CESI
ÉCOLE D'INGÉNIEURS

Recherche Opérationnelle

29 Avril 2025

Alban CALVO - Evan JOASSON - Mathéo PINGET

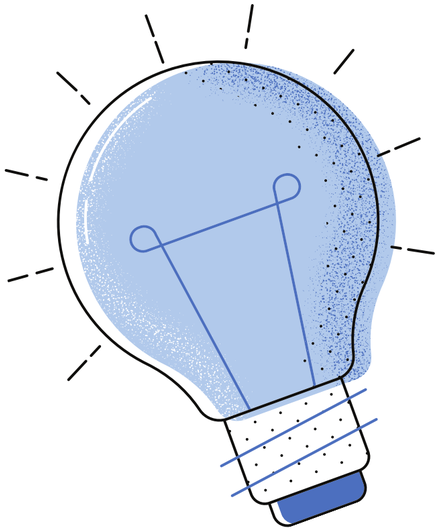
Sommaire

Résumé du problème :

- Mission : L'ADEME à lancé un appel pour promouvoir l'expérimentation de nouvelles solutions de mobilité.

- Objectif :
  - Réduire la consommation d'énergie
  - Réduire les émissions de gaz à effet de serre.

- Comment ?
  - Faire évoluer les comportements
  - Améliorer les modes de transport et leur efficacité



Résoudre un problème d'optimisation ?

- Problème :
  - Limiter les déplacements
  - Limiter la consommation des véhicules lors des livraison
- Problème algorithmique :
  - Relier un sous ensemble de villes entre elles
  - Revenir au point de départ
  - Minimiser la durée de la tournée
- Contraintes :
  - Coût ou restriction de passage sur certaines arêtes : Certaines routes peuvent être plus coûteuses ou interdites (par exemple, travaux ou routes bloquées).
  - Dépendances entre visites : Une ville ne peut être visitée qu'après en avoir visité une autre (par exemple, une livraison doit précéder une collecte).

Contenu de l'étude :

- Modélisation du problème
- Analyse de la complexité du problème
- Code python capable de :
  - Générer des instances aléatoires
  - Résoudre le problème à l'aide d'au moins deux méthodes différentes
    - Dont une méthode exacte
  - Générer des statistiques concernant les performances des algorithmes

Modélisation du problème :

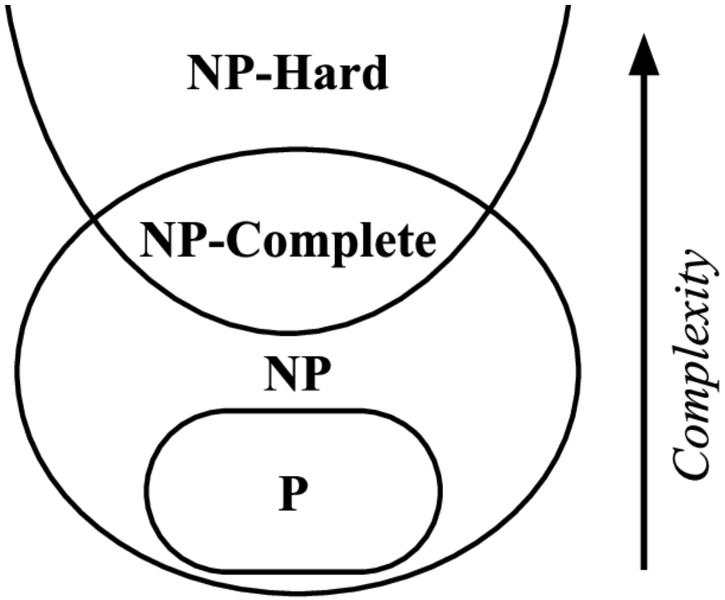
- Réseau routier représenté un graphe G = (S, A)
  - Ensemble des sommets S = {s0, s1,..., sn}
  - Ensemble des arêtes A ⊆ S x S
    - Arêtes (i,j) reliant une ville si à une ville sj
    - Chaque arête à un coût Cij ∈ R+
- Contrainte de dépendance :
  - D ⊆ S x S : (i,j) ∈ D signifie que si doit être visité avant sj
- Variables de décision xij ∈ {0, 1} :
  - 1 si le véhicule se rend de si à sj
  - 0 sinon

Modélisation du problème :

