

Rapport de Projet Informatique

Projet Tournée de livraisons avec Simulation par Colonie de Fourmis

L'objectif dans ce projet est d'écrire un programme permettant de résoudre une tournée de livraison en utilisant la méthode de résolution par simulation de colonies de fourmis.

Formules mathématiques utilisées :

Afin de calculer la probabilité pour qu'une fourmi empreinte un chemin plutôt qu'un autre, j'ai implémenté et utilisé la formule suivante :

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in N_i^k(t)} [\tau_{il}(t)]^\alpha \cdot \eta_{il}^\beta} & \text{si } j \in N_i^k \\ 0 & \text{sinon} \end{cases} \quad (1)$$

Dans cette formule :

- p est la probabilité de choisir un chemin ;
- k désigne la fourmi ;
- i désigne la ville où se situe la fourmi ;
- j désigne les possibles villes où peut aller la fourmi ;
- τ est la quantité de phéromone déposée entre les villes i et j ;

- η désigne la visibilité de la ville (moins la distance entre les deux villes sera grande, plus cette valeur sera élevée) : $\eta_{ij} = \frac{1}{d_{ij}}$
- α et β sont des paramètres permettant d'augmenter ou de diminuer l'importance d'une des coefficients τ ou η .

J'utilise ensuite la formule ci-dessous afin de calculer la dispersion des phéromones dans le temps :

$$\tau_{ij}(t + n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

Dans cette formule :

- τ désigne toujours la quantité de phéromone déposé entre les villes i et j ;
- i désigne la ville où se situe la fourmi ;
- j désigne les possibles villes où peut aller la fourmi ;
- ρ est un paramètre permettant de définir la vitesse à laquelle les phéromones s'évaporent.

Enfin, j'utilise la formule ci-dessous afin de déterminer la quantité de phéromone que déposera une fourmi sur son chemin :

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k(t) & \text{si } (u \in T_k(t) \wedge u = (i, j)) \\ 0 & \text{sinon} \end{cases}$$

- τ désigne toujours la quantité de phéromone déposé entre les villes i et j ;
- i désigne la ville où se situe la fourmi ;
- j désigne les possibles villes où peut aller la fourmi ;
- L est la longueur parcourue ;
- Q est une constante ;
- T est le tour effectué par la fourmi ;
- u est le nom donné aux différents chemins possibles.

Fonctionnement de l'algorithme :

On commence par lire le fichier d'entrée qui est composé du nombre de points ainsi que de la distance entre chacun de ces points. On rentre ces données dans un tableau nommé Distance avant de générer une matrice symétrique et où la diagonale vaut 0.

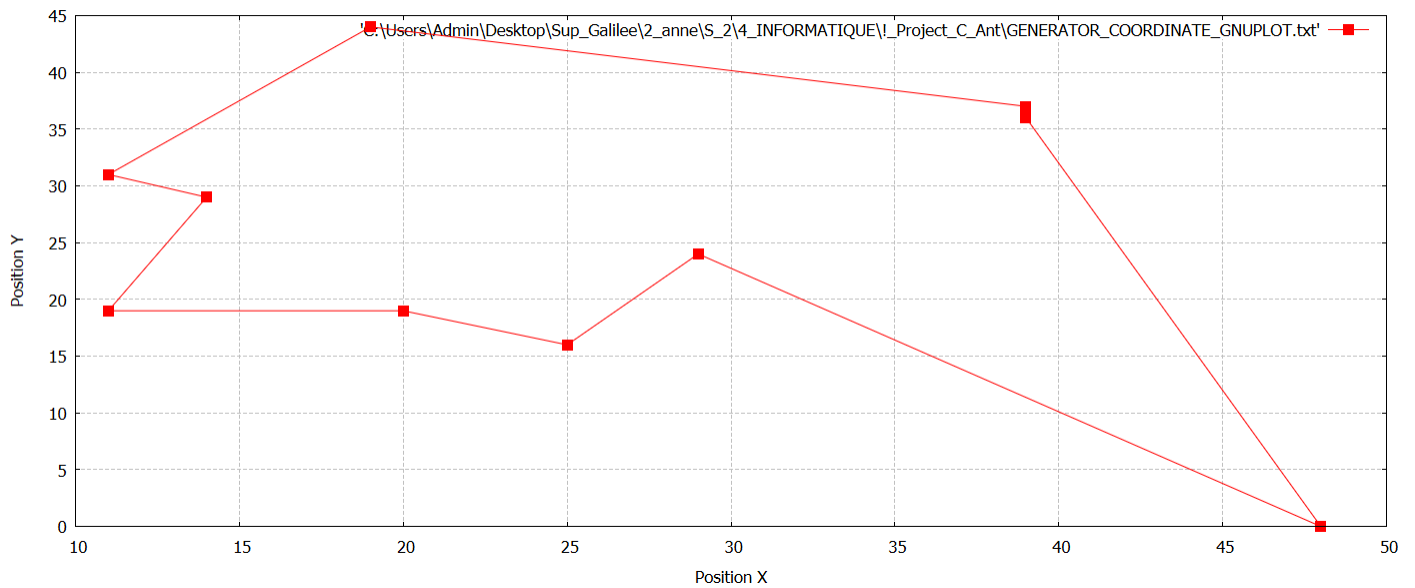
Je fais partir toutes les fourmis sur un point différent, il y a donc autant de fourmis qu'il y a de points. Chaque probabilité pour que la fourmi se déplace vers un point est calculée avec la première formule. Toutes les probabilités sont ensuite placées sur une échelle de 0 à 1 et à l'aide d'un curseur, je vais choisir la destination de la fourmi. Ici plus la probabilité qu'une fourmi se déplace sur un point, plus la zone occupée par cette probabilité sur l'échelle est grande. Une fois le chemin choisi, on y déplace la fourmi et on recommence le processus tout en enlevant les chemins déjà visités.

