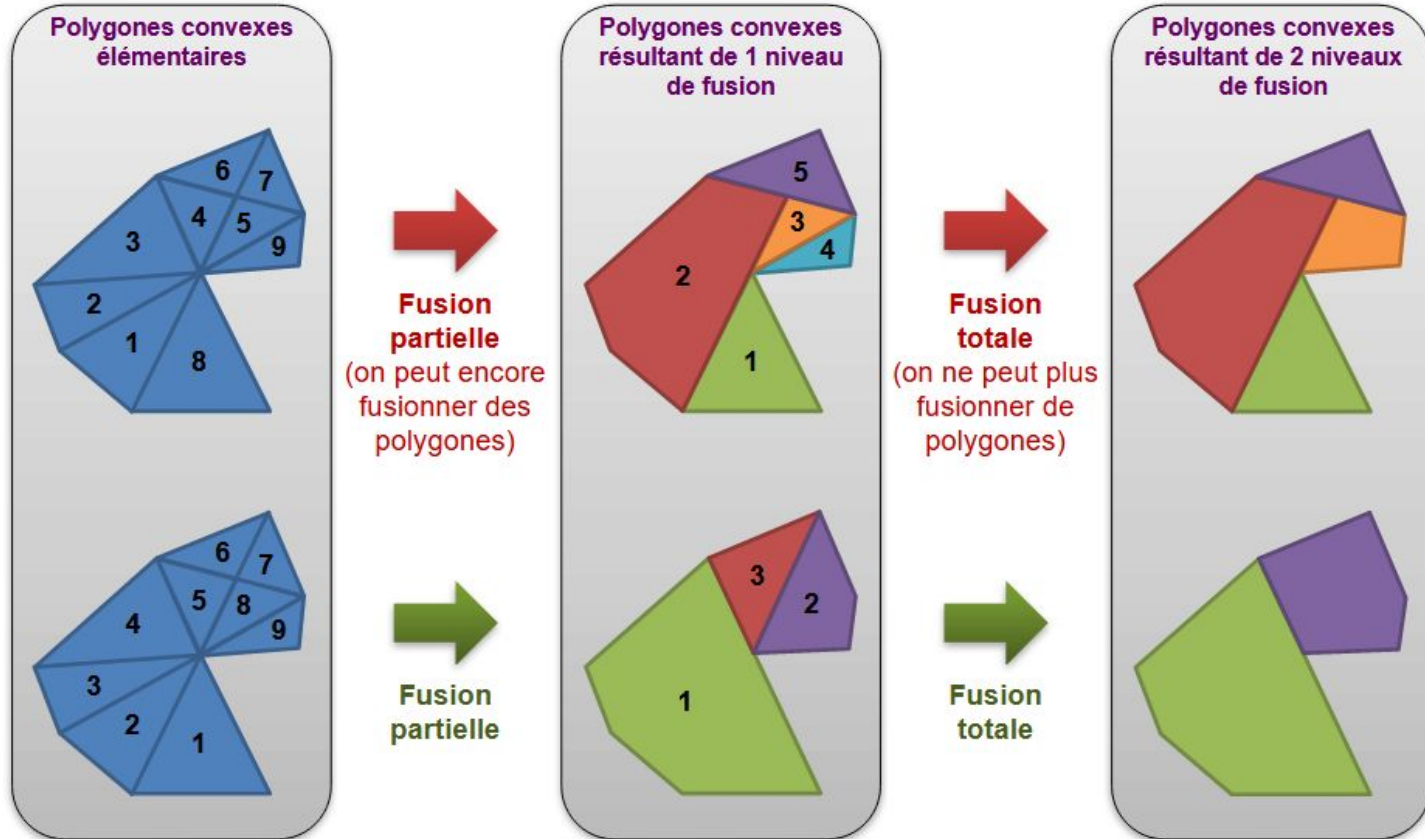
Principe :

- Essayer de fusionner les polyèdres deux à deux en parcourant la liste dans l'ordre
- Recommencer tant qu'on peut faire de nouvelles fusions

Sortie :

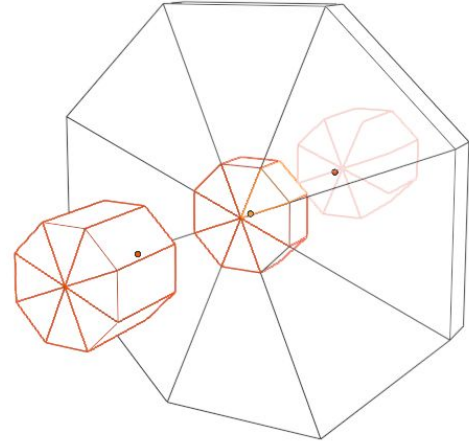
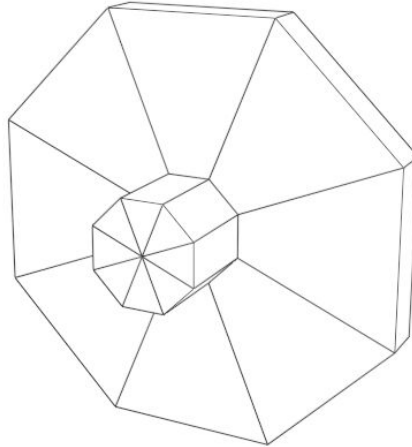
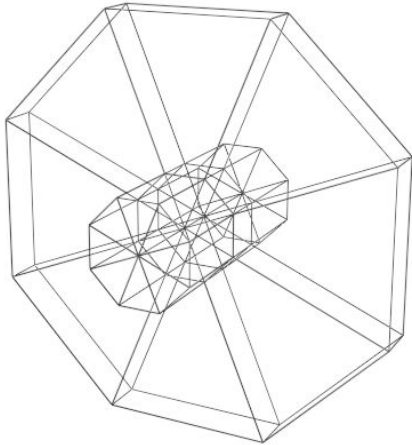
- Liste de polyèdres **après les fusions**

Algorithme de fusion d'un ensemble de polyèdres



Fusion : exemple avec une “roue”

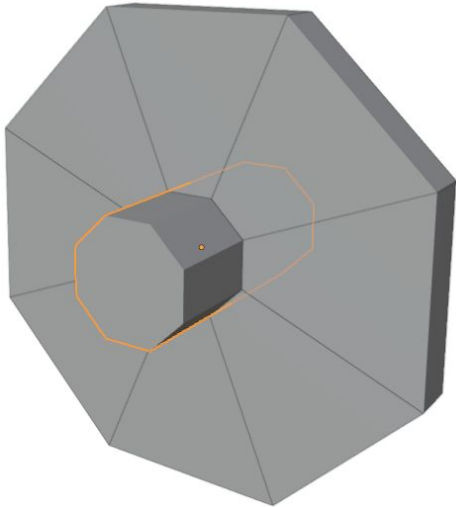
- Comment obtenir un nombre minimal de polyèdres convexes ?
- 32 polyèdres



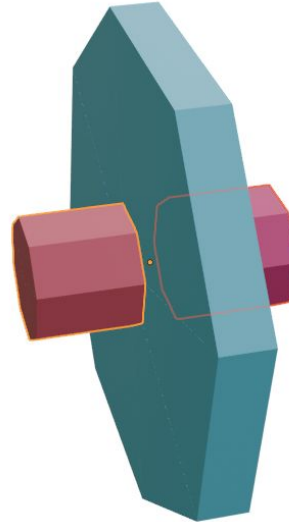
Fusion : exemple avec une “roue”

- L'ordre de fusion donne des résultats différents

9 polyèdres



Résultat optimal : 3 polyèdres





Graphe des fusions convexes



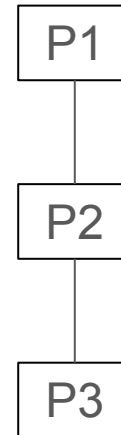
Graphe des fusions convexes

- Non orienté
- Sommets = Polyèdres
- Arête = Fusion convexe

Exemple :

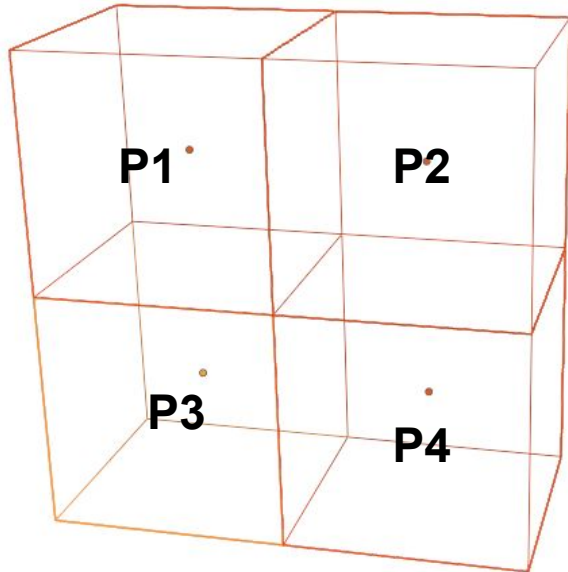
- La fusion de **P1** et **P2** est convexe
- La fusion de **P1** et **P2** est convexe
- La fusion de **P1** et **P2** n'est pas convexe

Exemple d'un graphe



Graphe des fusions convexes (exemple)

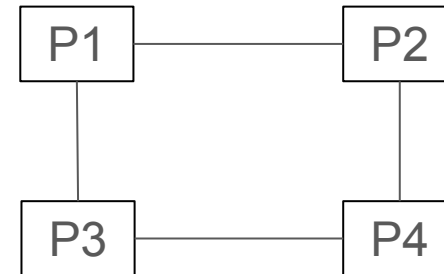
- 4 polyèdres convexes (cubes)



Liste initiale de polyèdres :