- Fonction objectif : Minimiser le coût total des trajets
  - min $\Sigma_{(i,j)\in A}$ (cij * xij)
  - xij = variable de décision
  - cij = coût de passage
- Départ du dépot et retour :
  - $\Sigma_{j\in S}$ ($xS0j = 1$)
  - $\Sigma_{j\in S}$ ($xiS0 = 1$)
- *Dépendances entre les villes ui < uj*
- *Arêtes impraticables :*
  - $\forall (i, j) \in A', xij = 0$
  - $A' \subset A$

Compléxité du problème :

- Problème de base : Voyageur de Commerce (PVC)
  - + dépendances entre visites
  - + arêtes impraticables
- Espace de recherche :
  - Pour n sommets → espace de permutations = n - 1
  - Ordre de grandeur O(n!)
- De type NP-complet :
  - PVC classique
  - PVC avec précédences
  - PVC avec arêtes impraticables
  - PVC avec contraintes multiples (NP-difficile)
- Appartenance à NP : vérifiable en temps polynomial
  - Vérification des dépendances : O(n)
  - Vérification des arêtes utilisées : $O(n^2)$
  - Calcul des coûts : O(n)



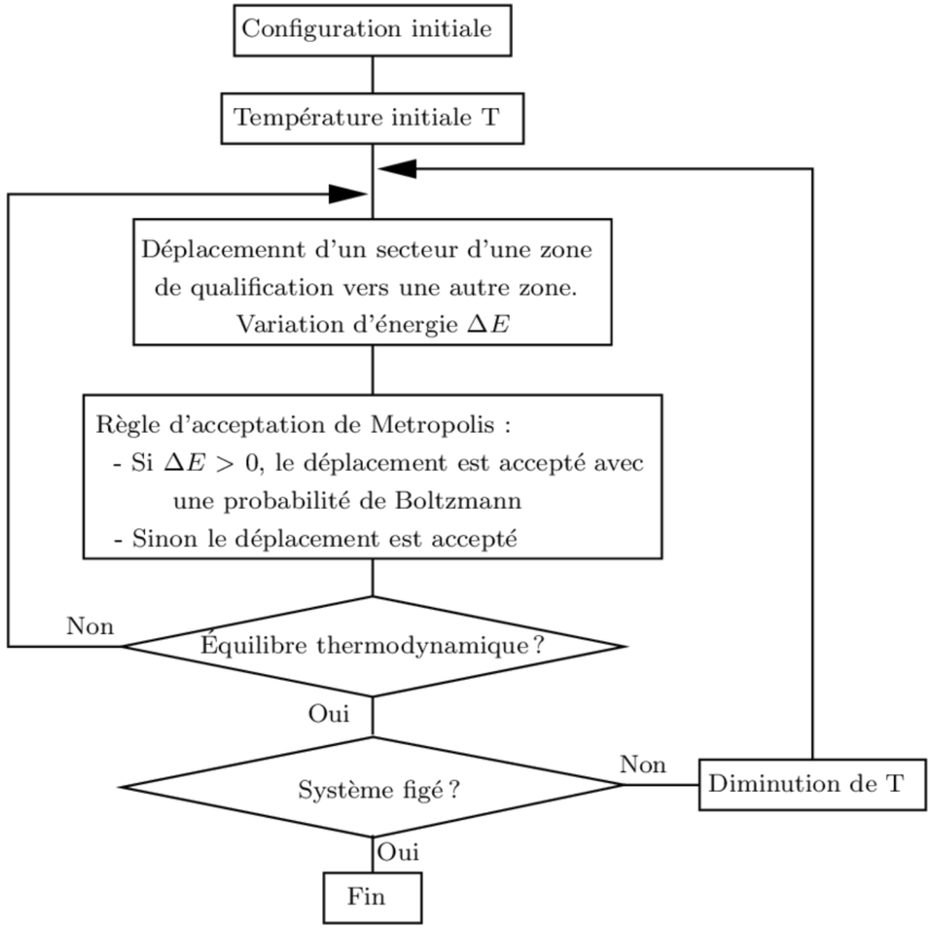
O	Type de complexité
$O(1)$	constant
$O(\log(n))$	logarithmique
$O(n)$	linéaire
$O(n \times \log(n))$	quasi-linéaire
$O(n^2)$	quadratique
$O(n^3)$	cubique
$O(2^n)$	exponentiel
$O(n!)$	factoriel

Méthodes de résolution :

- C'est quoi une métaheurstique ? :
  - Algorithme d'optimisation approximative conçue pour fournir des solutions de bonnes qualité
- Utilisation de métaheuristiques :
  - Justifiée car problème NP-complet
  - Solution exacte qui ne peut pas être trouvée en temps polynomial
  - MAIS peut être vérifiée en temps polynomial
- Quelles métaheuristiques ?
  - Recuit simulé
  - Algorithme génétique
  - Colonie de fourmis
- Comparaison des résultats avec une méthode de résolution exacte (PLNE)