Après que les fourmis aient effectué leur tour, on y dépose les phéromones en utilisant la troisième formule. Au fur et à mesure des itérations, j'utilise la deuxième formule pour calculer la dispersion des phéromones.

Ce programme permet de visualiser les chemins pris par les fourmis, la distance totale que cela représente et en second temps le taux de phéromone déposé sur la matrice symétrique pour chaque itération.

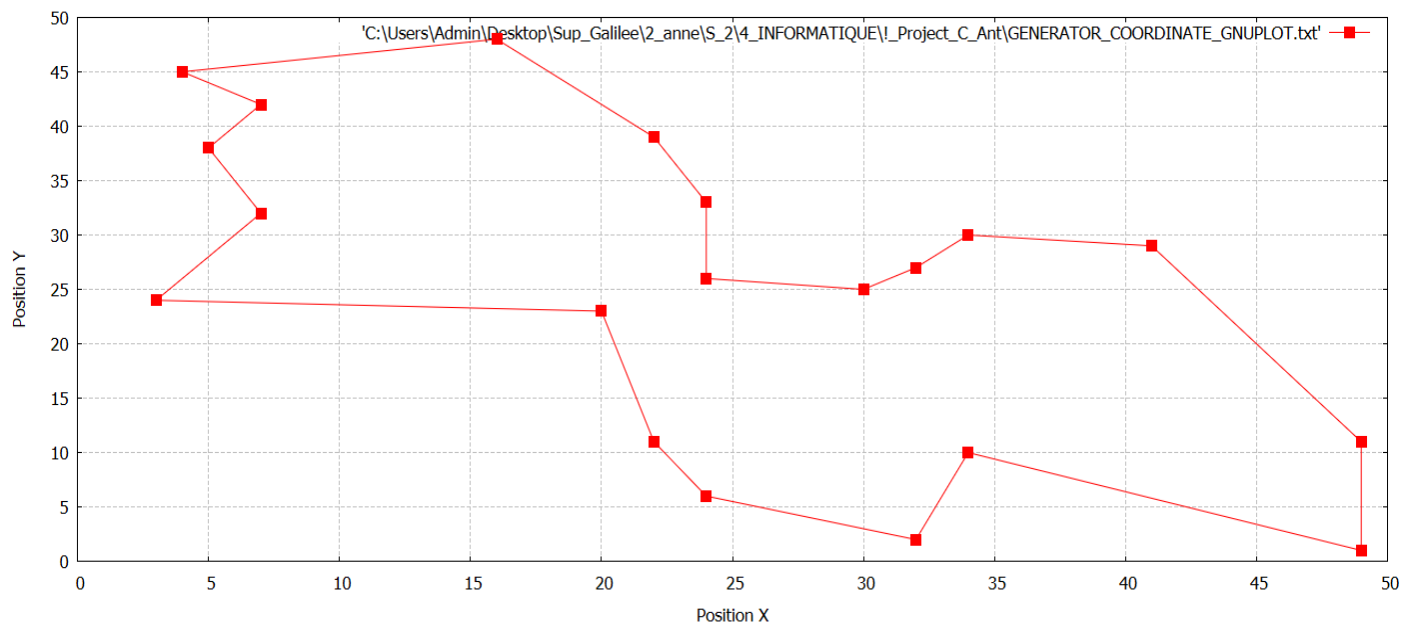
De plus, à l'aide de **Gnuplot**, on peut visualiser le meilleur chemin obtenu et dont voici quelques exemple :