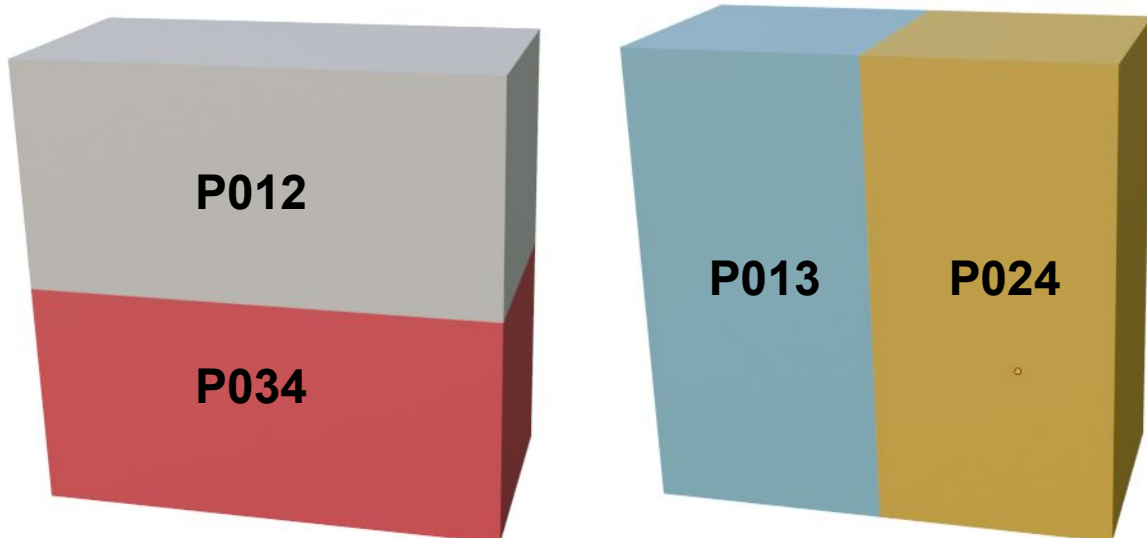
- P1 - P2 - P3 - P4

Pré-calcul du graphe pour toutes les paires de polyèdres :

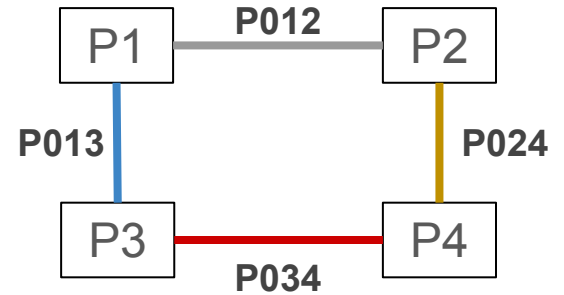


Graphe des fusions convexes (exemple)

- 4 polyèdres résultant des fusions pré-calculées



Graphe des fusions convexes

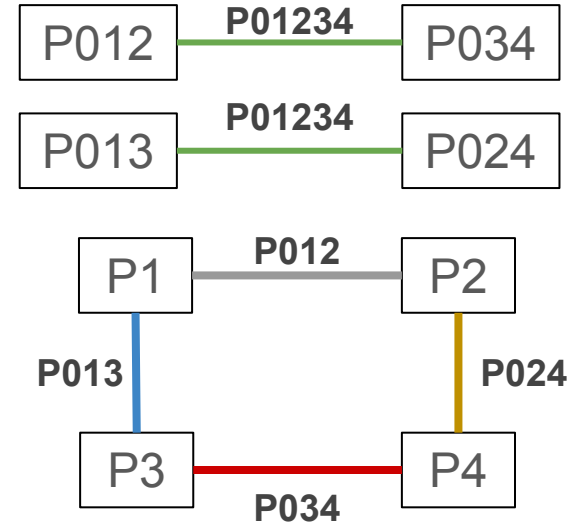


Graphe des fusions convexes (exemple)

- Complété dynamiquement par l'algorithme de fusion



Graphe des fusions convexes

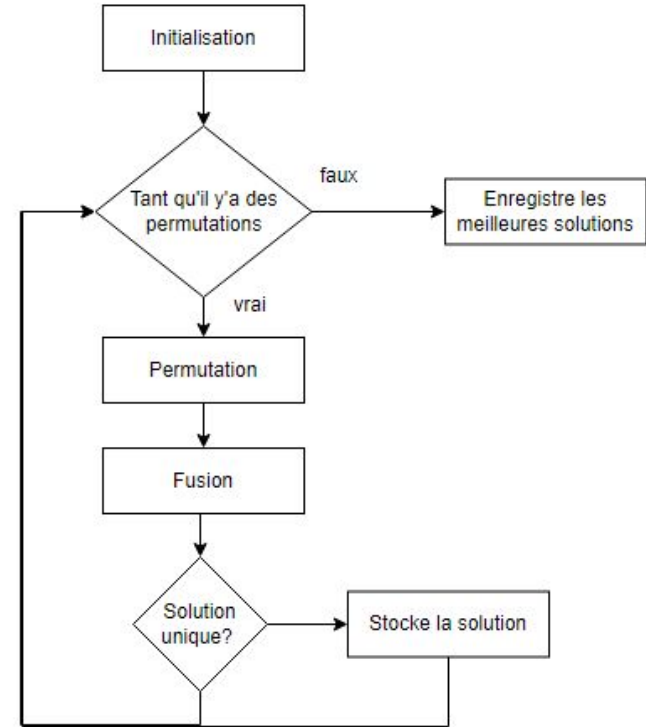




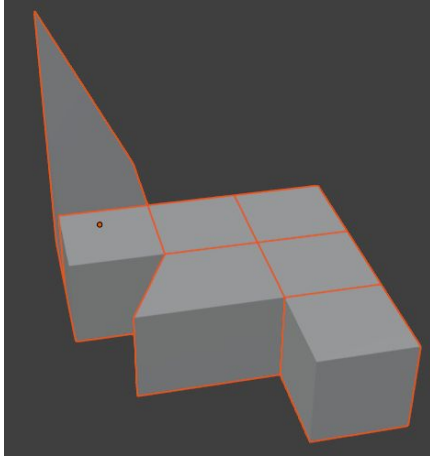
Méthode brute-force

Algorithme :

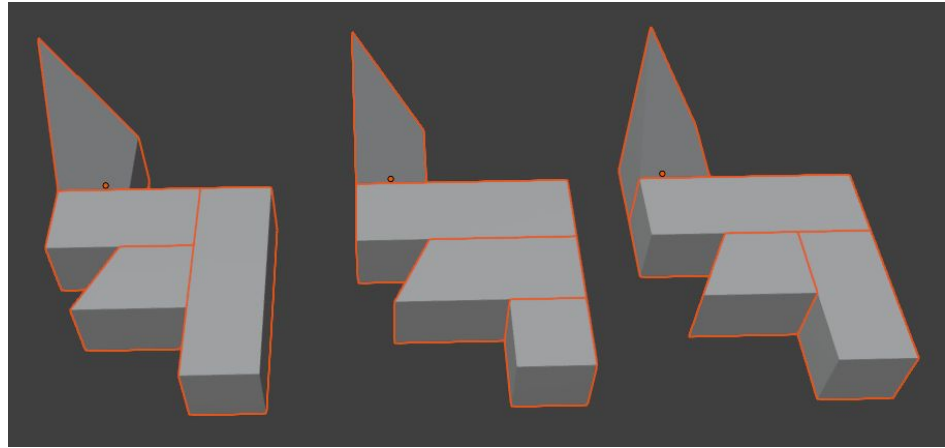
- Recherche exhaustive
- Plusieurs solutions peuvent donner les mêmes fusions
- Temps de calcul dépend du nombre de permutations (de l'ordre de $n!$)



Exemple :



*Fichier *.obj source*



Solutions générées par l'algorithme



Métaheuristiques

- Approcher la meilleure solution rapidement



Fonction objectif

Comment évaluer une solution (ensemble ordonné de polyèdres) ?

Résultat à minimiser :

- Ratio entre le nombre de polyèdres avant et après les fusions
- **Récompense** et **pénalité** pour prendre en compte d'autres critères

$$1 + \left(\frac{\text{Nombre de polyèdres après fusion}}{\text{Nombre de polyèdres}} \right) + \text{pénalité} - \text{récompense}$$



Fonction objectif - Récompense

Récompense, donne un bonus

- Nombre de fusions
- Taille des fusions

Initialement : taille des fusions trop avantageuses par rapport au nombre de fusions

Finalement : Grosses fusions légèrement avantageées

Fonction objectif - Pénalité

Pénalité, donne un malus

- Distance dans le graphe

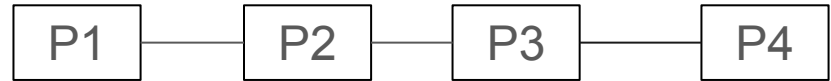
Exemple :

Solution : P1, P4, P3, P2

Fusion P1 et P4 : **pénalité1**

Fusion P3 et P2 : **pénalité2**

pénalité1 > **pénalité2**

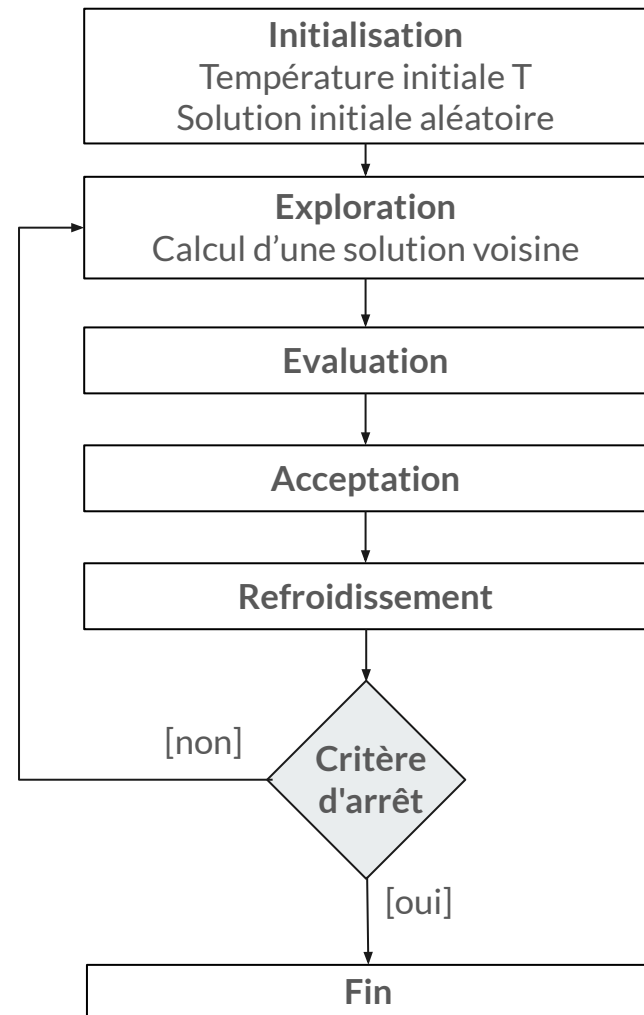




Recuit simulé

Principe :