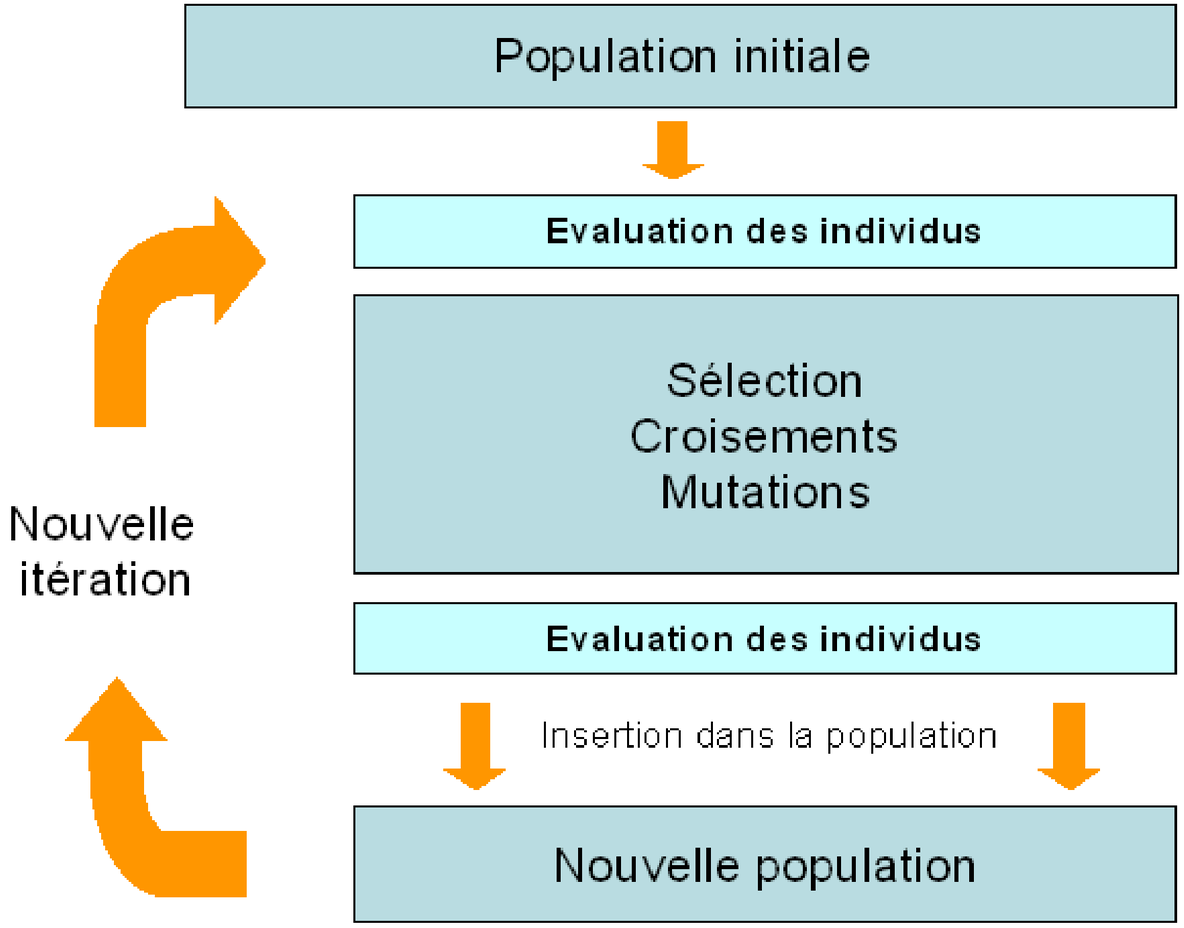
Recuit simulé :

- Définition :
  - Algorithme inspiré du refroidissement des métaux. Cherche un minimum global. Accepte des solutions moins bonnes pour éviter les optima locaux
- Principe :
  - Part d'une solution initiale
  - Explore le voisinage
  - Accepte ou refuse les solutions selon la température
  - Plus la température baisse plus l'algorithme devient strict
- Paramètres :
  - Température initiale
  - Taux de refroidissement
  - Température minimale
  - Nb max d'itérations



