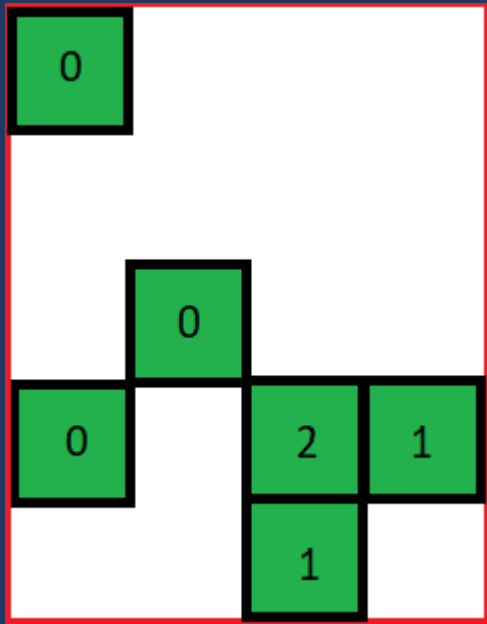


Présentation de Projet Recherche Opérationnelle

Modélisation du problème

Compacité

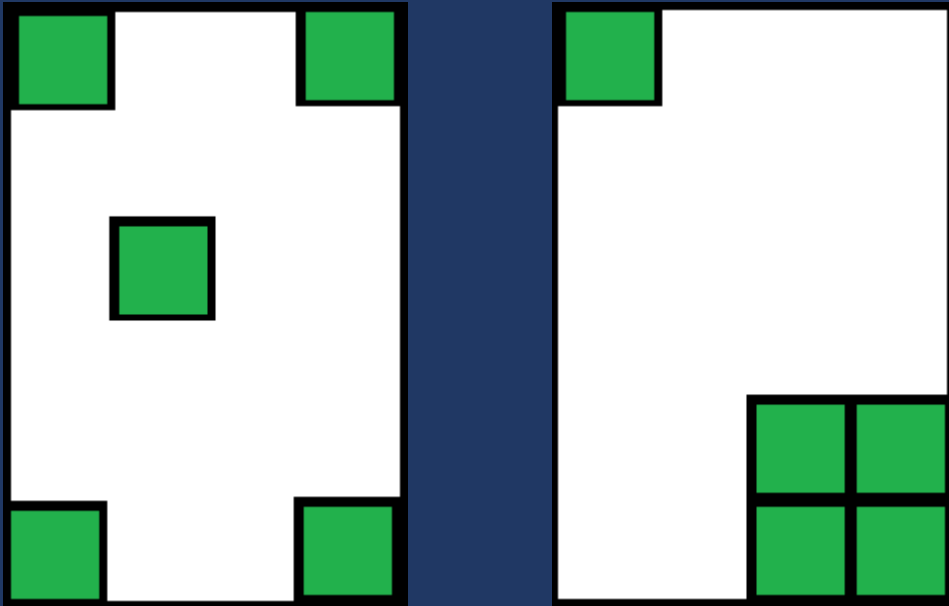
- S = surface du plus petit rectangle contenant tous les champs
- a_i = nombre de champs adjacents au champ i .
- Compacité = $\frac{S}{1 + \sum a_i}$
- Exemple :



