

# Systèmes complexes

---

Compte rendu projet fournis



Etudiant : Ruffieux Mathis

# Sommaire

<b>I) Introduction</b>	p. 3
<b>II) Fonctionnement</b>	p.3
<b>III) Developpement</b>	p.7
<b>IV) Expériences</b>	p.8
<b>V) Conclusion</b>	p.23

## I) Introduction

Ce projet a pour but de comprendre le fonctionnement et les effets des algorithmes génétiques par optimisation en prenant l'exemple d'une colonie de fourmis.

Il a été réalisé en JavaScript en se basant sur les règles vues en cours.

Sur la simulation, les couleurs des phéromones posées par les fourmis et de l'odeur de la nourriture sont visibles mais pas les fourmis.

Après une simulation, on peut cliquer sur le bouton Data pour afficher les paramètres et scores de chaque fourmis et les exporter en format texte ou csv.

## II) Fonctionnement

### 1) Les règles

Le terrain est de dimension X par Y modifiable. Chaque fourmis se déplace sur une case et plusieurs fourmis peuvent être sur une même case en même temps.

Les fourmis ne peuvent pas traverser les murs, ni la nourriture.

### 2) Les phéromones

A l'initialisation, les fourmis démarrent sur le nid sans nourriture dans les mains.

Lorsqu'une fourmis n'a pas de nourriture, elle pose des **phéromones rouges** sur sa case.

Lorsqu'une fourmis se déplace sur une case adjacente adjacente à celle de la nourriture, elle devient porteuse de nourriture, elle pose alors des **phéromones vertes**.

Lorsqu'une fourmis se déplace sur une case adjacente à celle du nid, elle dépose la nourriture.

Toutes les fourmis se déplacent d'une case en même temps et posent leur phéromones en même temps sur la nouvelle case.

Après chaque déplacement, les phéromones (rouge et vert) de toutes les cases du terrain sont décrémentés de 5%.

De plus, après chaque déplacement d'une fourmis, la fourmi, le nombre de phéromones posés par la fourmi est décrementé de 1%. Par exemple, si elle pose de base 100 phéromones lors de son premier déplacement, elle en posera 99 à son prochain, puis 98 et ainsi de suite.

Ce processus permet qu'il y ait plus de **phéromones vertes** proche de la nourriture et plus de **phéromones rouges** proches du nid.

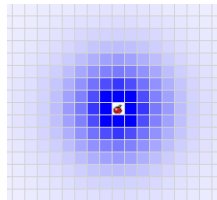
### 3) La nourriture

La nourriture est représentée par une pomme. 

La nourriture à une odeur (**phéromones bleues**). Chaque source de nourriture peut avoir une intensité d'odeur différente. L'odeur sur chaque case diminue d'un quart selon la distance par rapport à la source, suivant la formule :

$$\text{odeur} = \text{odeurNourriture} / (\text{distance}/4)$$

La distance est la racine de  $(a^2 + b^2)$  par rapport à la nourriture.



Voici à quoi ressemble l'odeur :

Cette odeur est fixe pendant toute la durée de la simulation.

#### 4) Durée de vie

Les fourmis ont une durée de vie initiale en nombre de pas. Une simulation s'arrête lorsque toutes les fourmis sont mortes. Une fourmi meurt lorsque son âge (nombre de pas) dépasse sa durée de vie.

Une fourmi morte ne se déplace plus et ne pose plus de phéromones, jusqu'à la génération suivante.

#### 5) Déplacements

Une fourmi a une direction (Nord Sud Est West). En fonction de sa direction, elle peut se déplacer tout droit, à droite ou à gauche. Lorsqu'elle est bloquée (3 murs autour d'elle), elle fait demi-tour.

Pour choisir son orientation, chaque fourmi dispose de deux paramètres qui lui sont propres.

Le paramètre **Proba**, est la probabilité d'ignorer l'odeur et les phéromones pour se déplacer aléatoirement. Il est compris entre 0 et 1.

Lorsqu'elle se déplace aléatoirement (ou qu'il n'y a aucune phéromone autour d'elle), elle a 80% de chances d'aller tout droit, 10% à droite et 10% à gauche. Cela lui permet de parcourir de plus grandes distances et d'éviter de tourner en rond avec 33% de chances pour chaque case.

Le paramètre **Alpha**, est l'importance accordée par la fourmi à l'odeur plutôt qu'aux phéromones. Il est compris entre 0 et 1 suivant la formule :

$$\text{probabilité de tourner à gauche} = (\text{odeur gauche} / \text{odeur des 3 cases}) * \alpha + (\text{phéromones verts gauche} / \text{phéromones verts des 3 cases}) * (1 - \alpha)$$

Ainsi, plus alpha est proche de 1 et plus la fourmi suivra l'**odeur des cases**, plus alpha est proche de 0 et plus la fourmi suivra **les phéromones des cases**.

Une fourmis ne va suivre l'odeur de la nourriture seulement lorsqu'elle ne détient pas de nourriture, sinon elle ne suit que les phéromones rouges et Alpha n'a pas d'incidence sur son choix de déplacement.