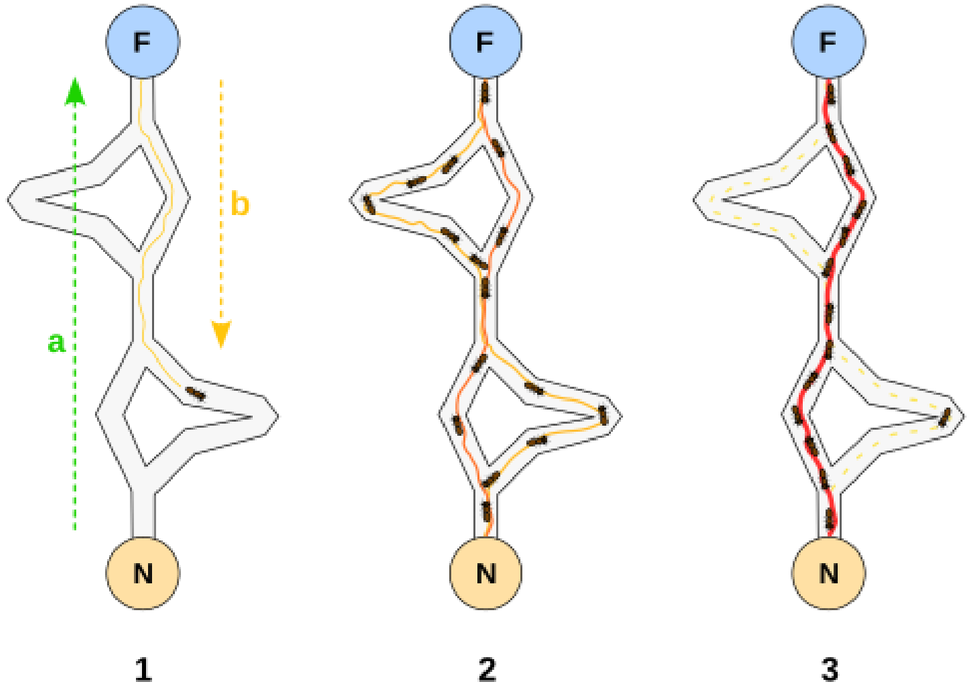
Algorithme génétique :

- Définition
  - Méthode d'optimisation inspirée de l'évolution naturelle, appliquée à une population de solutions potentielles pour résoudre un problème donné.
- Principe de fonctionnement
  a. Création d'une population initiale de solutions (généralement aléatoire)
  b. Évaluation de la qualité de chaque solution (fonction d'évaluation)
  c. Sélection des meilleures solutions
  d. Croisement et mutation pour générer une nouvelle population
  e. Répétition du processus jusqu'à obtenir une solution satisfaisante
- Paramètres clés
  - Taille de la population : nombre de solutions simultanément considérées
  - Nombre de générations : nombre d'itérations de l'algorithme
  - Taux de mutation : probabilité de modifier une solution
  - Nombre d'individus sélectionnés : proportion des meilleures solutions conservées pour générer la prochaine génération



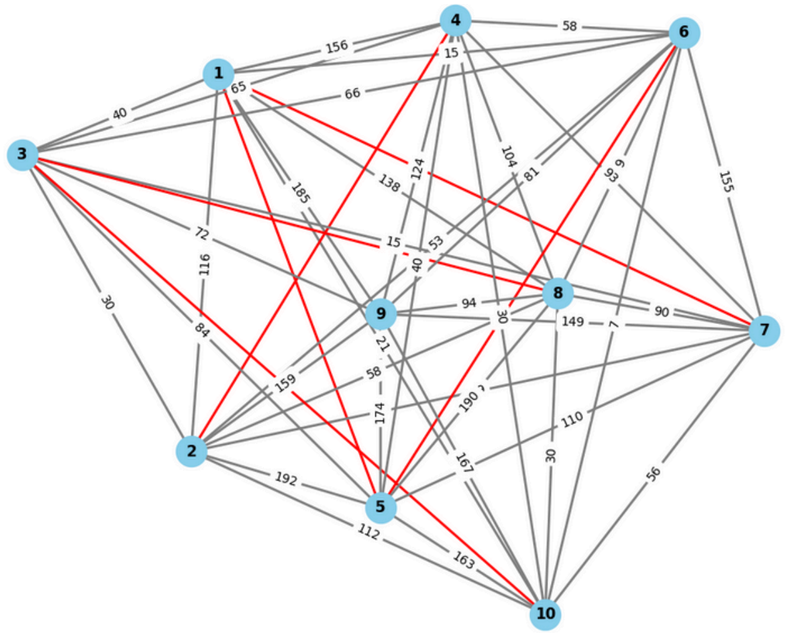
Colonie de fourmis :