- S'inspire de la physique, refroidissement lent d'un matériaux
- Méthode d'optimisation
- Modification progressive d'une solution pour explorer ses voisins



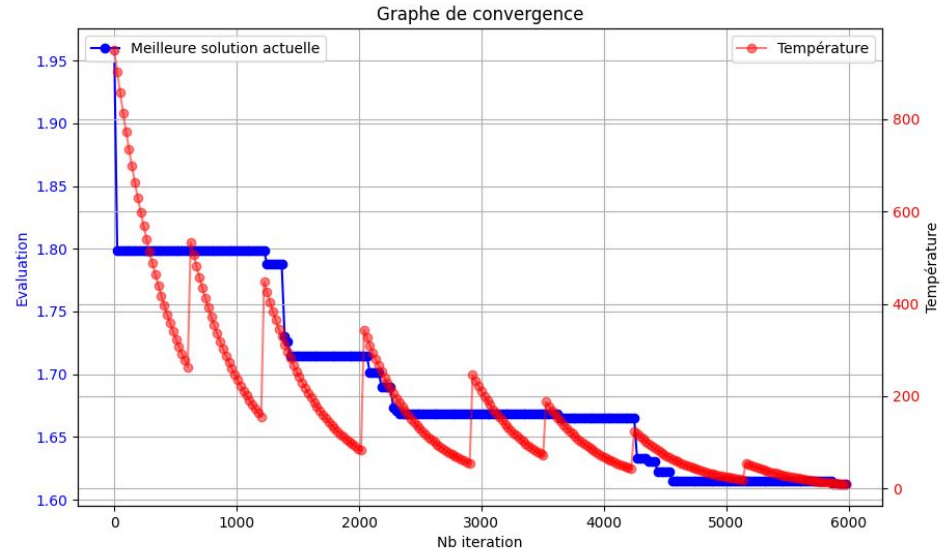
Refroidissement et critère d'arrêt

Température (en rouge sur le graphique)

- Facteur de refroidissement
- Peut augmenter si l'algorithme stagne un certain nombre d'itérations (ici **600**)

Arrêt

- Nombre d'itérations maximum atteint





Exploration

Calculer une solution voisine à la solution courante

- Permutation (mutation)

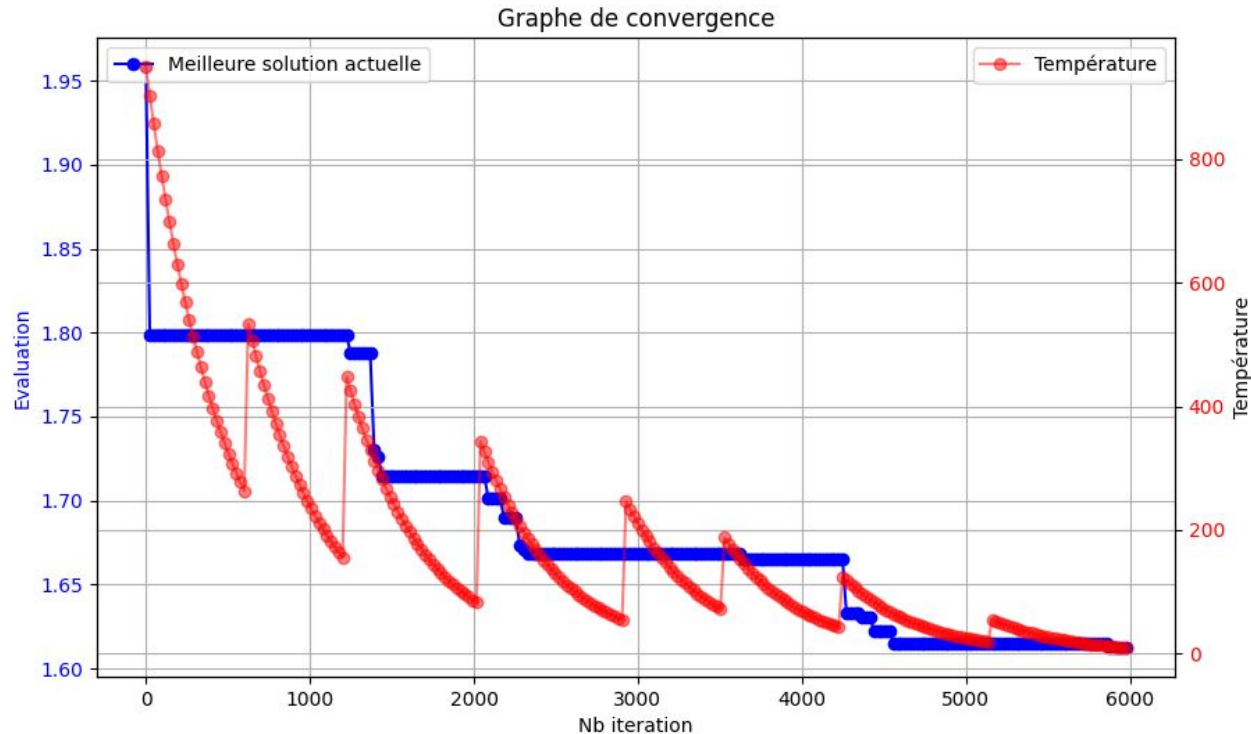
Exemple d'une permutation

- Solution courante : **P1** - **P2** - P3 - P4
- Un voisin possible : **P2** - **P1** - P3 - P4

Exploration

1. Permutation de 2 éléments côte à côte, 3 fois

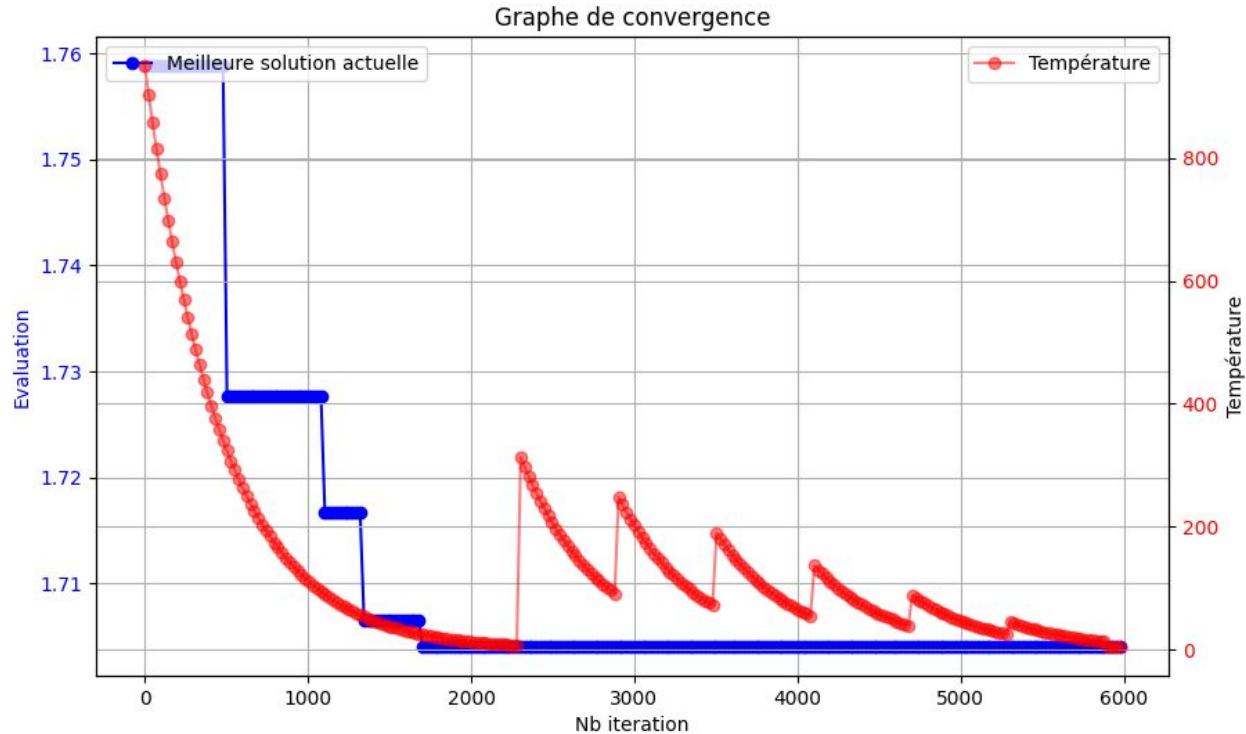
- Evaluation diminue peu
- Exemple : **1**, **2**, 3, 4, **5**, **6**, 7, 8, **9**, **10** → **2**, **1**, 3, 4, **6**, **5**, 7, 8, **10**, **9**