ANT Route



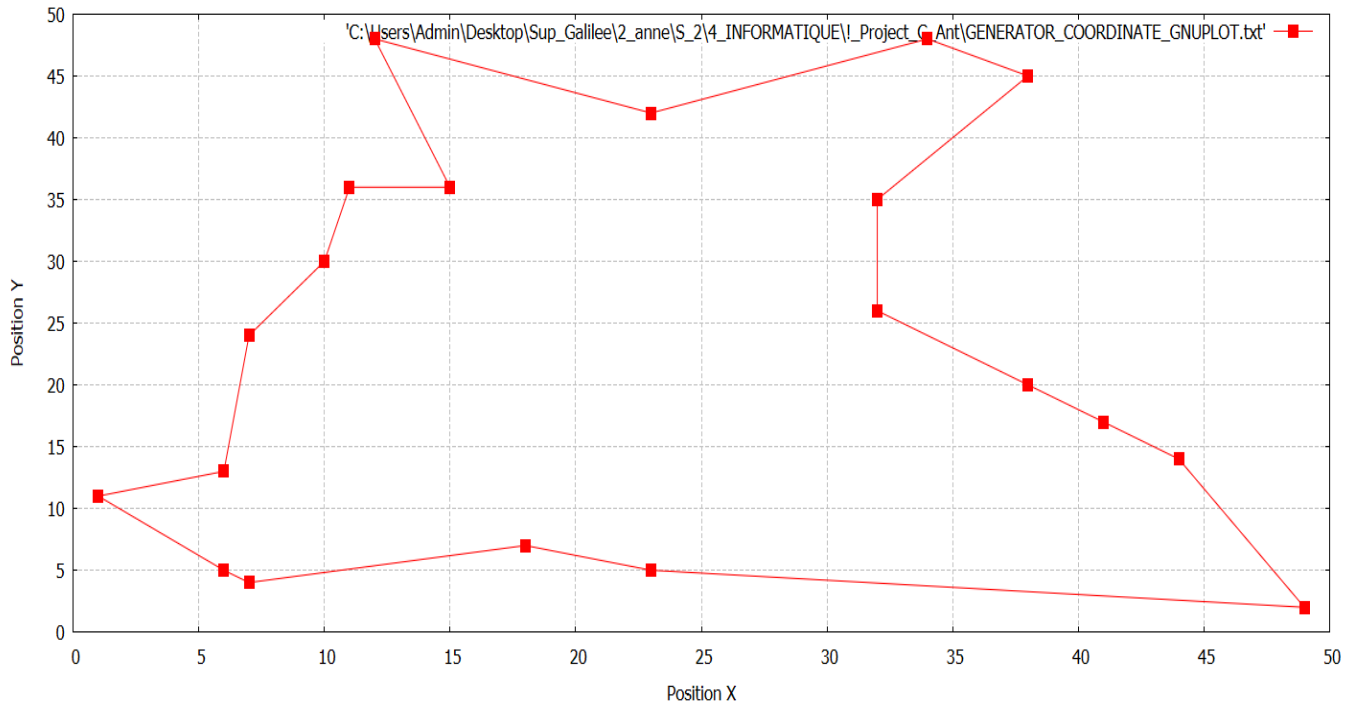
10 points en 5 itérations

ANT Route



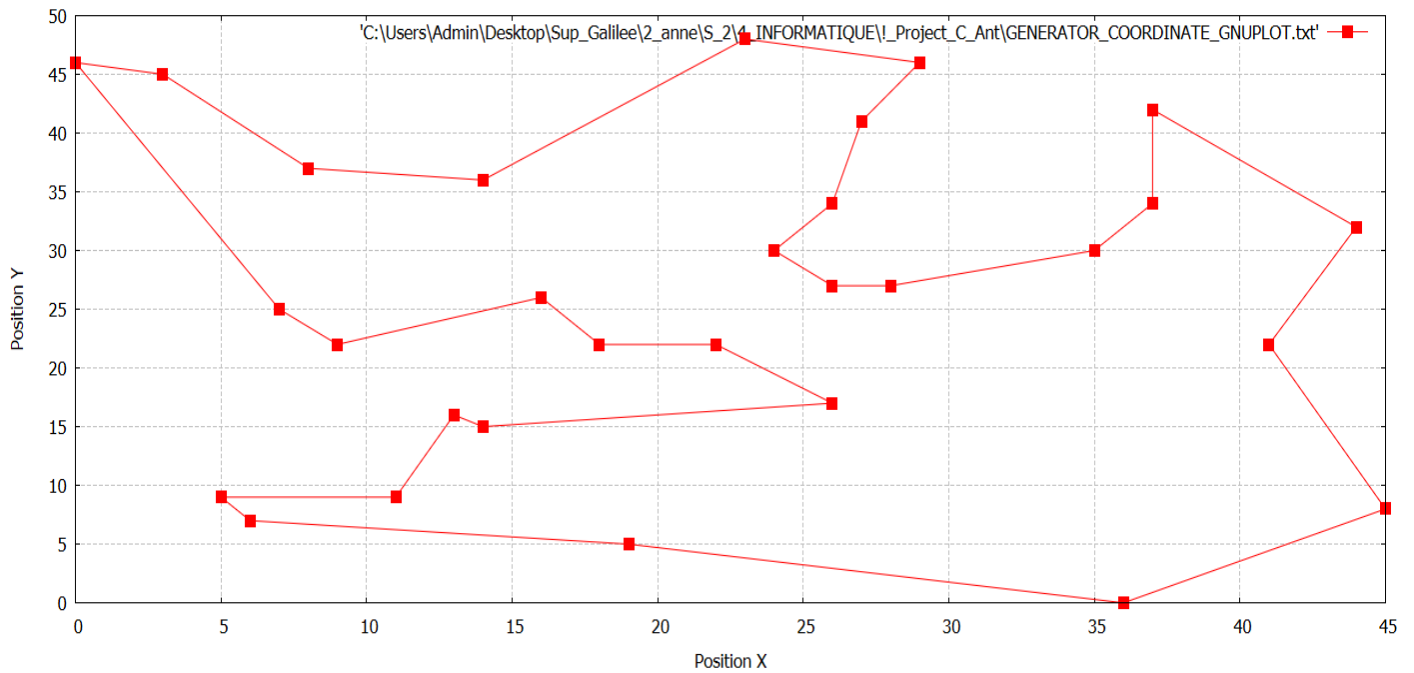
20 points en 7 itérations