- Définition :
  - Algorithme inspiré du comportement des fourmis cherchant un chemin optimal en déposant des traces de phéromones.
- Principe :
  - Des agents (fourmis) explorent les chemins possibles
  - Elles déposent des phéromones sur les bons chemins
  - Plus un chemin est utilisé plus il est attractif
  - Evaporation des phéromones pour ne pas bloquer sur un mauvais choix
- Paramètres :
  - nb fourmis
  - nb itérations
  - alpha : phéromones
  - béta : influence de la distance
  - taux d'évaporation
  - Q : intensité du dépôt de phéromones



Résolution du problème :

- Génération d'une matrice d'adjacence aléatoire → graphe connexe complet :
  - Arbre couvrant
  - Remplissage aléatoire
  - Taux de routes en travaux (-1)
  - Points de collecte (2 sommets choisis aléatoirement hors sommet 1)
  - n sommets = 100
- Pondération des arêtes :
  - Coût max = 200
  - -1 = arête impraticable
  - 0 arête inexistante
  - *X coût d'une arête*
- *2 sommets sont des points de collecte pour la contrainte de dépendance*
- *Export de la matrice dans un fichier CSV et génération du graphe*




```
Points collectes: [6, 8]
[
    [0, 116, 40, 156, -1, 15, -1, 138, 185, 21],
    [116, 0, 30, -1, 192, 53, 29, 58, 159, 112],
    [40, 30, 0, 65, 84, 66, 15, -1, 72, -1],
    [156, -1, 65, 0, 40, 58, 93, 104, 124, 30],
    [-1, 192, 84, 40, 0, -1, 110, 190, 174, 163],
    [15, 53, 66, 58, -1, 0, 155, 9, 81, 7],
    [-1, 29, 15, 93, 110, 155, 0, 90, 149, 56],
    [138, 58, -1, 104, 190, 9, 90, 0, 94, 30],
    [185, 159, 72, 124, 174, 81, 149, 94, 0, 167],
    [21, 112, -1, 30, 163, 7, 56, 30, 167, 0],
]
```

Comparaison des métaheuristiques :

```
================================================================================
COMPARATIVE ANALYSIS FOR MATRIX SIZE 10
================================================================================

+--------------------+----------+--------------+--------------+--------+----------+--------------+--------------+---------+
|      Algorithm     | Min Cost |   Avg Cost   |   Med Cost   | Cost % | Min Time |   Avg Time   |   Med Time   | Time %  |
+--------------------+----------+--------------+--------------+--------+----------+--------------+--------------+---------+
|      Genetic       |  441.0   | 483.2 ± 27.8 | 479.0 ± 30.0 |  0.0%  |   0.20   | 0.21 ± 0.01  | 0.21 ± 0.01  | 548.8%  |
| Simulated Annealing |  495.0   | 555.8 ± 38.6 | 571.0 ± 27.0 | 12.2%  |   0.03   | 0.03 ± 0.00  | 0.03 ± 0.00  |  0.0%   |
|     Ant Colony     |  465.0   | 476.2 ± 12.2 | 469.0 ± 4.0  |  5.4%  |   0.69   | 0.73 ± 0.03  | 0.71 ± 0.02  | 2165.7% |
|        PLNE        |  441.0   | 441.0 ± 0.0  | 441.0 ± 0.0  |  0.0%  |   0.19   | 0.23 ± 0.08  | 0.19 ± 0.01  | 505.1%  |
+--------------------+----------+--------------+--------------+--------+----------+--------------+--------------+---------+
```

```
================================================================================
COMPARATIVE ANALYSIS FOR MATRIX SIZE 20
================================================================================

+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Algorithm     | Min Cost |    Avg Cost   |    Med Cost   | Cost % | Min Time |   Avg Time   |   Med Time   |  Time %  |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Genetic       |  1080.0  | 1104.4 ± 27.7 | 1091.0 ± 11.0 | 36.7%  |   0.09   | 0.10 ± 0.00  | 0.10 ± 0.00  | 9149.3%  |
| Simulated Annealing |  1730.0  | 1865.8 ± 102.3 | 1863.0 ± 94.0 | 119.0% |   0.00   | 0.00 ± 0.00  | 0.00 ± 0.00  |  0.0%    |
|     Ant Colony     |  874.0   | 911.2 ± 30.6  | 897.0 ± 23.0  | 10.6%  |   0.21   | 0.22 ± 0.01  | 0.23 ± 0.01  | 21097.8% |
|        PLNE        |  790.0   | 790.0 ± 0.0   | 790.0 ± 0.0   |  0.0%  |   0.28   | 0.29 ± 0.01  | 0.29 ± 0.01  | 27950.1% |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
```

```
================================================================================
COMPARATIVE ANALYSIS FOR MATRIX SIZE 30
================================================================================

+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Algorithm     | Min Cost |    Avg Cost   |    Med Cost   | Cost % | Min Time |   Avg Time   |   Med Time   |  Time %  |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Genetic       |  886.0   | 1074.8 ± 120.7 | 1075.0 ± 97.0 | 56.0%  |   0.34   | 0.34 ± 0.00  | 0.34 ± 0.00  | 11099.7% |
| Simulated Annealing |  1906.0  | 2019.4 ± 94.4 | 2026.0 ± 104.0 | 235.6% |   0.00   | 0.00 ± 0.00  | 0.00 ± 0.00  |  0.0%    |
|     Ant Colony     |  776.0   | 842.0 ± 39.4  | 849.0 ± 19.0  | 36.6%  |   0.91   | 0.95 ± 0.02  | 0.96 ± 0.01  | 30368.0% |
|        PLNE        |  568.0   | 568.0 ± 0.0   | 568.0 ± 0.0   |  0.0%  |   1.02   | 1.04 ± 0.02  | 1.04 ± 0.01  | 33827.5% |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
```