Exploration

2. Permutation de 2 éléments côte à côte, 1 fois

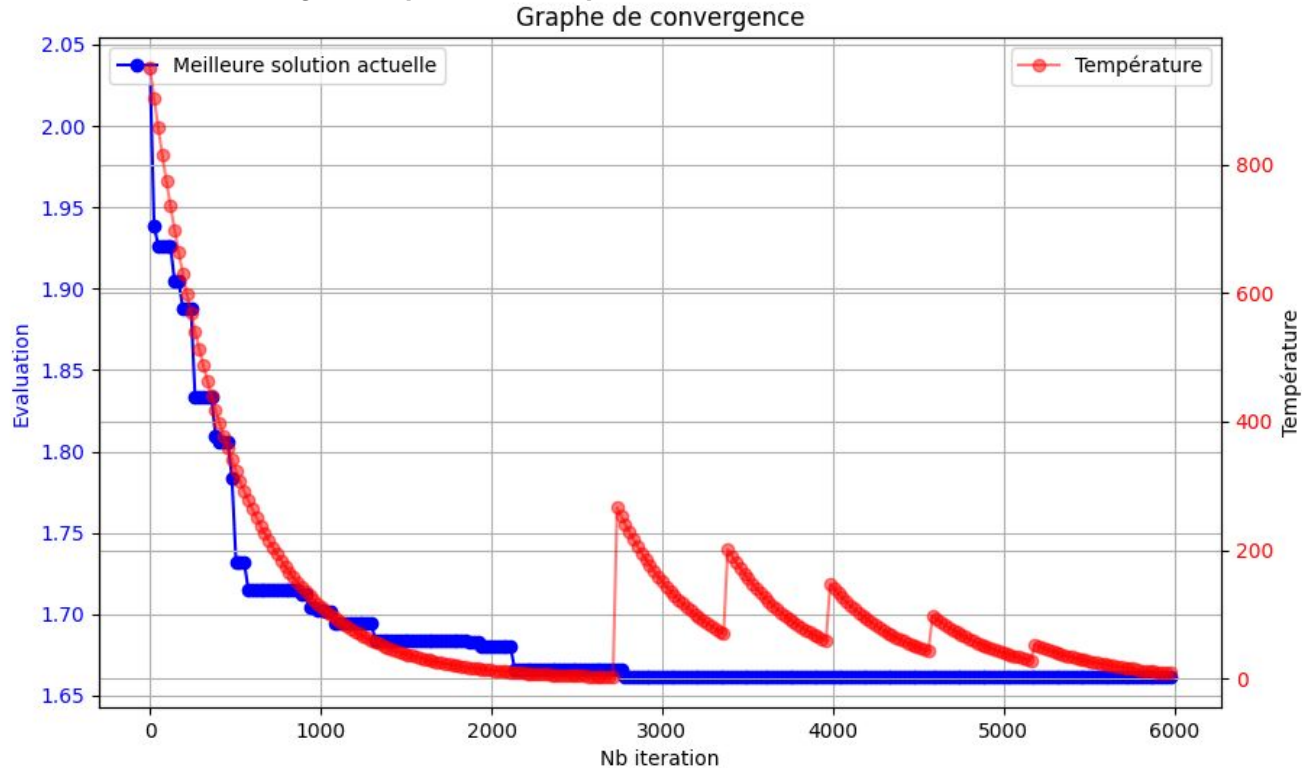
- Evaluation diminue très peu
- Exemple : **1,2**,3,4,5,6,7,8,9,10 → **2,1**,3,4,5,6,7,8,9,10



Exploration

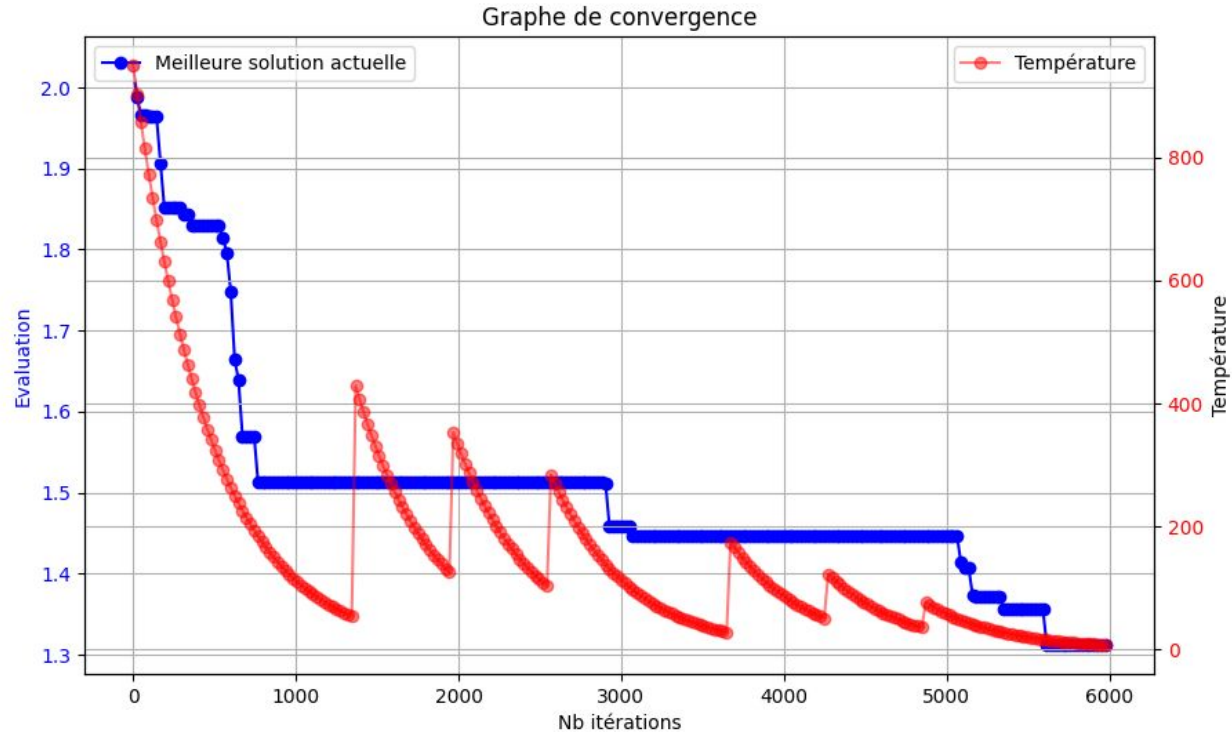
3. Nombre de permutations variable (de 5 fois à 2 fois)

- Pas bien ajusté pour les optima locaux



Exploration

4. 3 permutations aléatoires, 3 fois
- Convergence lente, mais sort des minima locaux

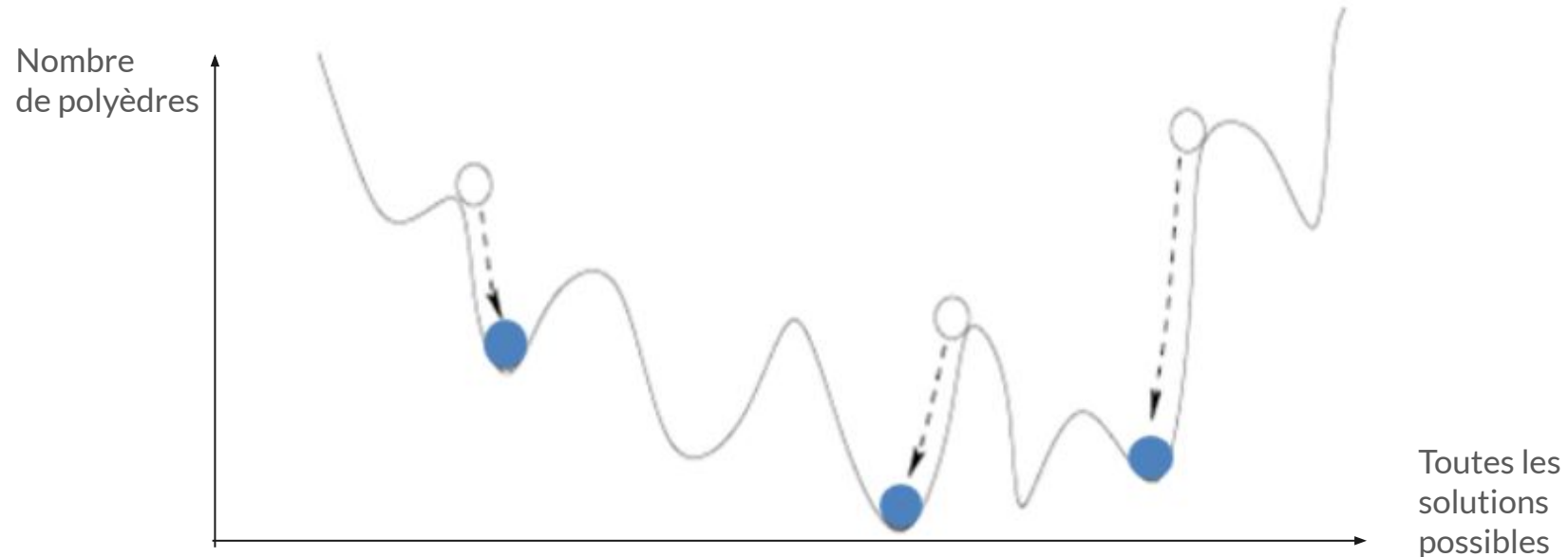




Évaluation et acceptation

- **Évaluation** de la solution voisine générée
 - Comparaison avec la solution courante
 - Si la solution voisine est meilleure, elle est acceptée
- **Critère d'acceptation :**
 - Accepter des **solutions plus mauvaises** avec une certaine probabilité
 - **Dépend de la température** courante
 - Température haute = Proba d'acceptation haute
 - Température basse = Proba d'acceptation basse