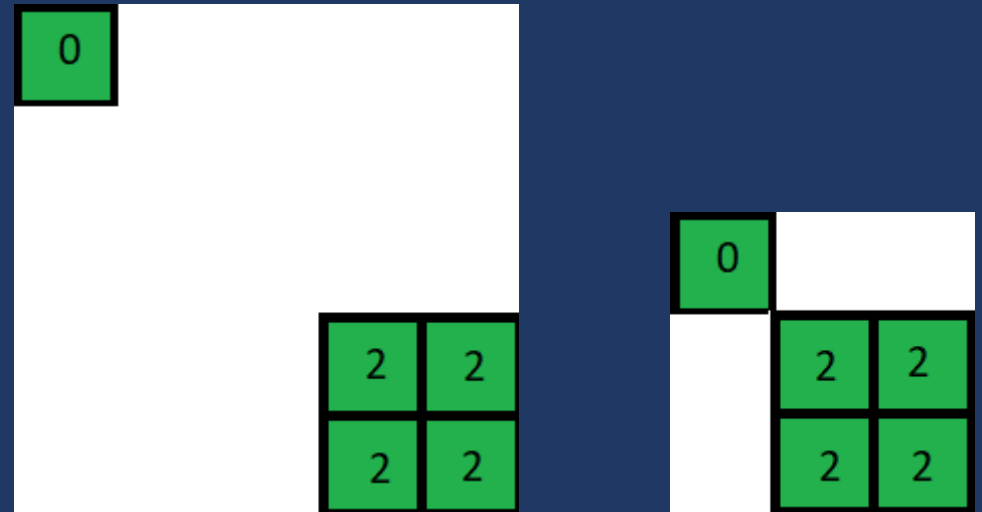
$$\text{Compacité} = \frac{4 \times 5}{1 + 4} = 4$$

Justification de la modélisation de la compacité

Surface

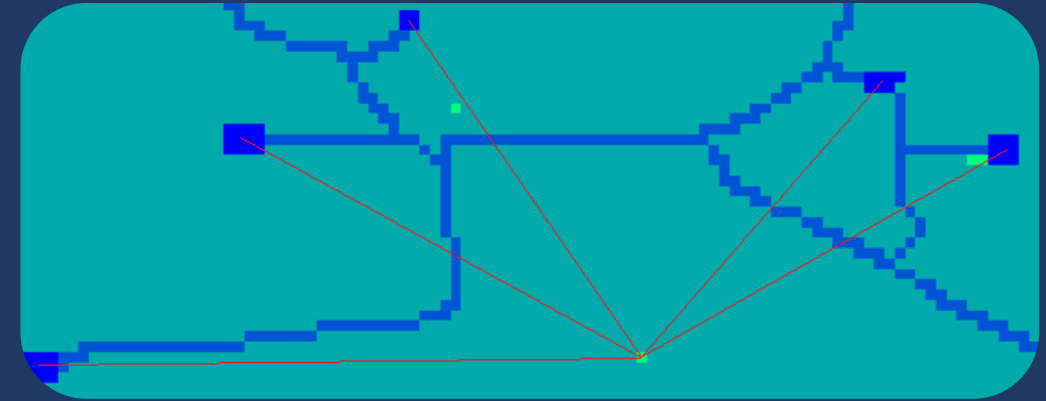


Adjacence



Distance des habitations

- n_i = distance moyenne entre le champ i et les habitations
- d_{ij} = distance entre le champ i et l'habitation j
- $n_i = \frac{1}{d} \sum_j d_{ij}$
- Score de distance des habitations = $\frac{1}{n} \sum_i n_i$



Production

- p_i = score de production du champ i retrouvé dans la carte de production
- Score de production = $\sum_i p_i$