ANT Route



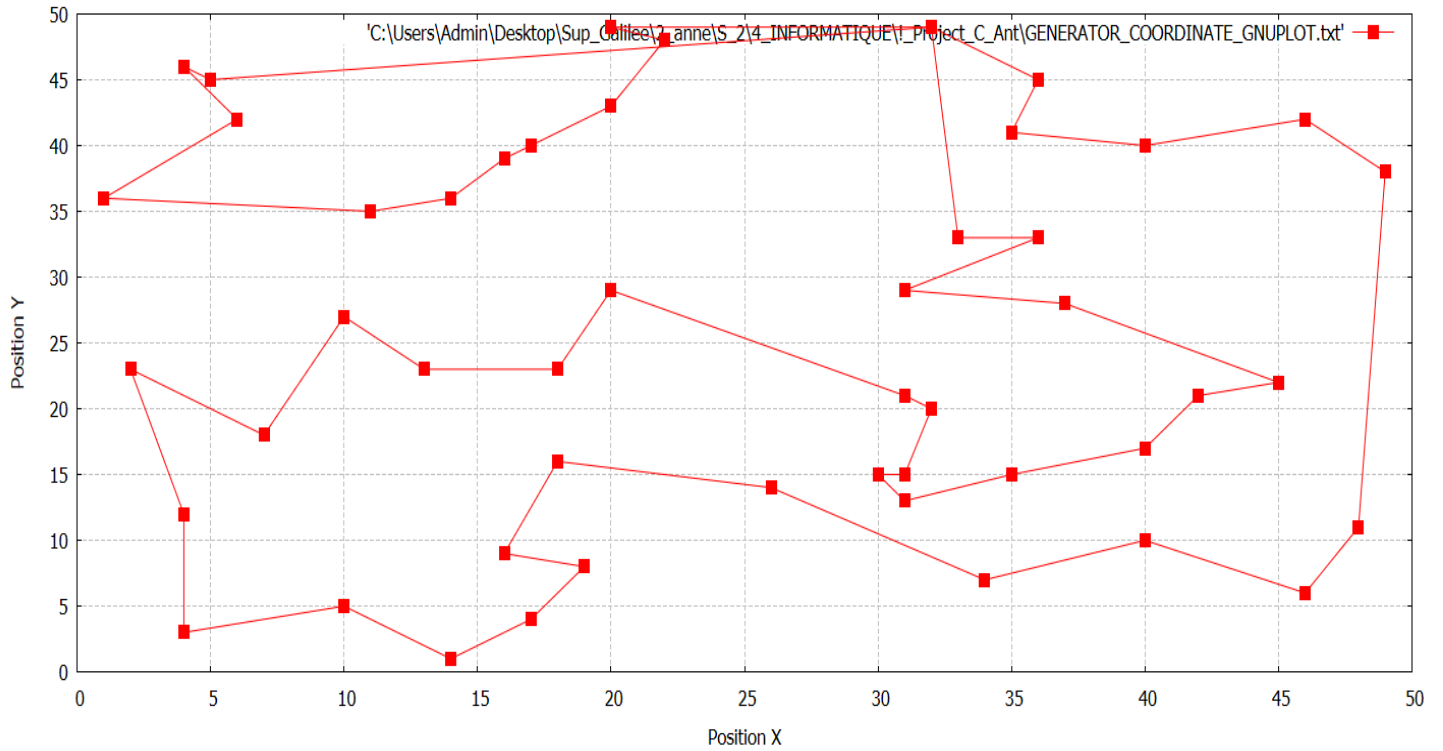
20 points en 20 itérations

ANT Route



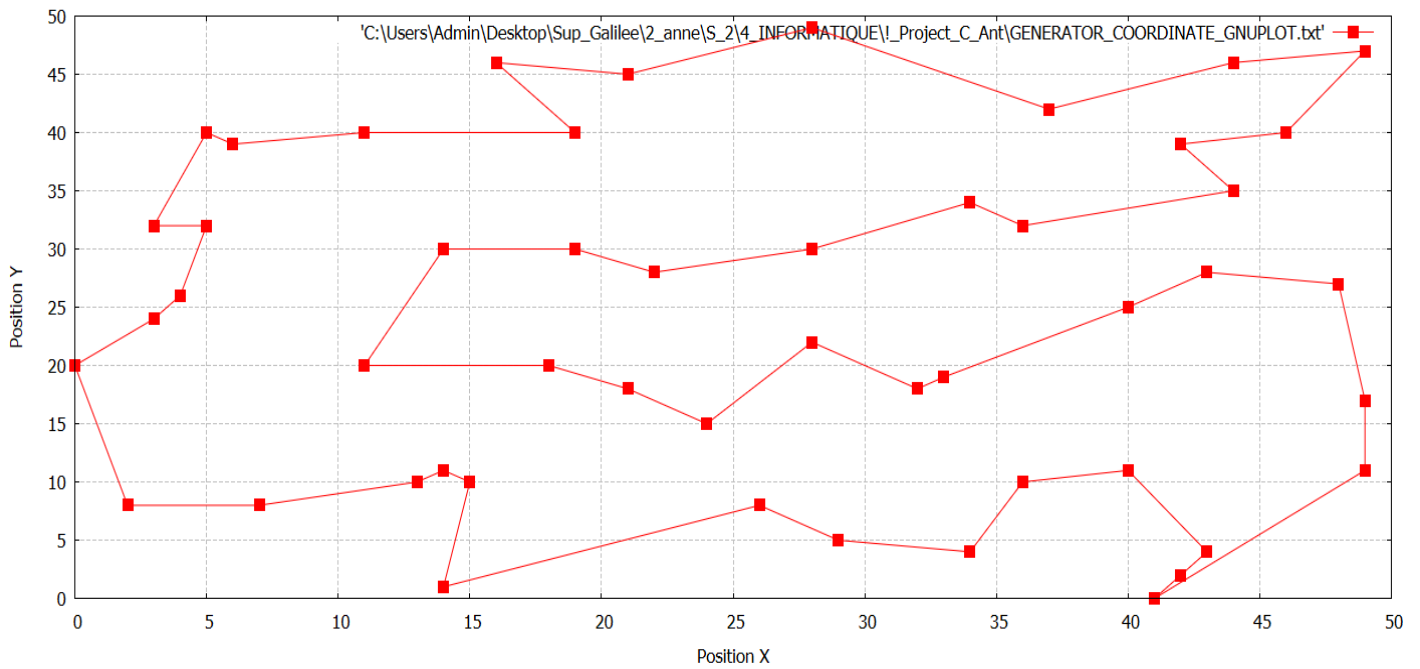
30 points en 10 itérations