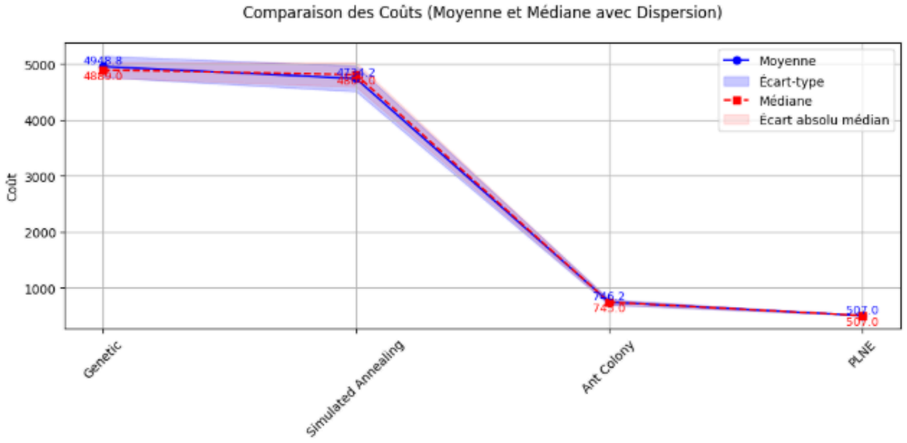
```
================================================================================
COMPARATIVE ANALYSIS FOR MATRIX SIZE 50
================================================================================

+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Algorithm     | Min Cost |    Avg Cost   |    Med Cost   | Cost % | Min Time |   Avg Time   |   Med Time   |  Time %  |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Genetic       |  1139.0  | 1288.2 ± 83.8 | 1312.0 ± 50.0 | 113.3% |   2.18   | 2.22 ± 0.04  | 2.20 ± 0.03  | 36180.3% |
| Simulated Annealing |  2800.0  | 3419.6 ± 424.7 | 3501.0 ± 301.0 | 424.3% |   0.01   | 0.01 ± 0.00  | 0.01 ± 0.00  |  0.0%    |
|     Ant Colony     |  675.0   | 712.0 ± 34.0  | 693.0 ± 18.0  | 26.4%  |   4.27   | 4.34 ± 0.04  | 4.36 ± 0.02  | 70985.1% |
|        PLNE        |  534.0   | 534.0 ± 0.0   | 534.0 ± 0.0   |  0.0%  |  11.39   | 11.50 ± 0.08 | 11.48 ± 0.07 | 189708.8% |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
```