## 6) Evolution

L'évolution d'une génération de population de fourmis à une autre suit les règles d'un algorithme génétique avec sélection, croisement et mutation.

Pour chaque fourmis, les paramètres à faire évoluer sont :

- le paramètre Alpha
- le paramètre Proba
- la durée de vie (Lifetime)

### Sélection :

A l'issue d'une génération, la fonction d'aptitude retourne la fourmis ayant ramené le plus de nourriture au nid.

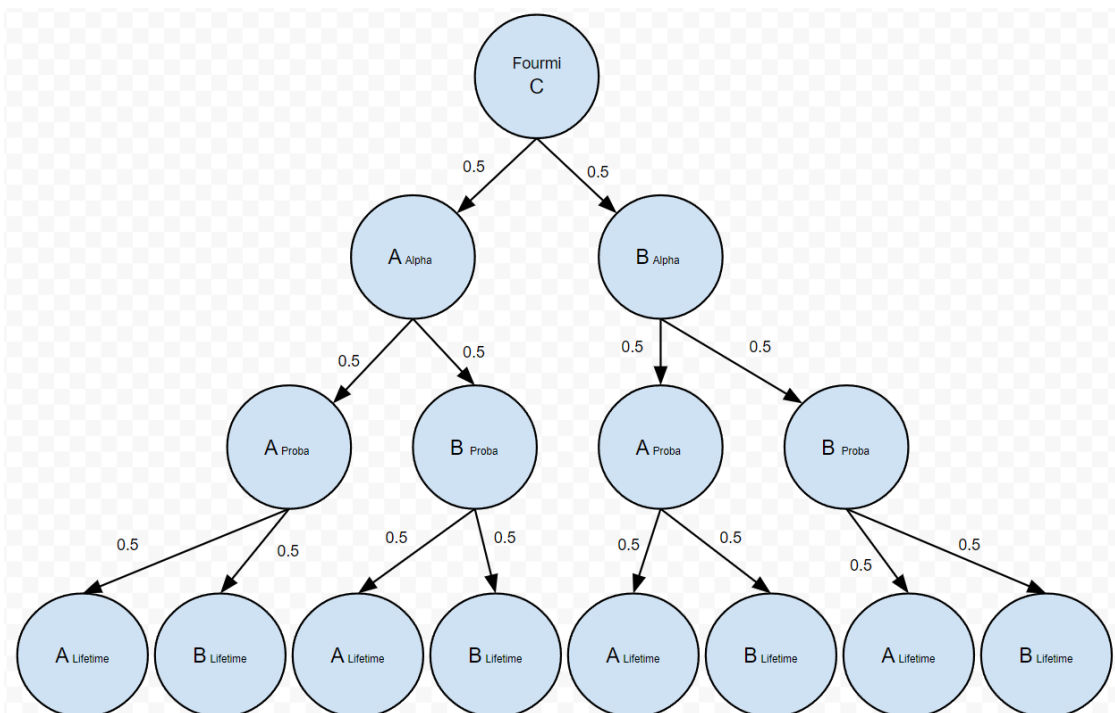
### Croisement :

La nouvelle population de fourmis aura le même nombre de fourmis que la précédente.

Pour cela, le croisement de chaque nouvelle fourmis sera effectué entre la fourmis précédente et la fourmis sélectionnée par la fonction d'aptitude.

Chaque nouvelle fourmis aura une chance sur deux d'avoir Alpha de la meilleure fourmis ou de la fourmis parente, une chance sur deux d'avoir Proba de la meilleure fourmis ou de la fourmis parente et une chance sur deux d'avoir la durée de vie de la meilleure fourmis ou de la fourmis parente.

Par exemple, si on suppose que la fourmis sélectionnée par la fourmis d'aptitude est A, et la fourmis d'indice i est B, voici la nouvelle fourmis C créée :