ANT Route



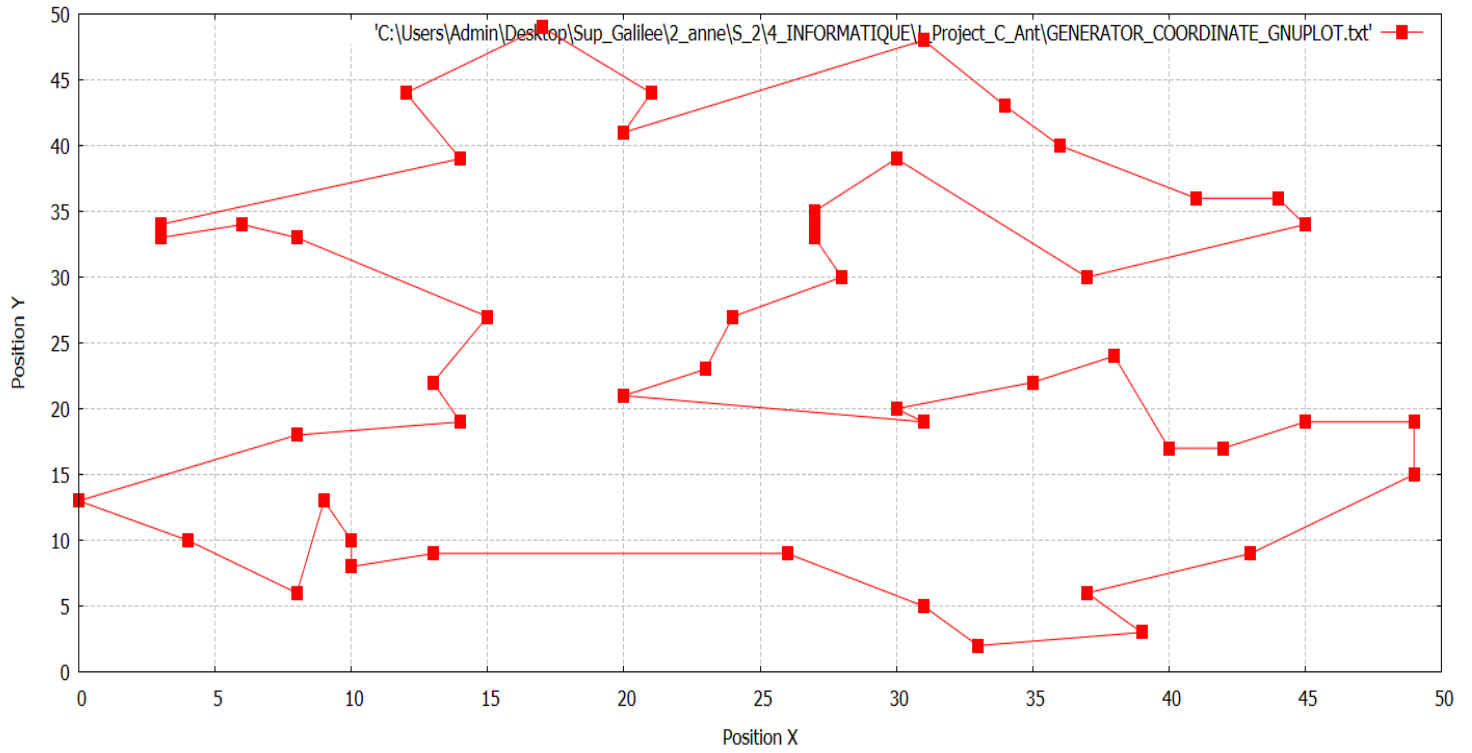
50 points en 5 itérations

ANT Route



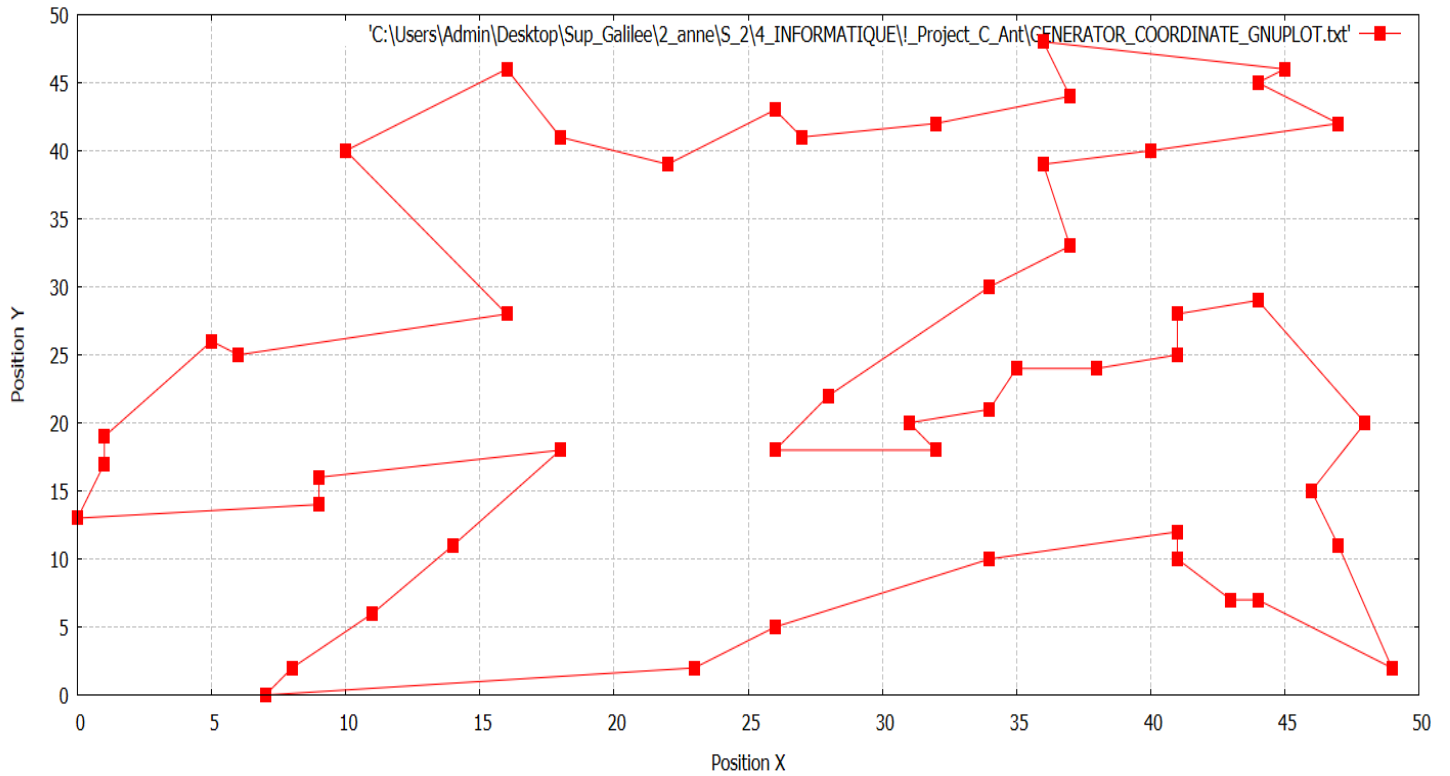
50 points en 50 itérations