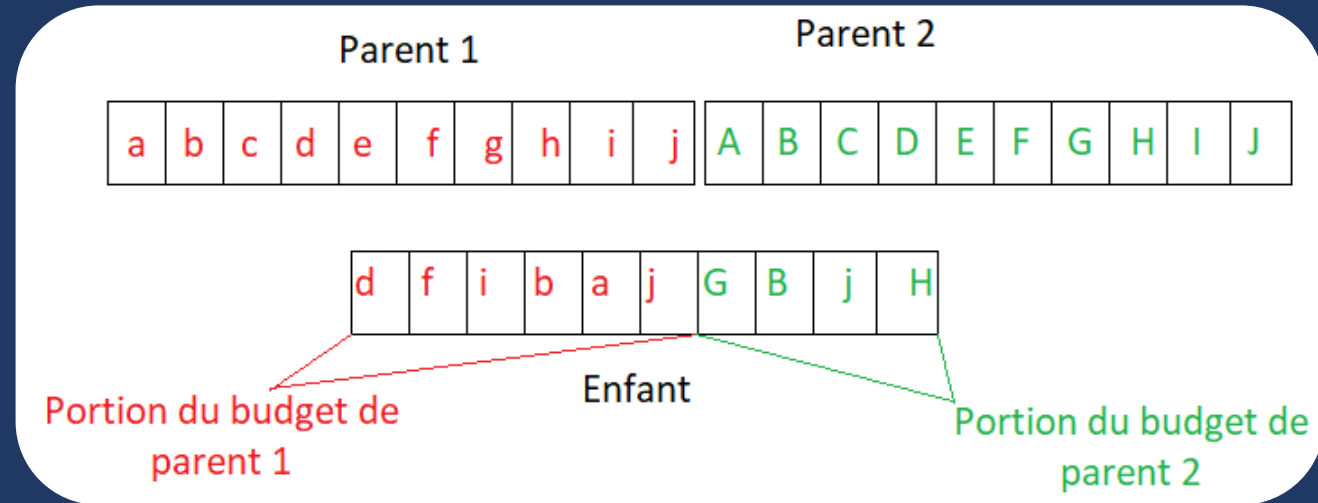
Principes de l'algorithme génétique



Initialisation

- Un parent est une liste de champs
- Lors de la première génération, des champs sont ajoutés aléatoirement au parent tant que l'ajout de ceux-ci ne dépasse pas le budget alloué
- Plusieurs iterations sont effectuées afin de se rapprocher le plus possible du budget maximal

Crossover



- Deux parents se croisent pour former un enfant
- Le croisement se fait en deux parties:
 - Une portion du budget de l'enfant est choisie aléatoirement. Les champs du parent 1 sont ajoutés aléatoirement tant que le coût total ne dépasse pas la portion de budget allouée.
 - Lorsque les champs du parent 1 ont été ajoutés, les champs du parents 2 sont à leurs tours ajoutés aléatoirement tant que le coût total ne dépasse pas le budget total de l'enfant.
- Une mutation a lieu aléatoirement lors du croisement. Celle-ci consiste à intervertir, au hasard, un champ de l'enfant avec un champ aléatoire.

Nouvelle génération

Liste de scores classée
par production
croissante

1. Enfant 1 (1,1,4)
2. Enfant 2 (1,2,5)
3. Enfant 4 (2,4,3)
4. ...
5. ...
6. **Enfant 3 (4,3,1)**
7. ...

Liste de scores classée
par distance
décroissante

1. ...
2. ...
3. ...
4. Enfant 4 (2,4,3)
5. **Enfant 3 (4,3,1)**
6. Enfant 2 (1,2,5)
7. Enfant 1 (1,1,4)

Liste des scores classée
par compacité
décroissante

1. Enfant 2 (1,2,5)
2. Enfant 1 (1,1,4)
3. Enfant 4 (2,4,3)
4. ...
5. ...
6. ...
7. **Enfant 3 (4,3,1)**

Liste de score général
triée par ordre
décroissant

1. ...
2. ...
3. ...
4. **Enfant 3 (18)**
5. ...
6. ...
7. ...

Portion
d'élite

$$\text{Score général Enfant 3} = 6 + 5 + 7$$

Score général de l'Enfant i = position production + position proximité + position compacité