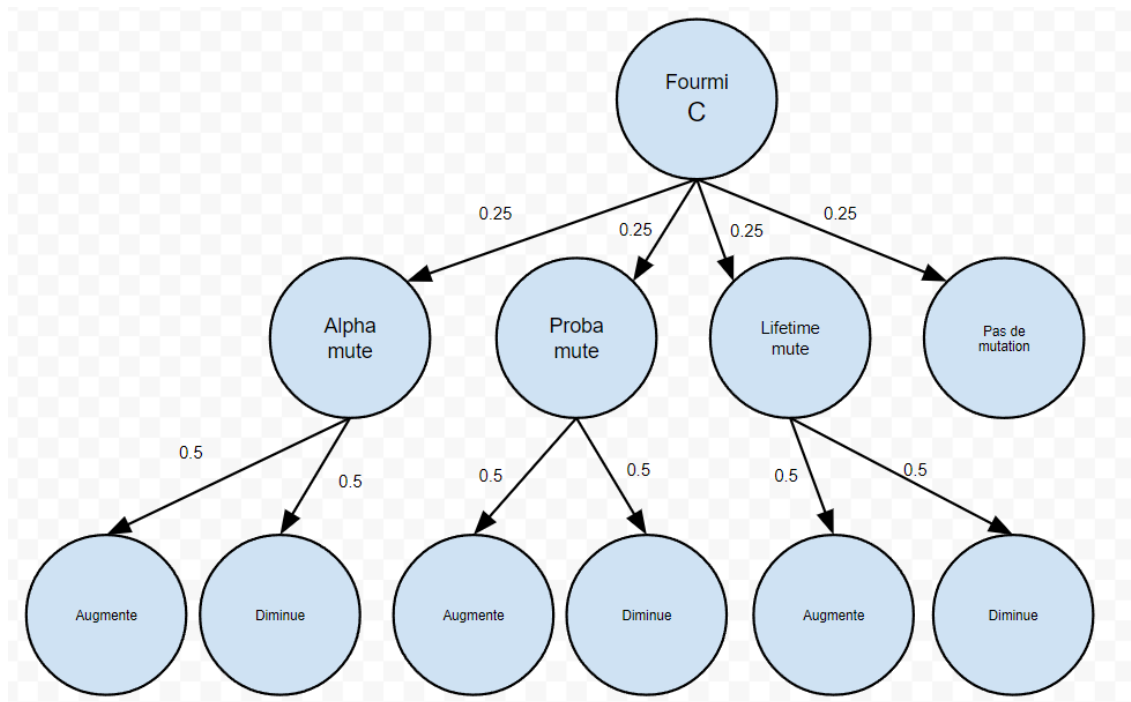


Il y a donc 8 possibilités de croisements pour chaque nouvelle fourmis.

### **Mutation :**

On fait muter au hasard l'un des paramètres ou aucun. Il y a une chance sur quatre que chacun des trois paramètres mute et une chance sur quatre qu'aucun ne mute.

Lorsqu'un paramètre mute, il a une chance sur deux d'augmenter de 10% de sa valeur ou de diminuer, permettant de faire des mutations progressives et pas brutales.



Il y a donc 7 possibilités de mutation.

Ainsi, la nouvelle génération est prête à démarrer. Toutes les fourmis se retrouvent au nid et les phéromones **vertes** et **rouges** de chaque case sont remises à 0.

## **III) Developpement**

Le projet est développé en HTML, CSS et JavaScript.  
Dans le code, il y a 3 modèles : Ant, Block et Terrain.

- **Ant** représente une fourmi, avec sa position dans le terrain (x,y) et ses paramètres qui lui sont propres. (Proba, Lifetime, Alpha)

Elle peut mourir, ramasser de la nourriture ou déposer de la nourriture.

- **Block** représente une case du terrain. Il peut être de type mur, nid, nourriture ou vide. Les fourmis ne se déplacent que sur les cases vides.

Un bloc dispose de phéromones rouges, vertes et bleues (pour l'odeur).

- **Terrain** contient l'ensemble des blocs dans un tableau en 2 dimensions pour une taille définie. Il peut modifier le type des blocs, poser un nid, poser de la nourriture. Lorsqu'on pose de la nourriture, il met à jour l'odeur de chaque bloc en fonction de la nourriture la plus proche.

Le fichier antsController.js contient toutes les fonctions nécessaires aux déplacements et à l'évolution des fourmis.

Le contrôleur génère de nouvelles générations de fourmis et pour chacune d'elles applique une fonction d'aptitude, de croisement et de mutation.

Pour déplacer les fourmis, on regarde les phéromones posées sur les blocs autour d'elle puis on calcule la probabilité d'avancer avec les formules expliquées précédemment.

De plus, lors de la création d'une nouvelle génération on enregistre l'historique des fourmis de la génération précédente dans un tableau oldAnts qui sera utile pour exploiter les données sur les expériences.

Le fichier dom.js gère toutes les modifications visuelles du DOM.

Il peut afficher un nouveau terrain, placer ou retirer des murs (*en cliquant sur une case lorsque l'expérience est en pause*), mettre en pause l'expérience, afficher ou cacher les données du bouton Data et surtout mettre à jour la couleur de chaque bloc en fonction de ses phéromones.

La couleur des blocs est sous la forme rgba() avec en rouge et vert les phéromones et en bleu l'odeur de la nourriture.

Ainsi, lorsque plusieurs types de phéromones se chevauchent il y a des dégradés de couleurs. Plus il y a de phéromones sur une case et plus l'intensité de la couleur sera forte suivant la formule :

$$\text{intensité} = (\text{phéromones} / [\text{phéromones de la case ayant le plus de phéromones du terrain}]) * 255$$

Les fourmis ne sont pas visibles pour ne pas ralentir l'expérience (400 fourmis) et comme le terrain est grand les cases sont trop petites pour y écrire en chiffre le nombre de fourmis sur chacune d'elles.

Dans la main.js, on initialise toutes les valeurs par défaut des variables : dimensions du terrain, position du nid et de la nourriture, nombre de fourmis, nombre de phéromones déposées par chaque fourmis etc.