ANT Route



50 points en 500 itérations

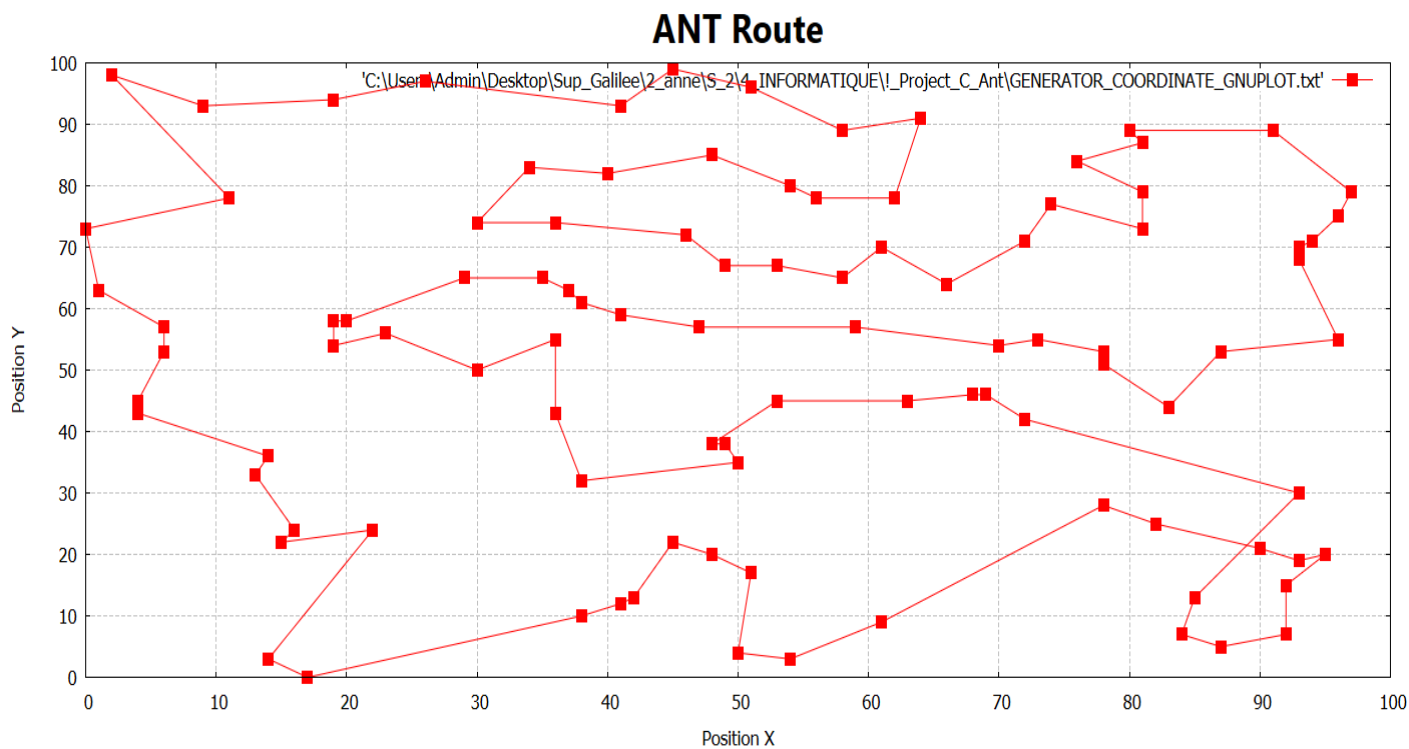
ANT Route



50 points en 1000 itérations

Remarque : Ici les tests sont effectués sur 50 points à de nouvelles positions à chaque fois puisque nous générons aléatoirement des points, plus précisément, leurs coordonnées.