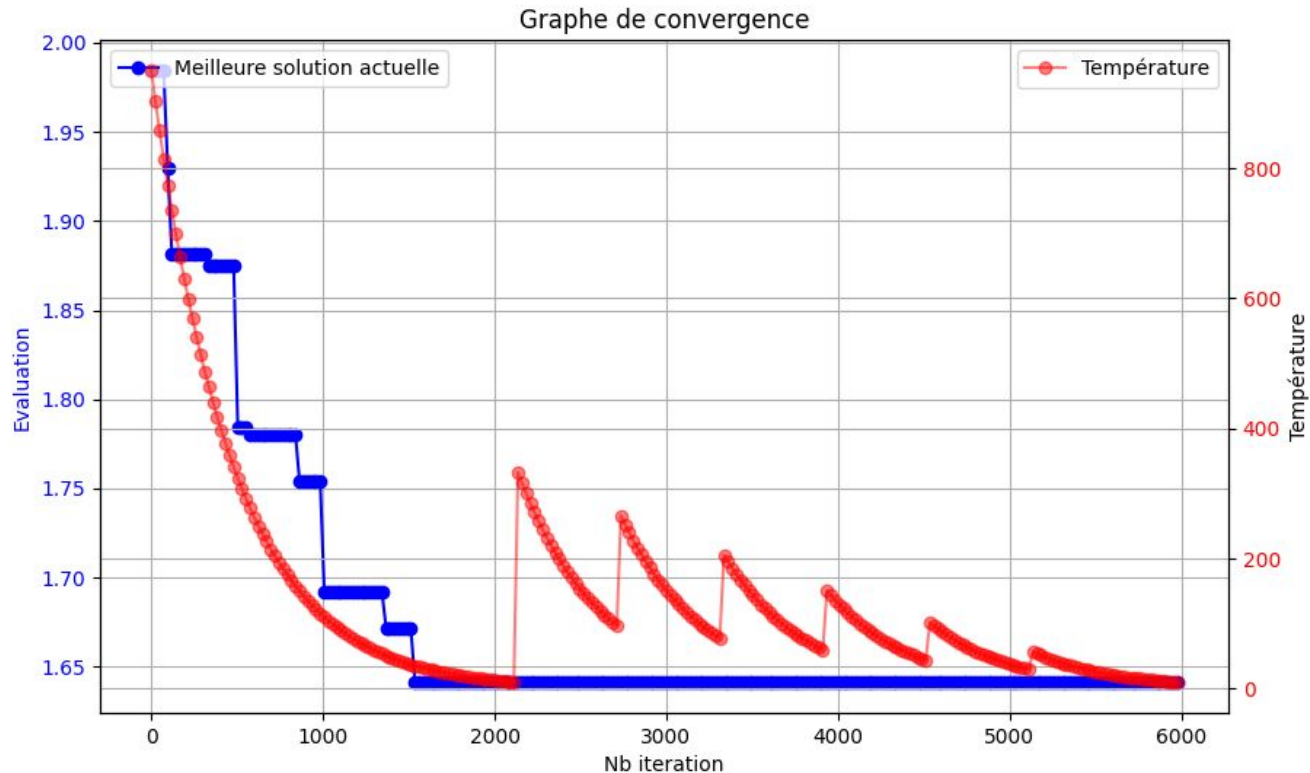
Évaluation et acceptation



Source : <https://www.mcours.net/cours/pdf/hassbg/hassbgli902.pdf>

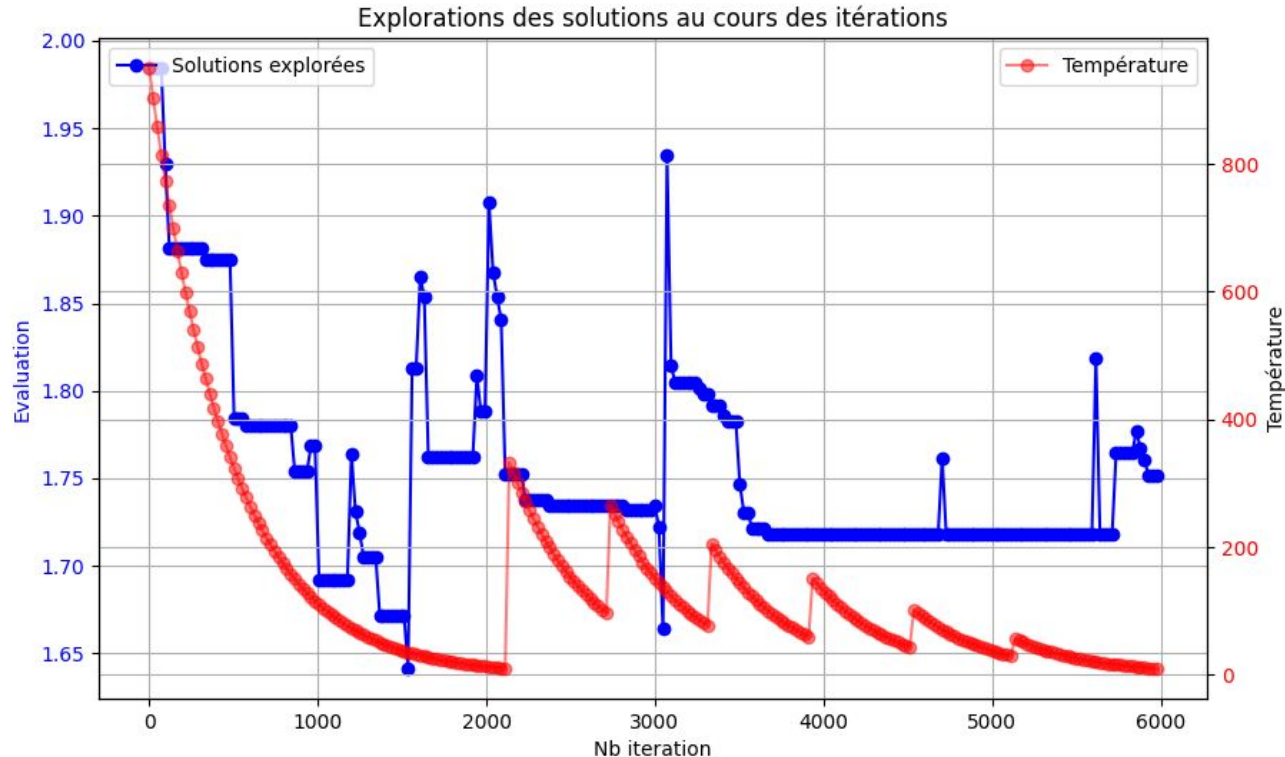
Évaluation et acceptation

1. Acceptation de plus mauvaises solutions trop peu probable
 - Graphe des meilleures solutions trouvées en fonction des itérations



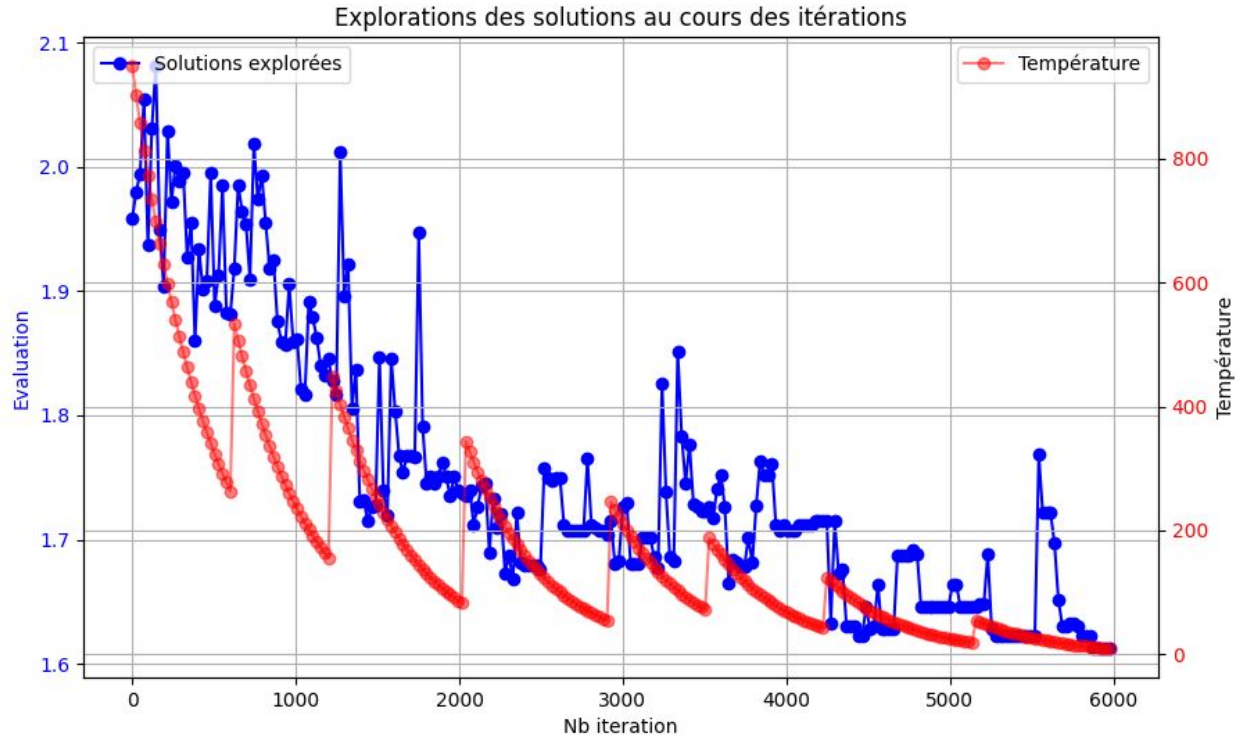
Évaluation et acceptation

1. Acceptation de plus mauvaises solutions trop peu probable
 - Graphe des solutions acceptées en fonction des itérations



Évaluation et acceptation

2. Acceptation de plus mauvaises solutions plus élevé
 - Graphe des solutions explorées en fonction des itérations





Conclusion des tests

Plus efficace avec :

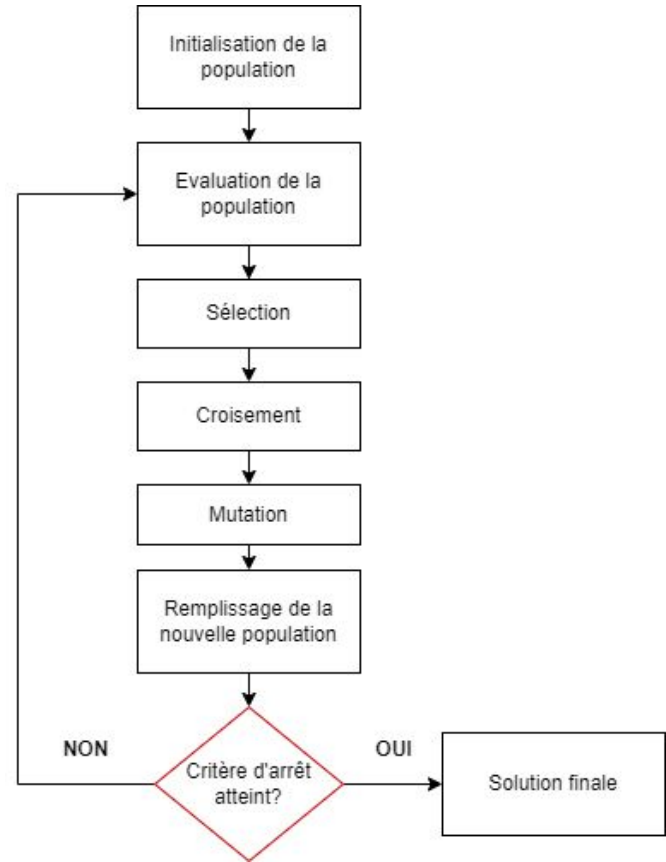
- Permutations aléatoires
- Peu de permutations ou un nombre variable de permutations
- Faire remonter la température rapidement si l'algorithme stagne