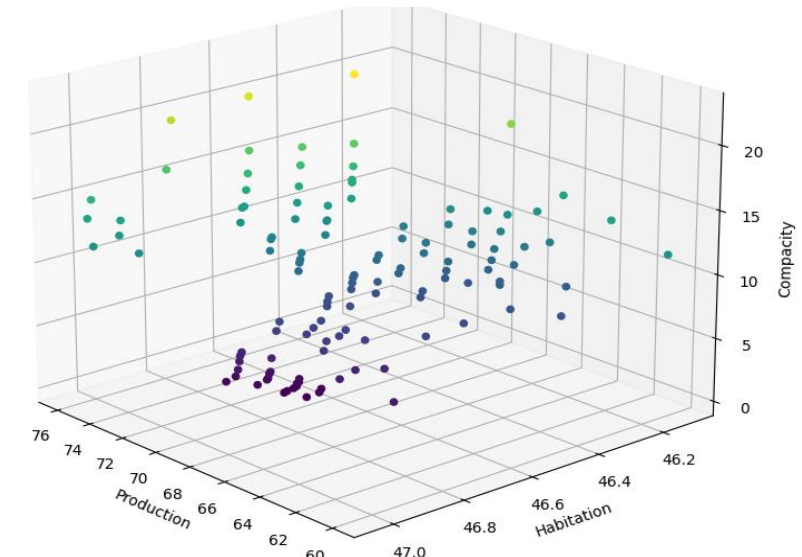
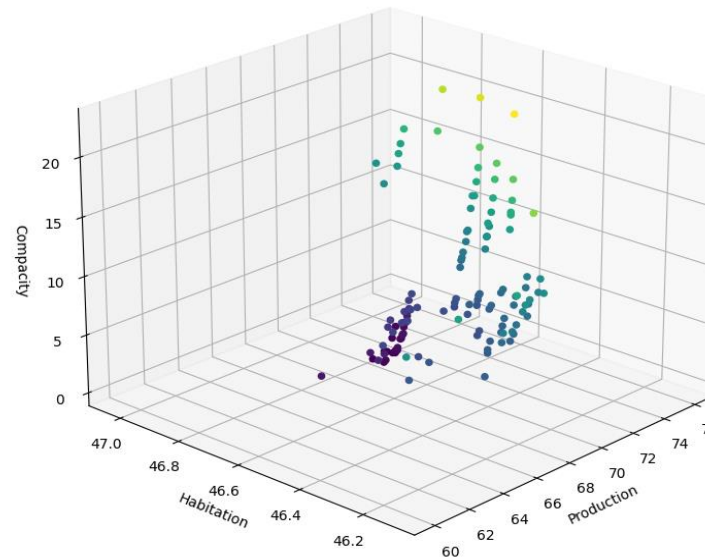
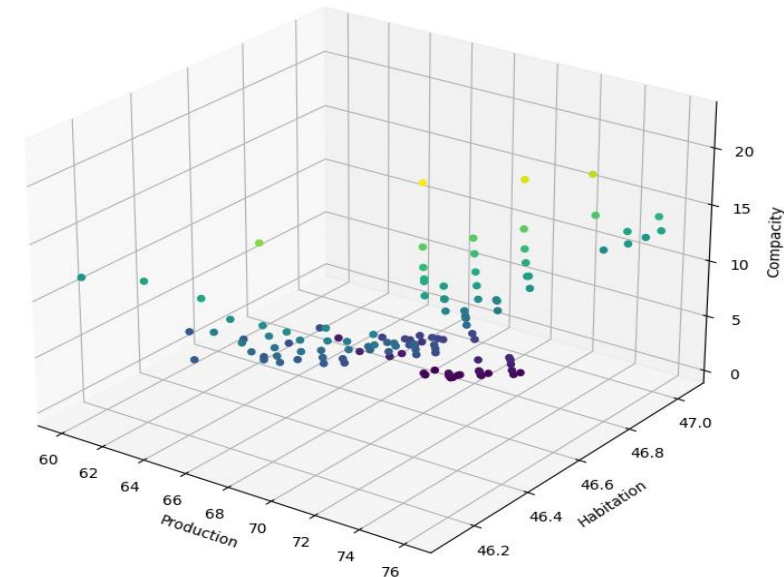
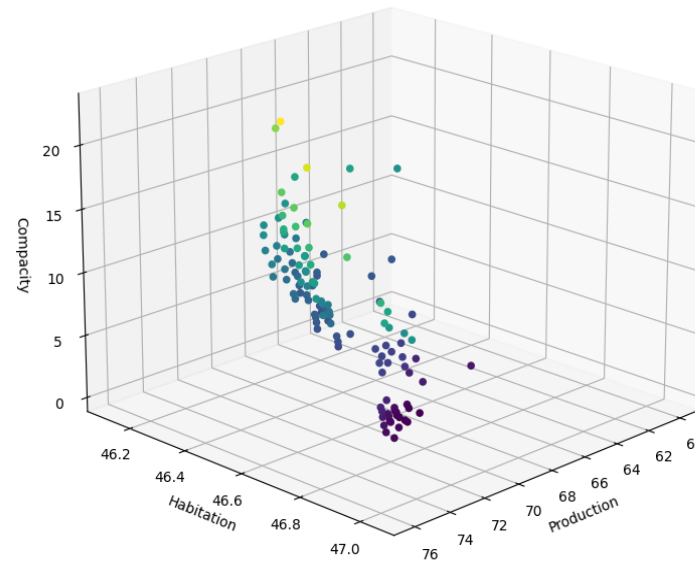
Pareto