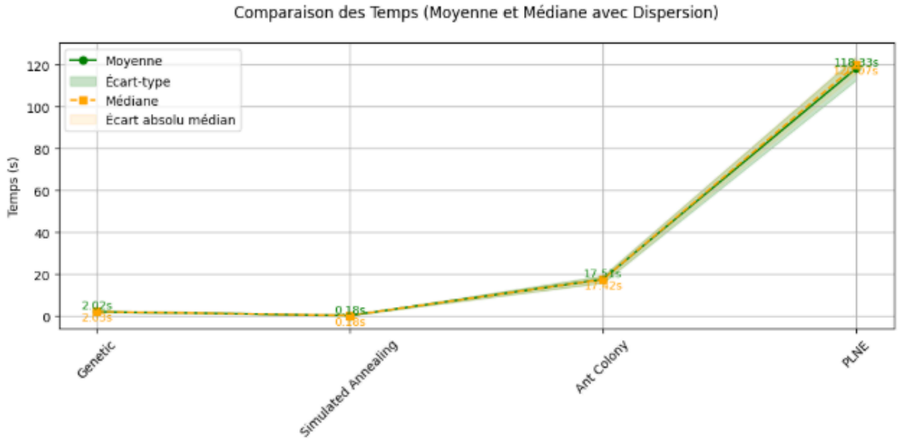
```
================================================================================
COMPARATIVE ANALYSIS FOR MATRIX SIZE 75
================================================================================

+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Algorithm     | Min Cost |    Avg Cost   |    Med Cost   | Cost % | Min Time |   Avg Time   |   Med Time   |  Time %  |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Genetic       |  3114.0  | 3330.4 ± 246.2 | 3202.0 ± 88.0 | 374.7% |   1.40   | 1.46 ± 0.07  | 1.41 ± 0.01  |  936.5%  |
| Simulated Annealing |  3353.0  | 3613.2 ± 190.1 | 3674.0 ± 211.0 | 411.1% |   0.13   | 0.14 ± 0.01  | 0.14 ± 0.00  |  0.0%    |
|     Ant Colony     |  790.0   | 843.4 ± 36.6  | 850.0 ± 34.0  | 20.4%  |  10.60   | 11.20 ± 0.50 | 11.36 ± 0.52 | 7760.2%  |
|        PLNE        |  656.0   | 656.0 ± 0.0   | 656.0 ± 0.0   |  0.0%  |  17.92   | 18.87 ± 0.65 | 18.81 ± 0.73 | 13187.0% |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
```

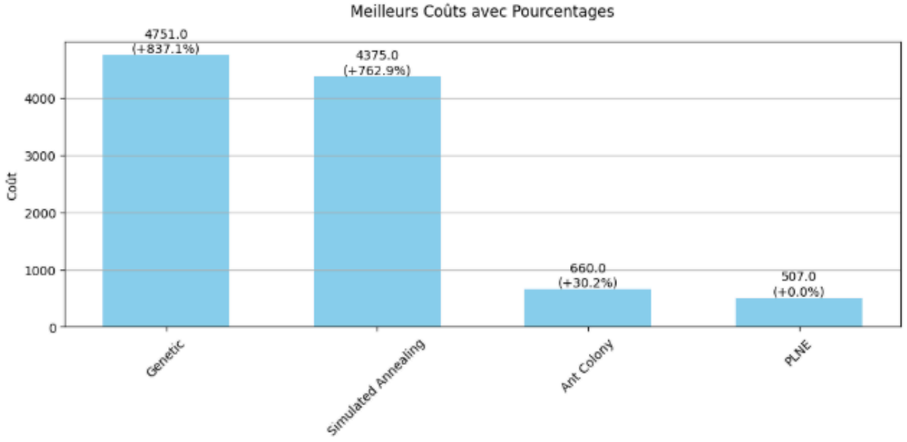
```
================================================================================
COMPARATIVE ANALYSIS FOR MATRIX SIZE 100
================================================================================

+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Algorithm     | Min Cost |    Avg Cost   |    Med Cost   | Cost % | Min Time |   Avg Time   |   Med Time   |  Time %  |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
|      Genetic       |  2178.0  | 2267.2 ± 86.1 | 2237.0 ± 59.0 | 329.6% |  16.32   | 16.57 ± 0.15 | 16.63 ± 0.12 | 74077.9% |
| Simulated Annealing |  6901.0  | 7317.0 ± 302.3 | 7328.0 ± 261.0 | 1261.1% |   0.02   | 0.02 ± 0.00  | 0.02 ± 0.00  |  0.0%    |
|     Ant Colony     |  707.0   | 725.6 ± 20.7  | 718.0 ± 2.0   | 39.4%  |  32.53   | 32.74 ± 0.21 | 32.68 ± 0.14 | 147768.7% |
|        PLNE        |  507.0   | 507.0 ± 0.0   | 507.0 ± 0.0   |  0.0%  |  80.96   | 81.81 ± 0.82 | 81.45 ± 0.49 | 367961.7% |
+--------------------+----------+---------------+---------------+--------+----------+--------------+--------------+----------+
```