Algorithme génétique

Principe :

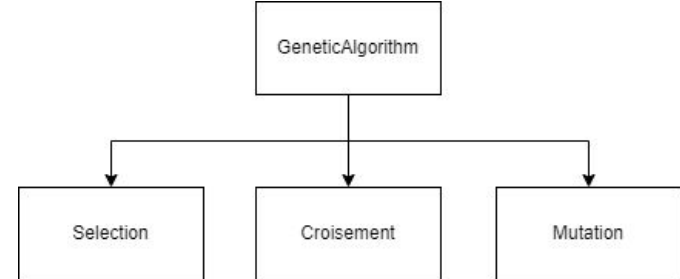
- S'inspire de la sélection naturelle
- Méthode d'optimisation
- Explore plusieurs directions en même temps



Implémentation

- Sélection élitiste
- Sélection par tournoi
- Sélection par roulette
- Croisement en N points
- Mutation par insertion

- Structure modulaire



Sélection élitiste

- Trie de la population par score
- Sélectionne les N individus avec les meilleurs scores

Exemple : N = 3

Score : 2.8

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Score : 5.2

2	5	1	3	4	6	7
---	---	---	---	---	---	---

Score : 6.5

6	3	2	7	1	4	5
---	---	---	---	---	---	---

Score : 6.5

5	4	7	1	3	2	6
---	---	---	---	---	---	---

Score : 18.4

6	2	1	4	3	7	5
---	---	---	---	---	---	---

Sélection élitiste

- Trie de la population par score
- Sélectionne les N individus avec les meilleurs scores

Exemple : N = 3

- Individus ajoutés dans la liste de parents

Score : 2.8

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Score : 5.2

2	5	1	3	4	6	7
---	---	---	---	---	---	---

Score : 6.5

6	3	2	7	1	4	5
---	---	---	---	---	---	---

Score : 6.5

5	4	7	1	3	2	6
---	---	---	---	---	---	---

Score : 18.4

6	2	1	4	3	7	5
---	---	---	---	---	---	---

Sélection élitiste

- Trie de la population par score
- Sélectionne les N individus avec les meilleurs scores

Exemple : N = 3

- Individus ajoutés dans la liste de parents

Score : 2.8

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Score : 5.2

2	5	1	3	4	6	7
---	---	---	---	---	---	---

Score : 6.5

6	3	2	7	1	4	5
---	---	---	---	---	---	---

Sélection par tournoi

- Sélectionne un sous-ensemble aléatoire de taille N
- Compare les scores
- Sélectionne le gagnant / l'individu avec le meilleur score

Exemple : N = 2

Score : 2.8

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Score : 5.2

2	5	1	3	4	6	7
---	---	---	---	---	---	---

Score : 6.5

6	3	2	7	1	4	5
---	---	---	---	---	---	---

Score : 6.5

5	4	7	1	3	2	6
---	---	---	---	---	---	---

Score : 18.4

6	2	1	4	3	7	5
---	---	---	---	---	---	---

Sélection par tournoi

- Sélectionne un sous-ensemble aléatoire de taille N
- Compare les scores
- Sélectionne le gagnant / individu avec le meilleur score

Exemple : N = 2

- Individu 2 et 5 dans le tournoi
- 2 a un meilleur score que 5

Score : 2.8

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Score : 5.2

2	5	1	3	4	6	7
---	---	---	---	---	---	---

Score : 6.5

6	3	2	7	1	4	5
---	---	---	---	---	---	---

Score : 6.5

5	4	7	1	3	2	6
---	---	---	---	---	---	---

Score : 18.4

6	2	1	4	3	7	5
---	---	---	---	---	---	---

Sélection par tournoi

- Sélectionne un sous-ensemble aléatoire de taille N
- Compare les scores
- Sélectionne le gagnant / individu avec le meilleur score

Exemple : N = 2

- Individu 2 ajouté dans la liste des parents

Score : 2.8

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Score : 5.2

2	5	1	3	4	6	7
---	---	---	---	---	---	---

Score : 6.5

6	3	2	7	1	4	5
---	---	---	---	---	---	---

Score : 6.5

5	4	7	1	3	2	6
---	---	---	---	---	---	---

Score : 18.4

6	2	1	4	3	7	5
---	---	---	---	---	---	---

Sélection par roulette

- Sélectionne aléatoirement un individu et l'ajoute à la liste de parents.

Exemple :

- Tir au hasard l'individu 3**
- Individu 3 ajouté dans la liste des parents**

Score : 2.8

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Score : 5.2

2	5	1	3	4	6	7
---	---	---	---	---	---	---

Score : 6.5

6	3	2	7	1	4	5
---	---	---	---	---	---	---

Score : 6.5

5	4	7	1	3	2	6
---	---	---	---	---	---	---

Score : 18.4

6	2	1	4	3	7	5
---	---	---	---	---	---	---

Croisement en N points

- Sélectionne N point de croisement différents
- Croise deux individus entre ces N points

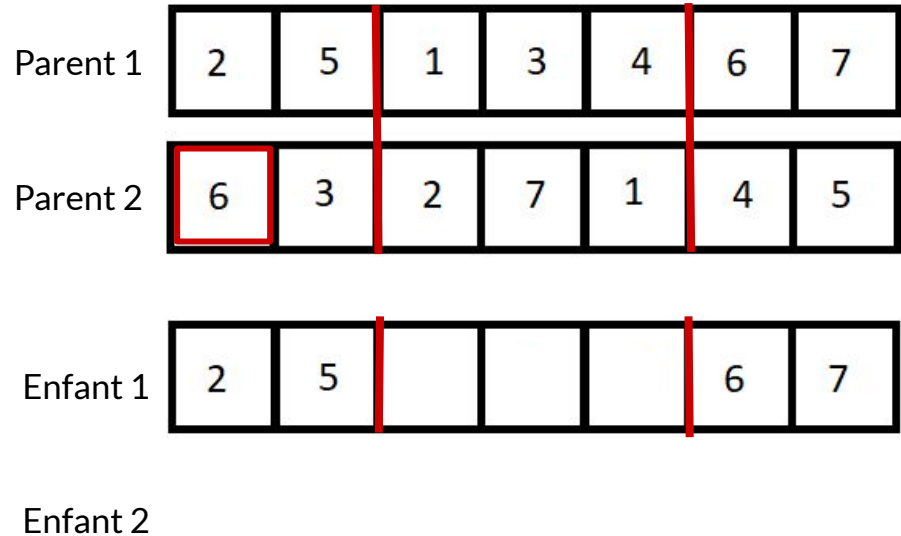
Exemple : N = 2

Parent 1	2	5	1	3	4	6	7
Parent 2	6	3	2	7	1	4	5

Croisement en N points

- Sélectionne N point de croisement différents
- Croise deux individus entre ces N points

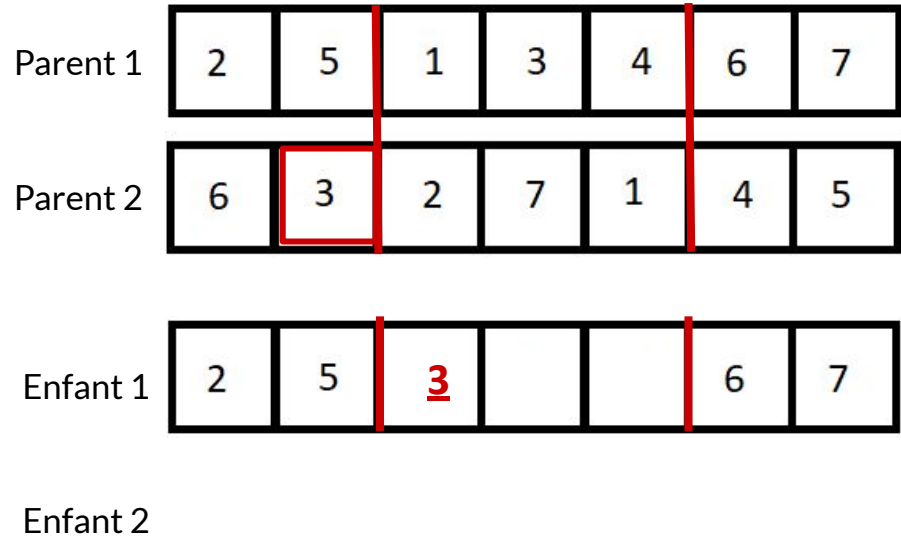
Exemple : N = 2



Croisement en N points

- Sélectionne N point de croisement différents
- Croise deux individus entre ces N points

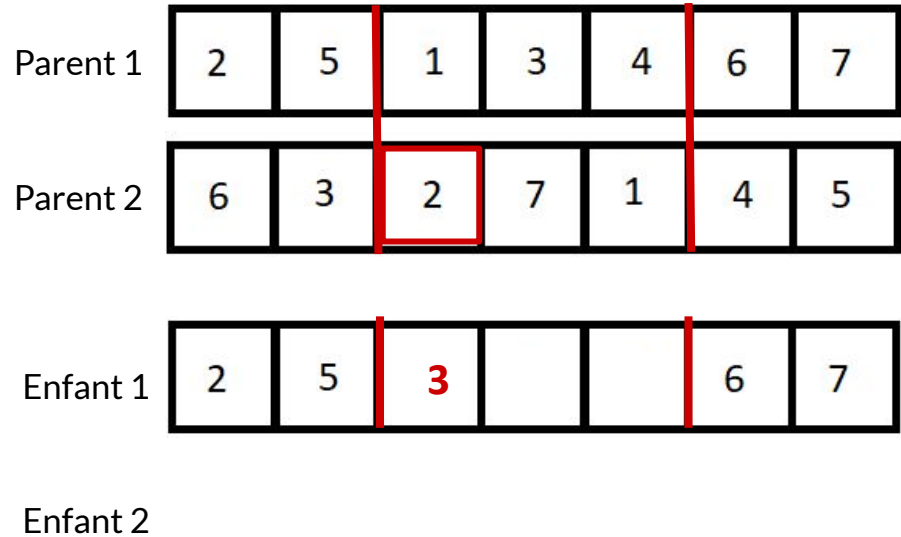
Exemple : N = 2



Croisement en N points

- Sélectionne N point de croisement différents
- Croise deux individus entre ces N points