Pareto

- Permet de trouver les solutions dominantes qui satisfont les 3 critères de manière équilibrée
- Seules les solutions à la frontière Pareto sont sélectionnées
- Une liste est initialisée avec la première génération et mise à jour à chaque nouvelle génération en excluant les solutions dominées

Représentation Pareto

- Représentation de la frontière en 3 dimensions :
 - Production : à maximiser
 - Compacité : à minimiser
 - Distance d'habitations : à minimiser



The background of the image is a dark blue gradient. A large, lighter blue circular arc is positioned on the left side, partially cut off by the edge. The word "PROMETHEE" is centered in the middle of the image in a white, sans-serif font.

PROMETHEE

PROMETHEE

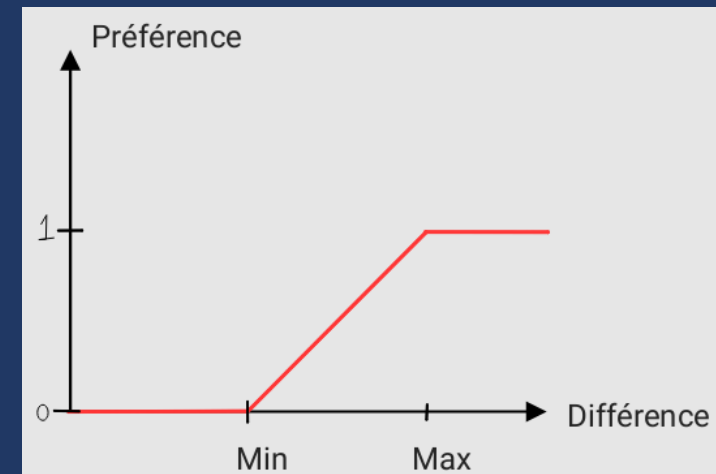
Algorithme utilisé pour sélectionner la meilleure alternative parmi les solutions Pareto

Pour chaque critère on définit:

- Un poids pour représenter l'importance d'un critère par rapport aux autres
- Une préférence avec une valeur de score minimum et maximum

Solutions comparées par leurs scores :