La fonction runSimulation déplace les fourmis vivantes, met à jour les phéromones, incrémente le temps et passe à la génération suivante lorsque toutes les fourmis sont mortes.

## IV) Experiences

### Légende des graphiques :

- En Orange le score
- En vert la durée de vie
- En bleu le paramètre Alpha
- En rouge le paramètre proba

### 1) Première simulation

Il y a au total 400 fourmis sur cette simulation.

Les 2 sources de nourritures (pommes) sont à la même distance du nid, mais l'une d'elle a une odeur deux fois supérieure.

Nous allons faire tourner cette simulation sur 40 générations pour l'analyser.

### Capture d'écran de la première simulation (fin de la génération 21)



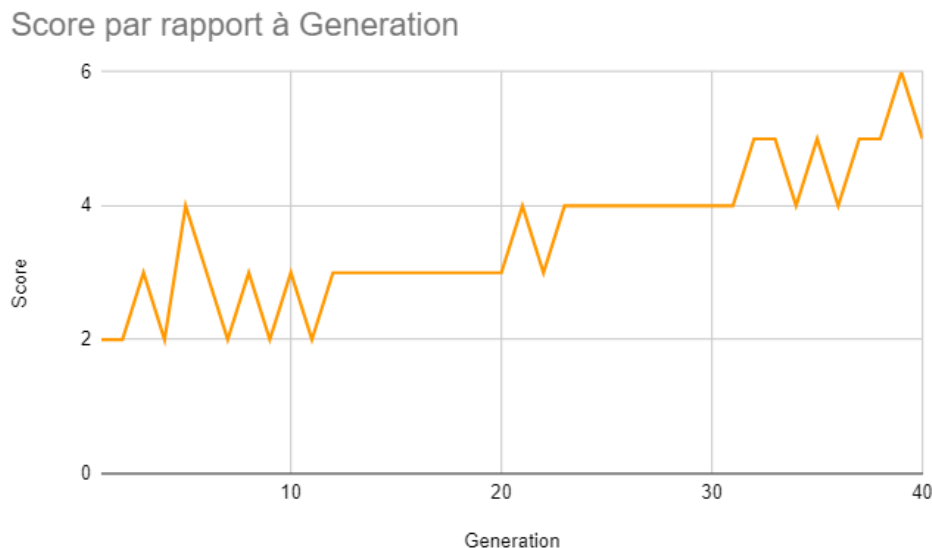
### Légende de la simulation (couleurs rgb par case) :

- En rouge les phéromones posés par les fourmis n'ayant pas de nourriture
- En vert les phéromones posés par les fourmis ayant de la nourriture
- En bleu l'odeur de la nourriture



Nous allons voir les paramètres évolutifs de la meilleure fourmis de chaque génération. Pour rappel, cette fourmis sélectionnée par la fonction d'aptitude et sert dans le croisement de la génération suivante. Elle est donc représentative d'une génération. Ses paramètres sont affichés dans la console à la fin de chaque génération.

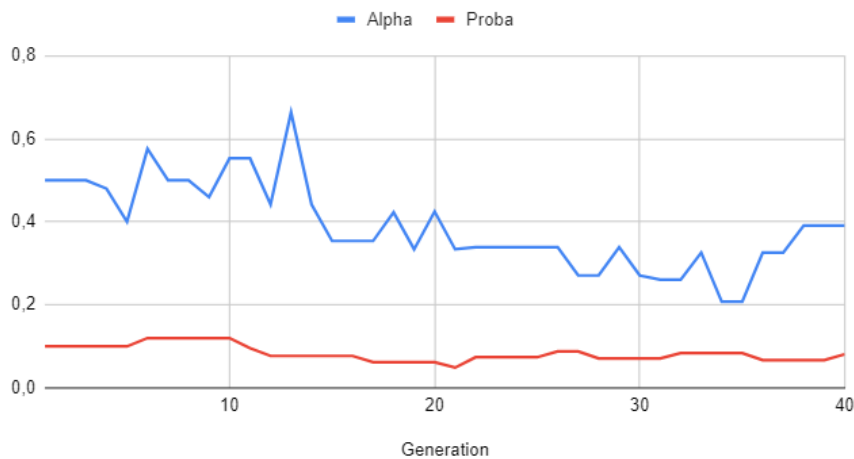
### Evolution du score de la meilleure fourmis par génération