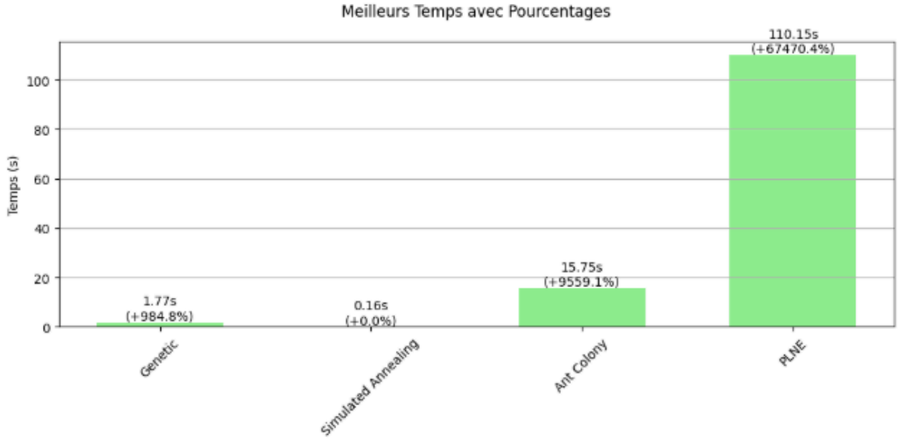
Comparaison des métaheuristiques :

Test sur une matrice 100







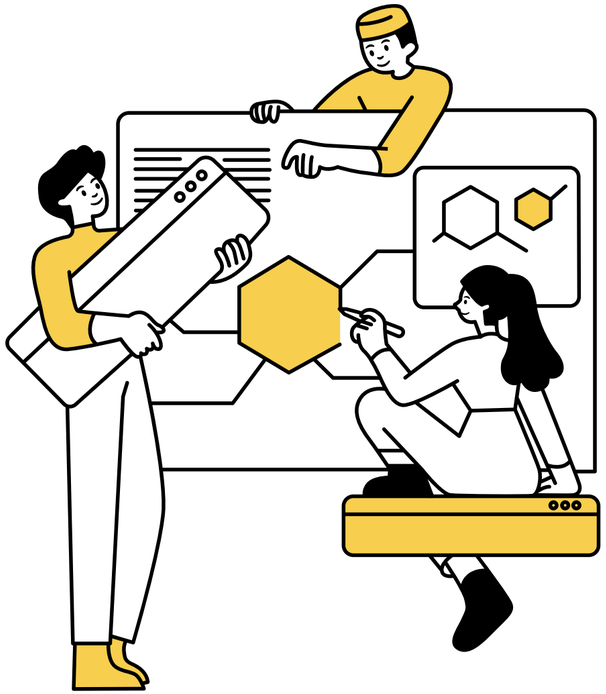



```
BEST SOLUTIONS FOUND:

Genetic:
Cost: 2178.0
Path: [1, 46, 56, '...', 23, 85, 1]

Simulated Annealing:
Cost: 6901.0
Path: [1, 46, 56, '...', 96, 47, 1]

Ant Colony:
Cost: 707.0
Path: [1, 46, 95, '...', 17, 64, 1]

PLNE:
Cost: 507.0
Path: [1, 46, 95, '...', 93, 64, 1]
```

Perspectives d'amélioration :

- Avoir plusieurs arêtes entre chaque sommet au lieu d'une seule
- Augmenter le nombre de contraintes pour rendre le problème plus réaliste :
  - Capacité du camion
  - Nb camions
  - Fenêtres temporelles
- Tester d'autres algorithmes :
  - GRASP (Greedy Randomized Adaptative Search Produce)
  - TABU

Conclusion



Bibliographie

- https://github.com/gregory-chatelier/tsp
- https://www.malaspinas.academy/prog_seq/exercices/09_voyageur_commerce/index.html
- https://igm.univ-mlv.fr/~dr/XPOSE2013/tleroux_genetic_algorithm/fonctionnement.html
- https://www.i2m.univ-amu.fr/perso/jean-philippe.preaux/PDF/pdf_proteges/OptimisationCombinatoire/Metaheuristiques2.pdf
- http://www.lps.ens.fr/~weisbuch/livre/b9.html
- https://webusers.i3s.unice.fr/~crescenz/publications/travaux_etude/colonies_fourmis-200605-rapport.pdf
- https://members.loria.fr/VThomas/enseignement/M2rar/slides/02-metaheuristiques_individuelles.pdf