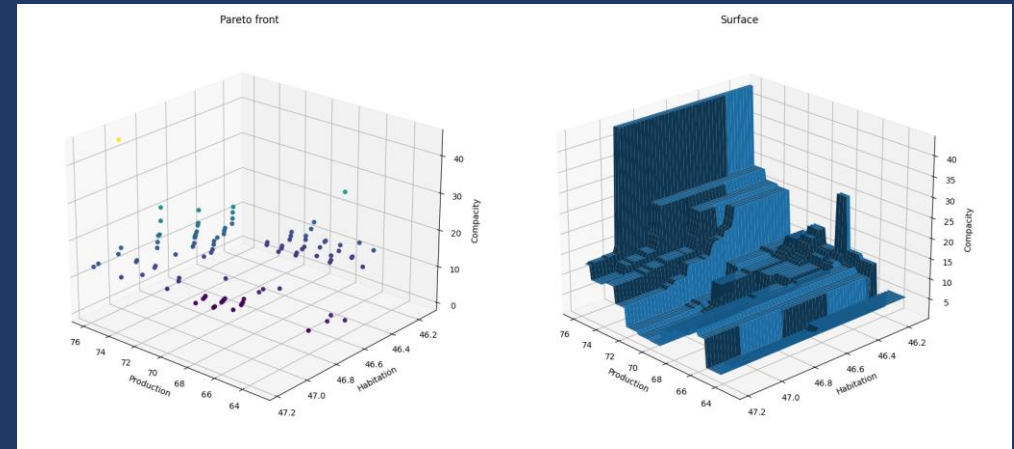
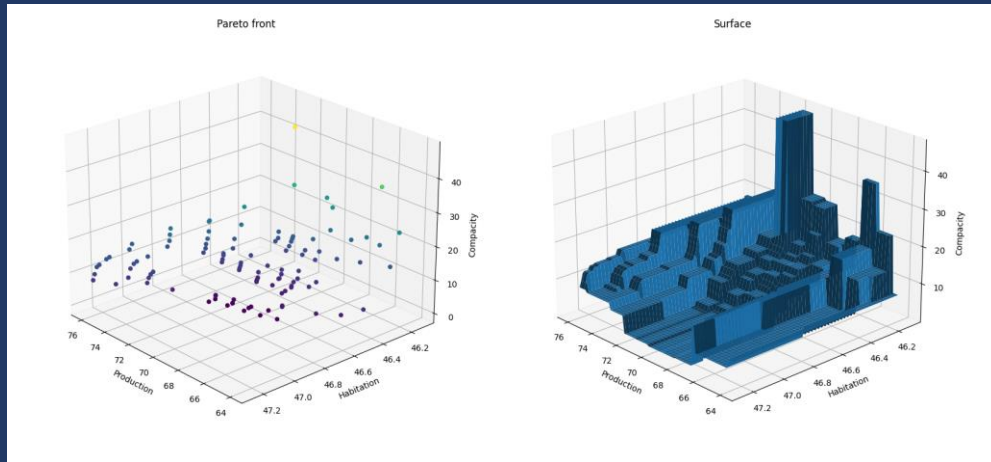
- On fait une différence entre les 2 scores
- Le poids est multiplié par la préférence et est ajouté au score de comparaison



Sélection finale de la solution

- À la fin de l'algorithme, il reste une matrice carrée des comparaisons entre tous les scores
- Pour chaque solution, un calcul du flow positif et négatif est effectué :
 - Flow positif : à quel point la solution est préférée par rapport aux autres
 - Flow négatif : comparaisons inverses
- Les solutions sont triées par leur flows nets et celle avec le meilleur flow est choisie

Résultats



La production est de 72,
la distance aux habitations est 46.84 et
le score de compacité 1.68.
Avec une population de 2000 et 3 comme rd.seed.
En 2000 générations.



La production est de 72,
la distance aux habitations est 46.81 et
le score de compacité 0.84.
Avec une population de 2000 et 1 comme rd.seed.
En 2000 générations.

Les poids sont de (0.7 , 0.3 , 0.5)