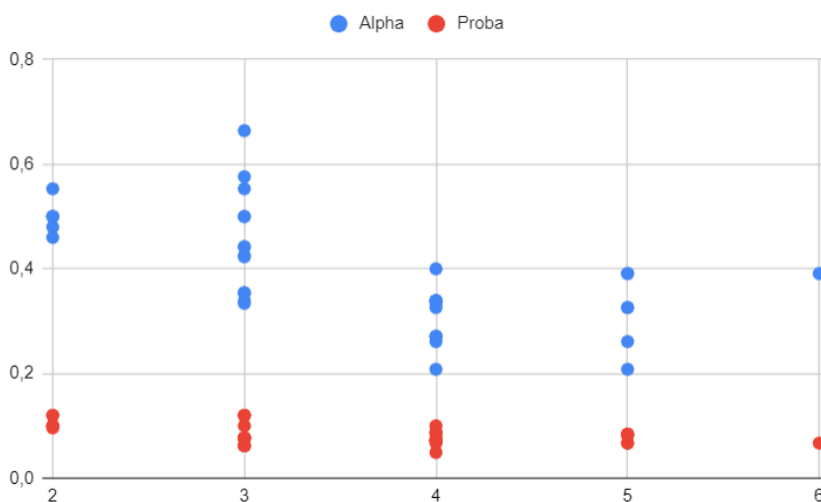
Le score est le nombre de nourriture rapporté par une fourmi.  
On observe que le score de la meilleure fourmis augmente petit à petit au fil des générations. A la génération 39 une fourmis a réussi à rapporter 6 nourritures au nid, alors qu'à la première seulement 2.  
On en déduit que l'algorithme génétique fonctionne car le score s'améliore.

### Evolution de Alpha et Proba de la meilleure fourmis par génération

## Alpha et Proba



## Paramètres Alpha et Proba de la meilleur fourmis selon le score

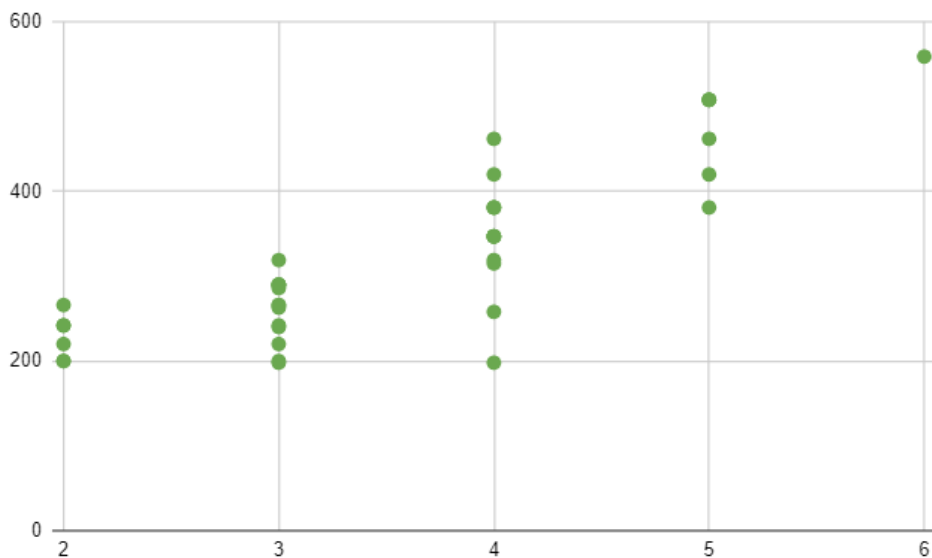


Ce graphique en nuage de points montre les paramètres Alpha et Proba en fonction du score de la fourmis. En abscisses, le score et en ordonnées, la probabilité des paramètres Alpha et Proba (compris entre 0 et 1).

On observe que les meilleures fourmis de chaque génération rapportant le plus de nourriture ont un Alpha entre 20 et 30% tandis que le Proba reste faible (vers 10%).

Dans ce cas, suivre les phéromones des autres fourmis est donc plus pertinent pour la meilleure fourmis que de suivre l'odeur de la nourriture.

### Durée de vie de la meilleure fourmis en fonction du score



Sur ce graphique on observe la durée de vie d'une fourmis en fonction du score. En abscisses, le score et en ordonnées la durée de vie (en nombre de pas maximum). On observe que plus le score est élevé, plus la durée de vie augmente. Notre hypothèse est donc que plus une fourmis vit longtemps et plus elle rapportera de nourriture.