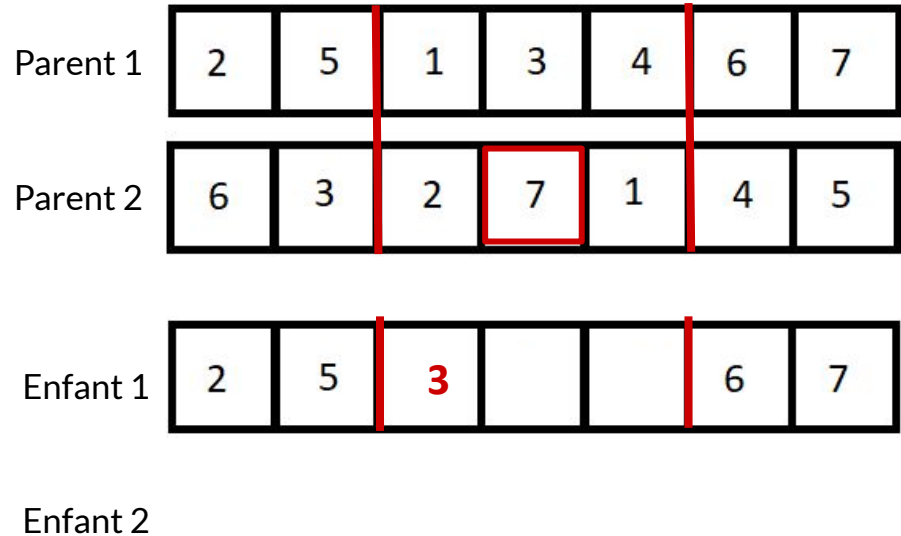
Exemple : N = 2



Croisement en N points

- Sélectionne N point de croisement différents
- Croise deux individus entre ces N points

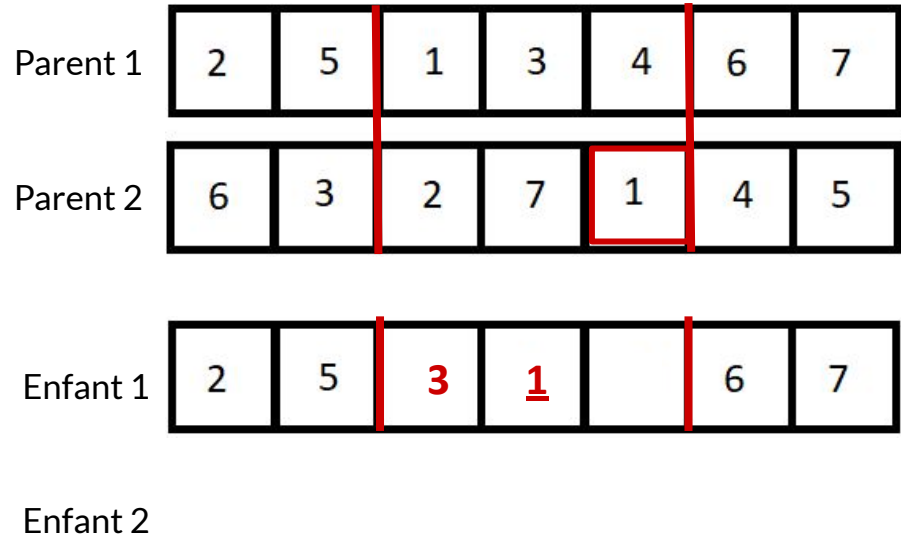
Exemple : N = 2



Croisement en N points

- Sélectionne N point de croisement différents
- Croise deux individus entre ces N points

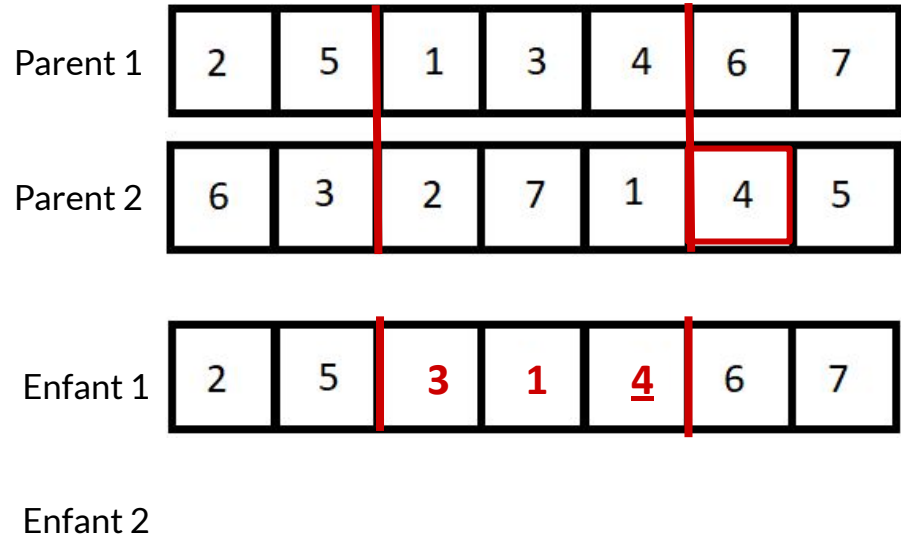
Exemple : N = 2



Croisement en N points

- Sélectionne N point de croisement différents
- Croise deux individus entre ces N points

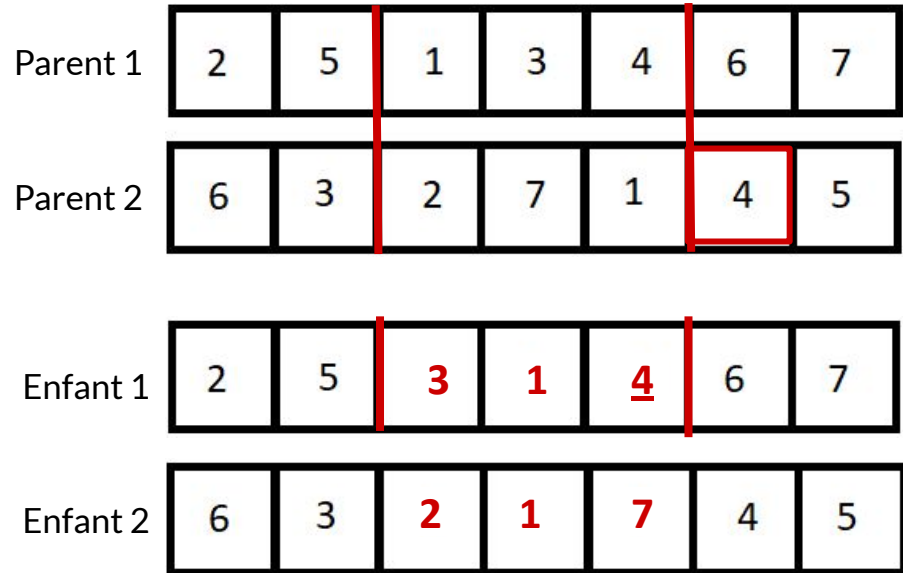
Exemple : N = 2



Croisement en N points

- Sélectionne N point de croisement différents
- Croise deux individus entre ces N points

Exemple : N = 2





Mutation par insertion

- Sélectionne aléatoirement une position n .
- Sélectionne aléatoirement une valeur v .
- Remplace la valeur à la position n par v

Exemple : $n = 5$ et $v = 2$



Mutation par insertion

- Sélectionne aléatoirement une position n .
- Sélectionne aléatoirement une valeur v .
- Remplace la valeur à la position n par v

Exemple : $n = 5$ et $v = 2$





Mutation par insertion

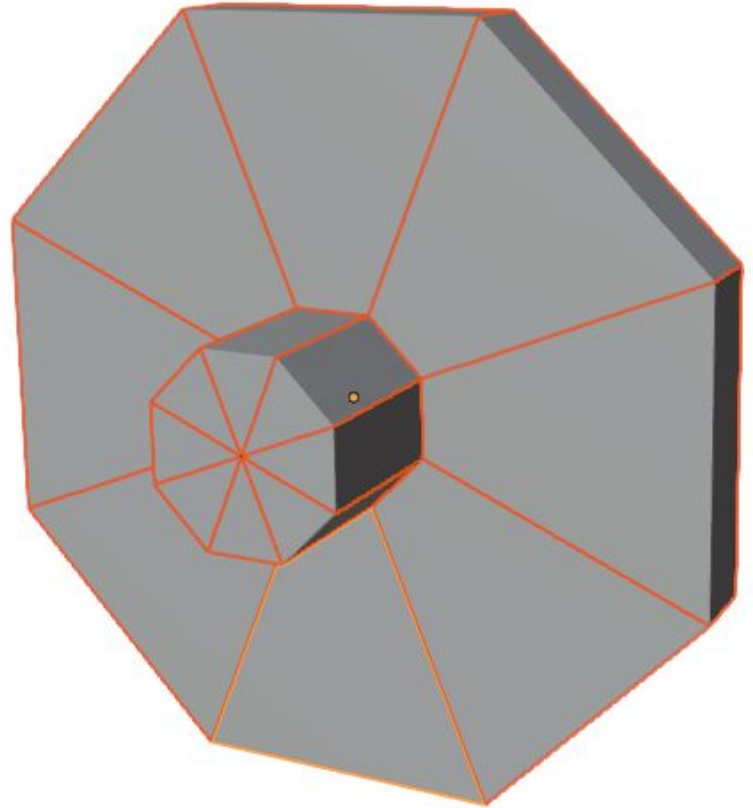
- Sélectionne aléatoirement une position n .
- Sélectionne aléatoirement une valeur v .
- Remplace la valeur à la position n par v