Discussion et conclusion

Discussion

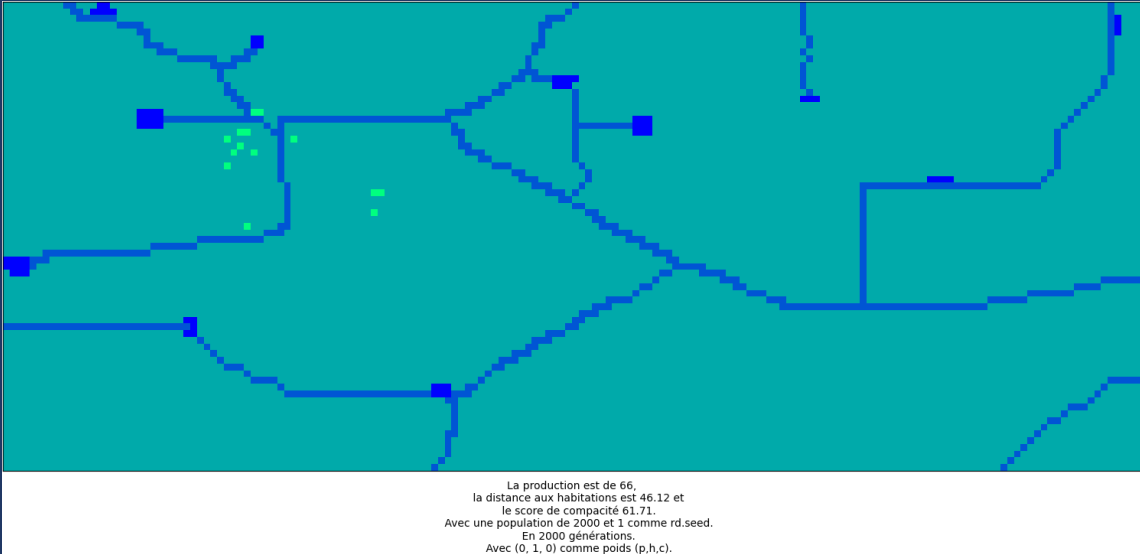
- Convergence géographique des solutions due à la modélisation de la distance des habitations.
- Il est possible qu'une solution converge de manière précoce vers une zone non-optimale. Lorsque cela arrive, les futures générations restent "coincées" dans ce maximum local. Ceci est dû à la nature du problème ainsi qu'à la nature des mutations qui ne permettent pas de déplacer l'entièreté des champs hors du maximum local.

Conclusion

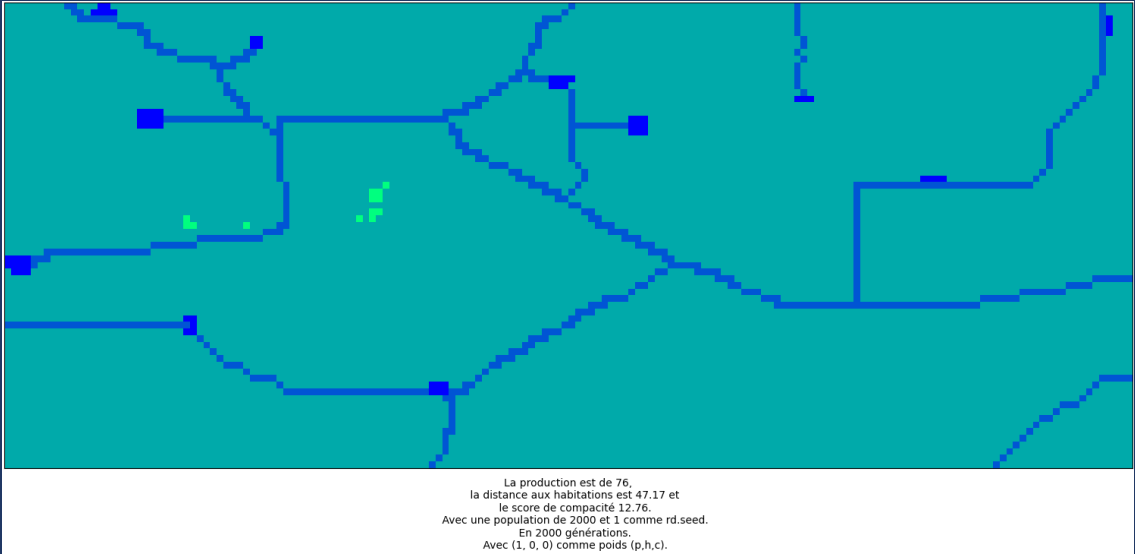
- La fonction de "fitness" permet une bonne sélection de la portion d'élite
- La représentation visuelle de la frontière de Pareto atteste de la qualité des résultats de l'algorithme génétique
- L'algorithme présente une bonne performance grâce à diverses optimisations telle que la mémoisation
- Une convergence relativement constante entre les différents seeds indique un bon fonctionnement. Les résultats obtenus restent consistants et sont en phase avec la modélisation des critères.