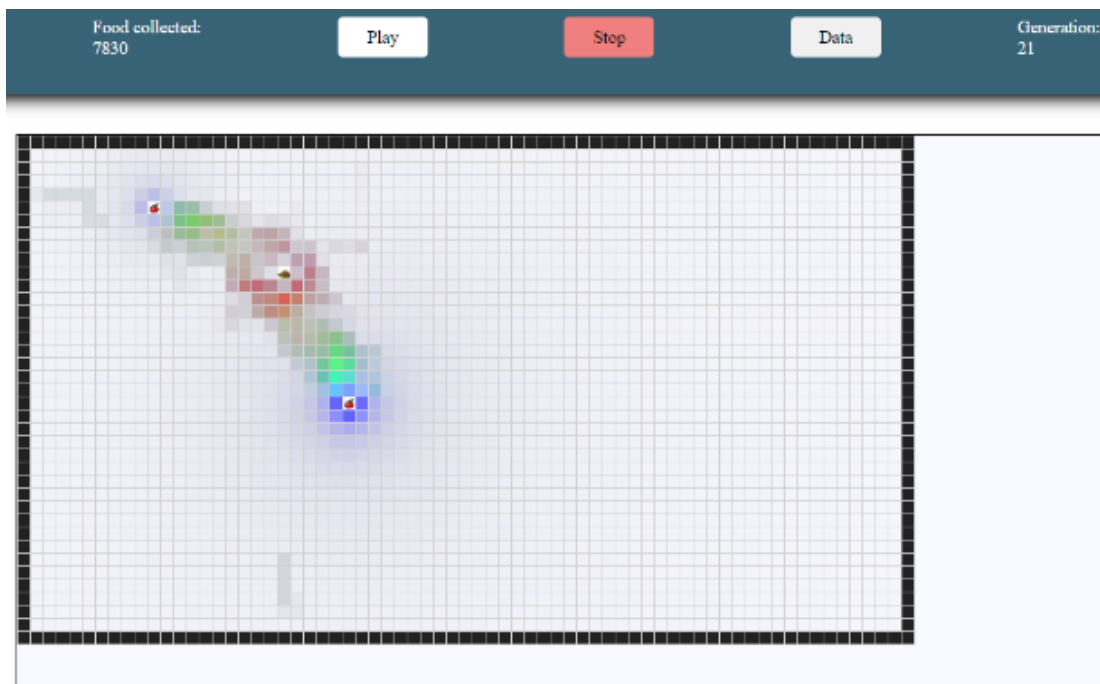
### **2) Deuxième simulation**

Cette deuxième simulation s'appuie sur la même configuration que la première, avec un terrain ouvert et deux sources de nourritures ayant des odeurs différentes en étant à la même distance du nid.

La différence, c'est que cette fois on va observer toutes les fourmis pour comparer leurs évolutions sur 20 générations.

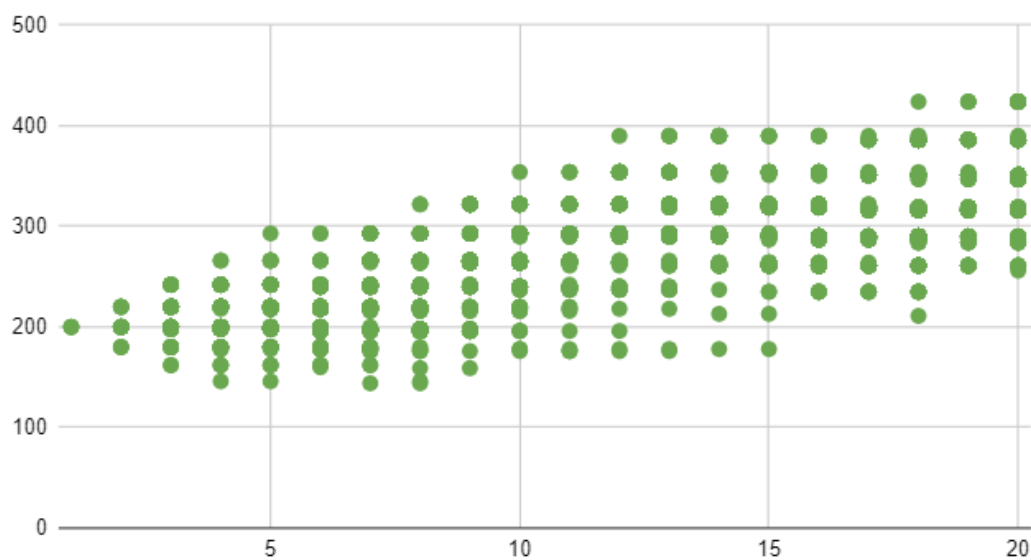
(Comme c'est une simulation différente de la première il se peut qu'on ait d'autres résultats)

### Capture d'écran de la deuxième simulation (fin de la génération 20)



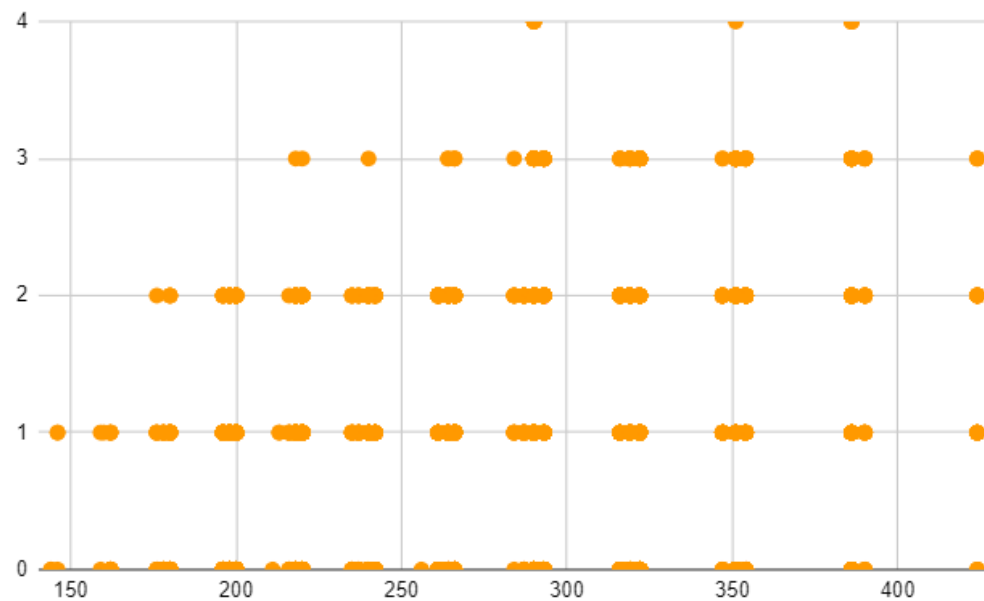
### Evolution de la durée de vie des fourmis selon la génération