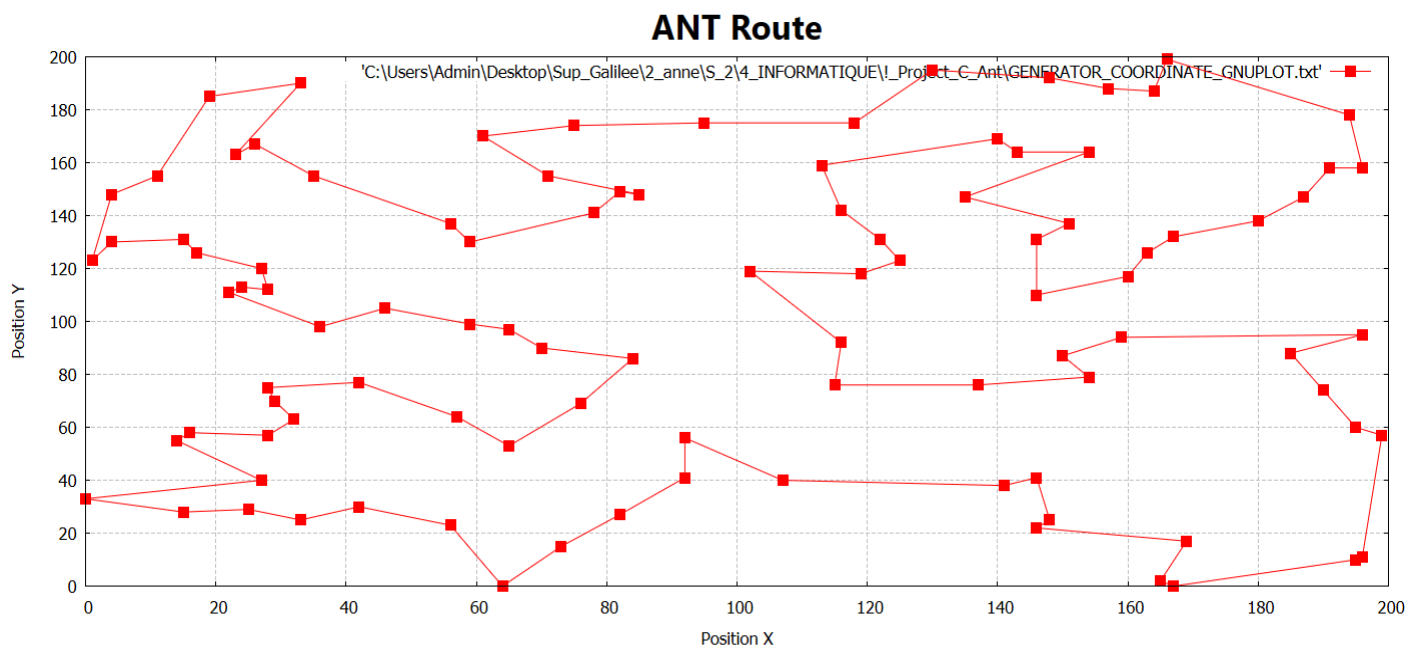
De plus, on peut évidemment réaliser ces tests pour différents nombres de points.



100 points en 200 itérations

Remarque : Sur cette dernière image, j'ai un résultat presque parfait, on observe notamment une petite boucle en bas à droite de l'image.

Cependant, ce problème se résout avec quelques itérations en plus.



100 points en 500 itérations

Fichiers des programmes et fichiers sortants :

Dans chacun des programmes, on revoit 3 fichiers sortants afin d'afficher les résultats.

Le fichier " **INTERMEDIATE_RESULTS_F.txt** " permet de visualiser les chemins et distances parcourus par les fourmis ainsi que la répartition des phéromones sur la matrice des distances pour chaque itération. On retrouve les distances minimales de chaque itération dans le fichier " **RESULTS_MIN_ROUTES_F.txt** ". On retrouve le plus petit chemin trouvé dans le fichier " **RESULT_FINAL_ROUTE_F.txt** ".

Les fichiers " **!_Compile_Instruction_Function_File.txt** " donnent les commandes permettant de compiler chacun des 3 programmes.

Le premier est le programme composé d'un fichier " **Function_Ant.h** " pour la déclaration des fonctions.

Un fichier " **Function_Ant.c** " pour la définitions des fonctions et ces fichiers sont utilisés dans le fichier " **Function_Ant_Roulette_Probability_File.c** " qui est la fonction principale.

De plus, notre programme fonctionne bien pour :

- nombre de ville : **100**
- nombre d'itération : **4500**

Gnuplot :

Un deuxième fichier est disponible contenant le premier programme et permettant de visualiser la plus petite route empruntée par une fourmi à l'aide de Gnuplot. Ce dernier n'est composé que d'une fonction main comprenant toutes les fonctions utilisées.

L'image de Gnuplot est affichée directement à la fin du programme. Les fichiers sortant possèdent le suffixe generator.

De plus, le premier fichier sortant débute avec la création des coordonnées, le calcul des distances et la mise en page comme le fichier d'entrée du début permettant ainsi d'utiliser le programme de base.

Cependant ce programme nécessite de modifier le bloc " **DRAW in GNUPLOT** " l'utilisation de Gnuplot.