Annexes

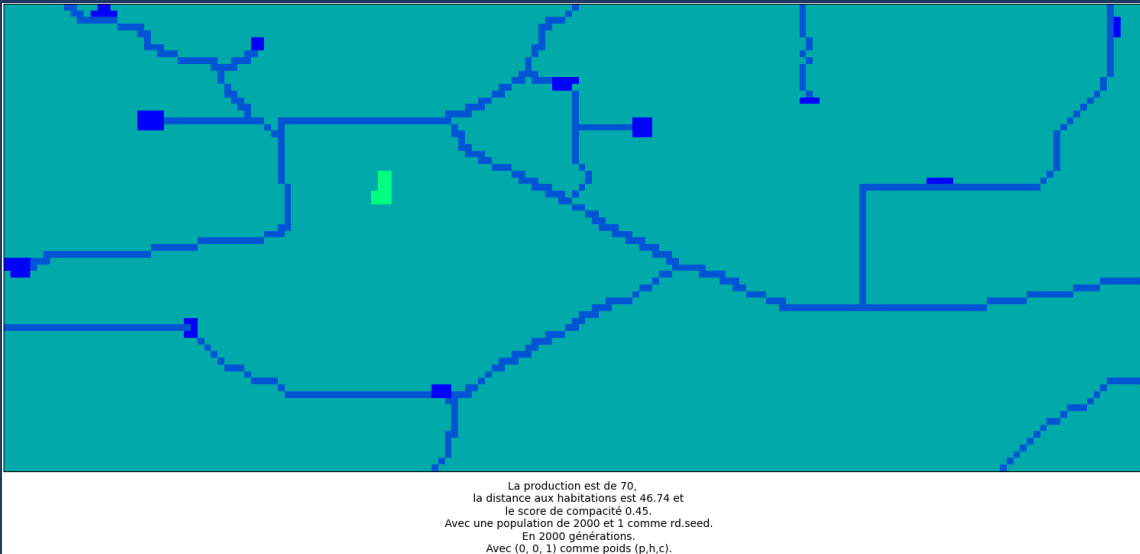
Le même seed a été utilisé pour les 4 résultat



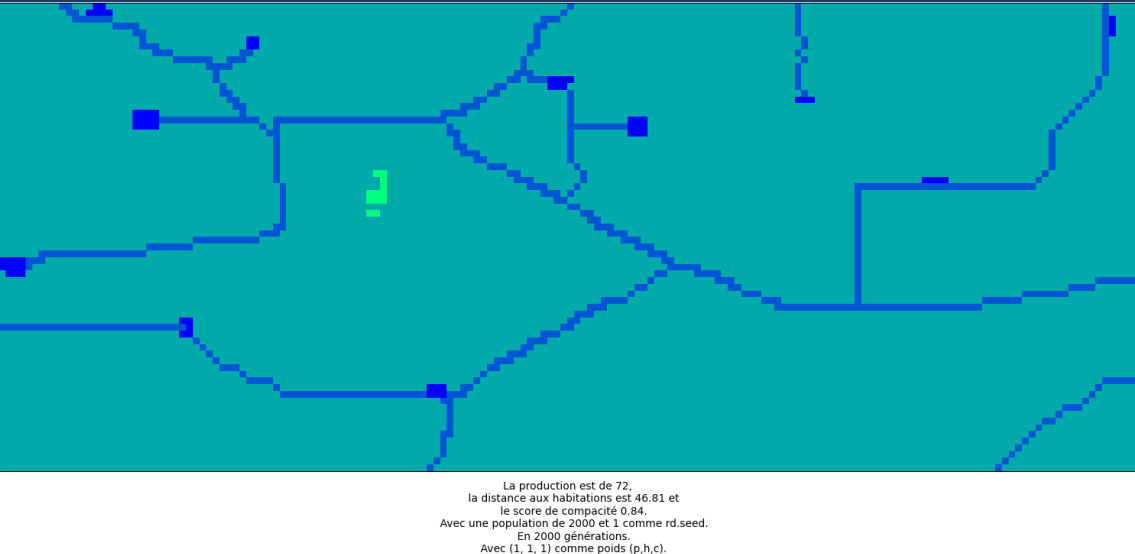
Distance à 1, le reste à 0



Production à 1, le reste à 0



Compacité à 1, le reste à 0



Les trois critères à 1