Generation et Lifetime



Chaque point représente une fourmis. Les données en double ont été ignorées.  
On observe que plus les générations augmentent, plus la durée de vie de l'ensemble des fourmis est incremented. Les fourmis ayant une durée de vie plus courte ont disparu.

### Durée de vie en fonction du score



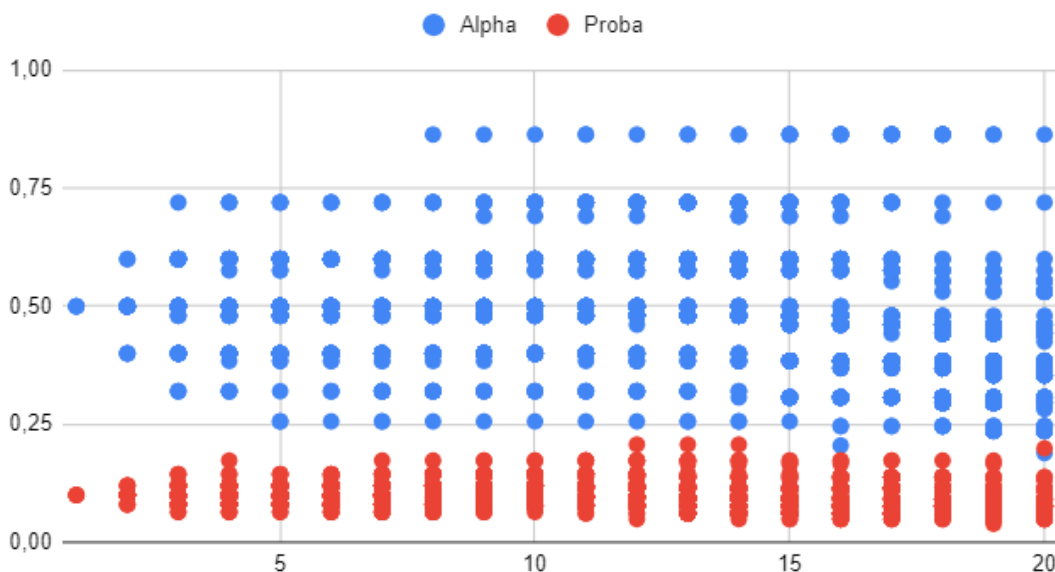
En abscisse la durée de vie et en ordonnée le score.

On observe que les fourmis ayant une durée de vie plus courte ne peuvent pas rapporter un maximum de nourriture.

Cependant même les fourmis ayant une durée de vie plus longue peuvent ne pas rapporter de nourriture au nid. La durée de vie des fourmis n'est donc pas le seul paramètre important pour maximiser le nombre de nourriture rapportées au nid car certaines avec une grande durée de vie sont performantes et d'autres avec la même durée de vie ne le sont pas.