Exemple : $n = 5$ et $v = 2$



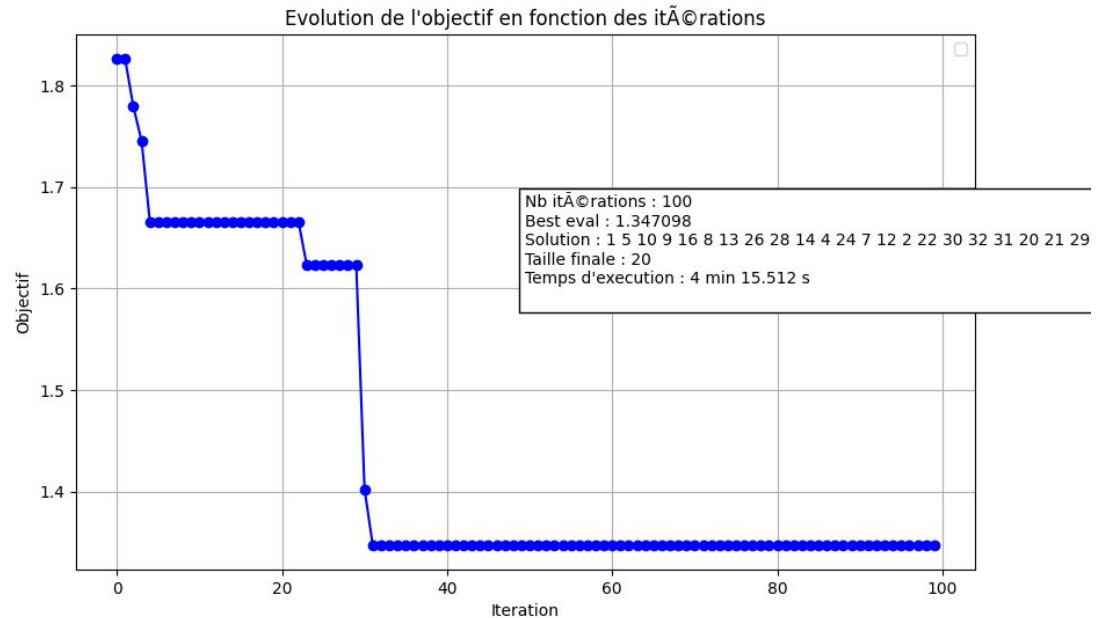
Comparaison et résultats

- Taille de population : 100
- Nombre d'itérations maximum : 100
- Probabilité de mutation : 0.8
- Crossover en 7 points
- Exemple de la roue



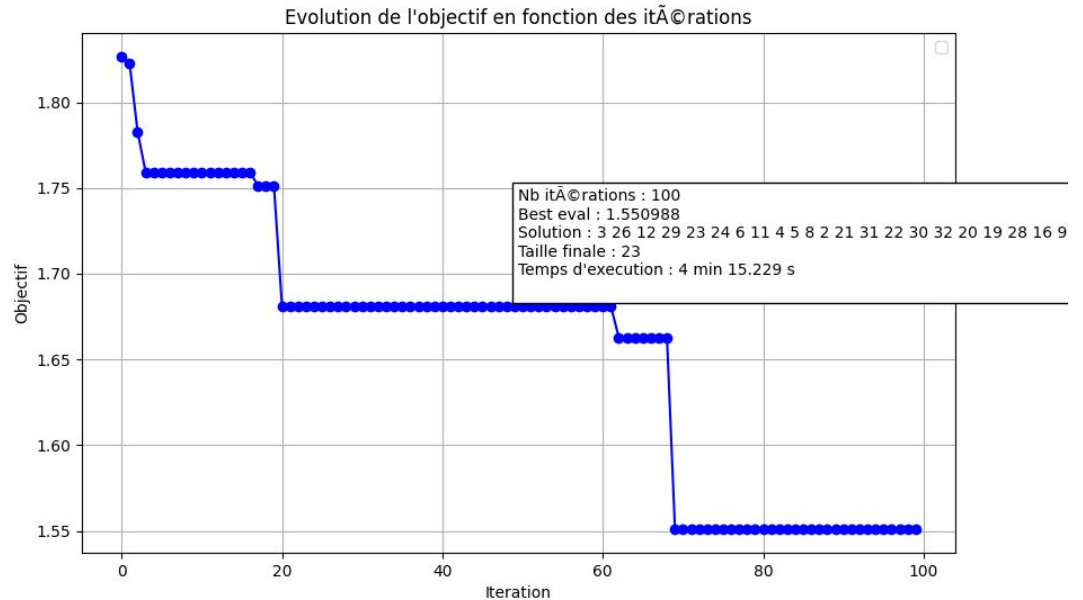
Sélection par tournoi

- Stagne pendant longtemps
- Possible manque de diversité
- Population qui se ressemble rapidement
- Taille finale : 20 polyèdres



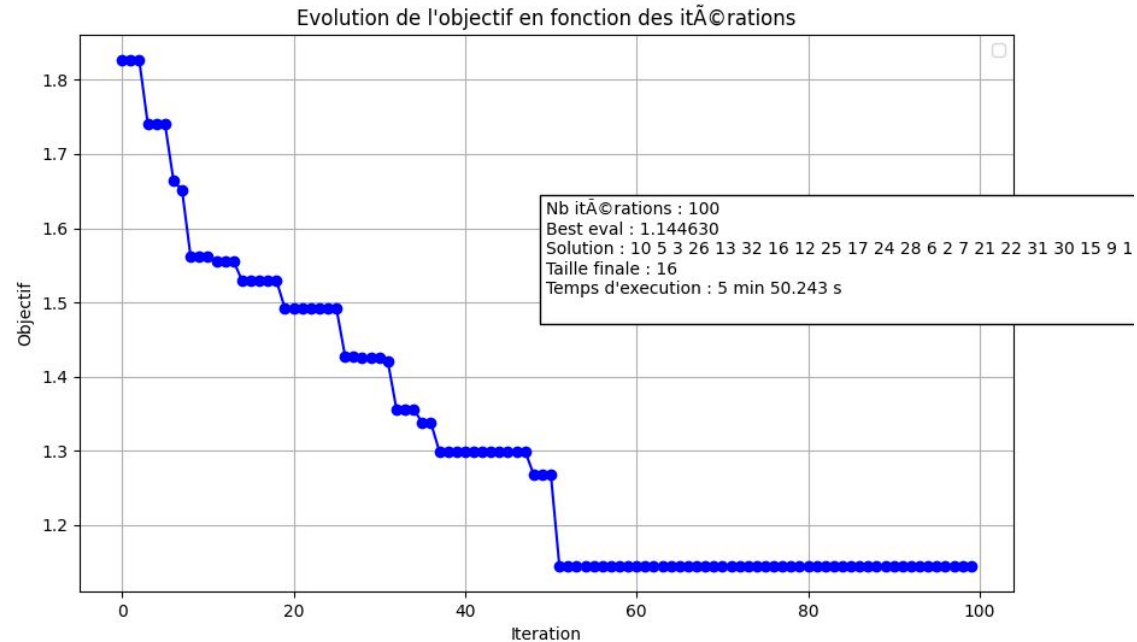
Sélection par roulette

- Tendance à se coincer sur ses plateaux / minima locaux
- Possible manque de diversité
- Population qui se ressemble rapidement
- Pire que la sélection par tournoi
- Taille finale : 23 polyèdres



Sélection élitiste

- Tendance à beaucoup moins se coincer sur des plateaux
- Converge vers l'objectif mais stagne à un certain stade
- Meilleure sélection testée jusqu'à présent
- Taille finale : 16 polyèdres





Création de la nouvelle population

- Remplir la première moitié avec les enfants des parents issus de la sélection
- Remplir la deuxième moitié avec les enfants des plus mauvais individus
- Permet de garder une certaine diversité



Améliorations possibles



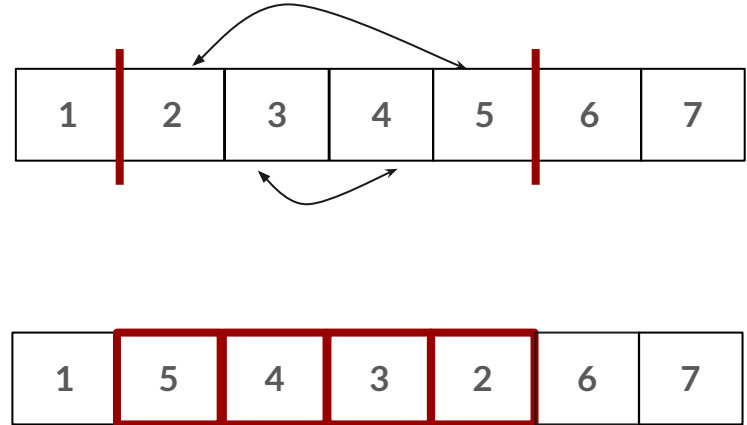
Perspectives

- Paralléliser le code au maximum
- Essayer d'autres métaheuristiques pour comparer
- Ajouter des sélections, croisements et mutations
- **Faire évoluer l'évaluation au cours de l'algorithme**
 - Permet de favoriser certains critères

Perspectives - mutation

Inverser l'ordre de N gènes côte à côte
(mutation par inversion)

- Préserve les fusions d'un même bloc
- Grand impact possible sur la pénalité d'évaluation
- Permet éventuellement de casser ou d'ajouter des fusions aux extrémités

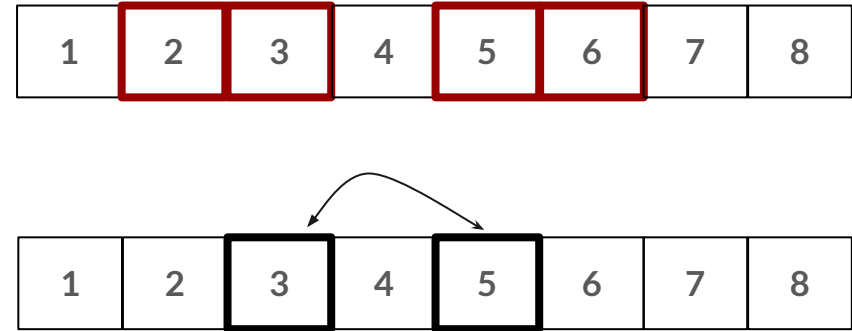


Perspectives - mutation

Éviter de casser les fusions avec les mutations selon certains critères

- Exemple
 - 2 et 1 sont fusionnables
 - 5 et 6 sont fusionnables

→ Éviter de permuter 3 et 5





Conclusion

- Algorithme de fusion
- Métaheuristique
- Optimisation



Merci pour votre attention