### Evolution de Alpha et Proba au fil des générations

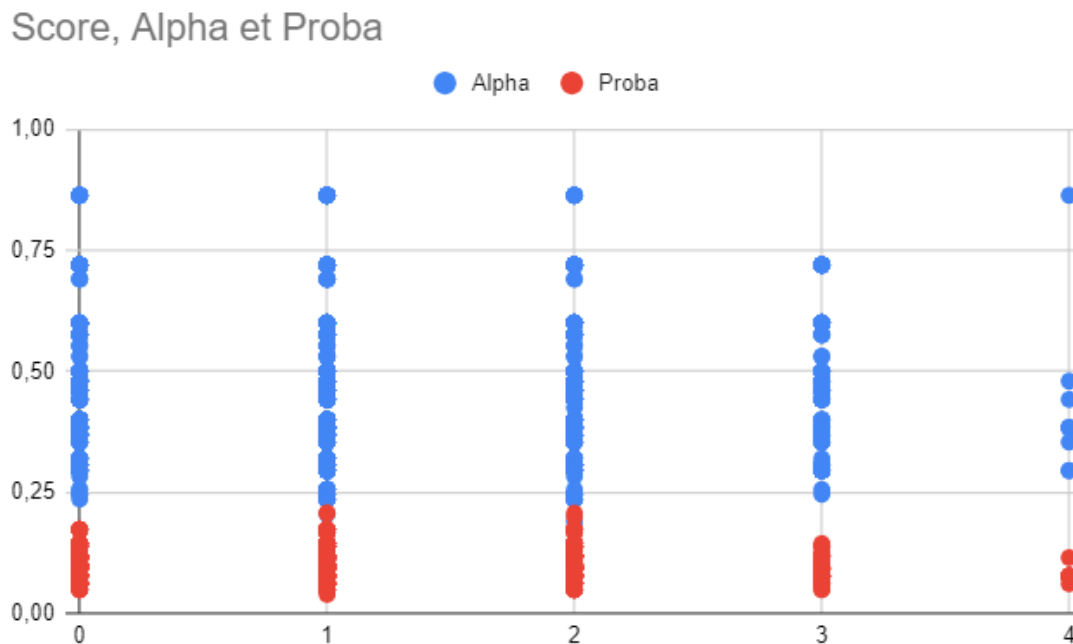
#### Generation, Alpha et Proba



On observe que le paramètre proba évolue très peu, il reste toujours faible. On peut supposer qu'une probabilité faible de suivre ni l'odeur ni les phéromones est plus optimale.

Le paramètre Alpha est très dispatcher, les fourmis suivant plutôt l'odeur ou plutôt les phéromones cohabitent et aucune d'entre elles ne disparaît totalement.

### Score des fourmis en fonction de Alpha et Proba



On observe toujours que le paramètre Proba reste faible. Les quelques fourmis ayant incrémentées ce paramètre n'ont rapporté qu'une ou 2 nourriture au nid. Les fourmis ayant une Proba plus faible sont plus performantes, ce qui explique pourquoi ce paramètre n'a pas évolué.

Le paramètre Alpha lui est plus optimal lorsqu'il est bas. En effet, la majorité des fourmis ayant rapporté 4 nourriture au nid suivent plutôt les phéromones que l'odeur de la nourriture avec Alpha entre 0.25 et 0.50, sauf pour une fourmis qui a peut être eu de la chance.

Il est difficile de conclure sur cette expérience, pour les fourmis, suivre l'odeur ou les phéromones semblent aussi importants l'un que l'autre.

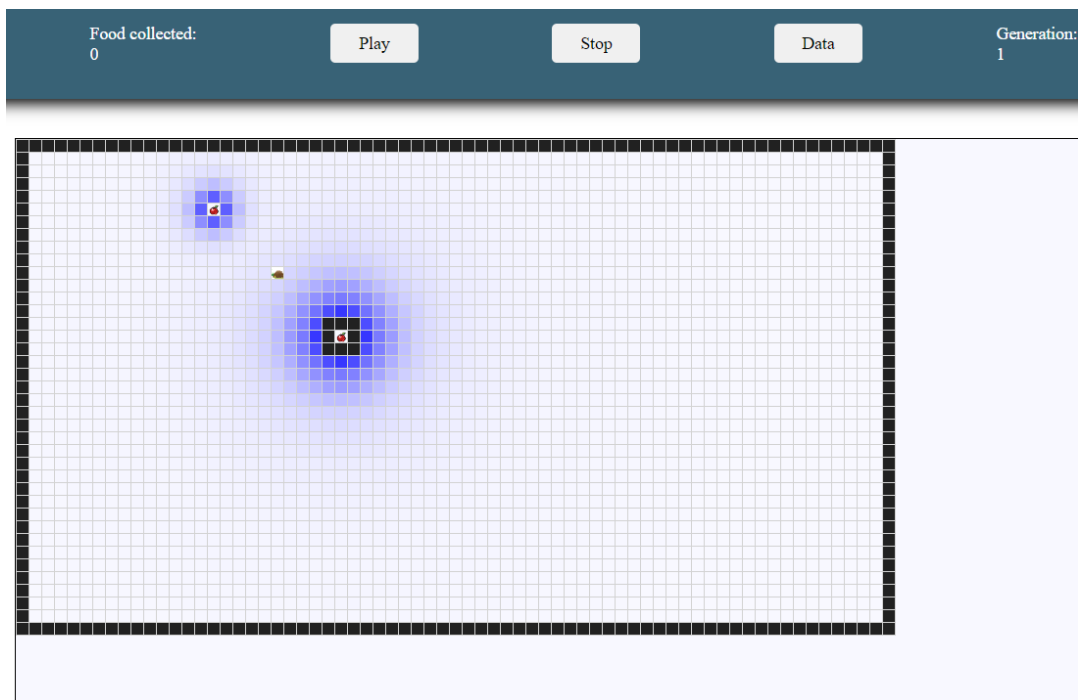
### 3) Troisième simulation

Pour cette nouvelle simulation, toujours avec 400 fourmis, on rapproche les nourritures du nid (10 cases chacune au lieu de 15) avec l'une d'elle ayant une odeur 2x supérieure à l'autre.

De plus, on la bloque avec des murs pour pas que les fourmis ne puissent y accéder.

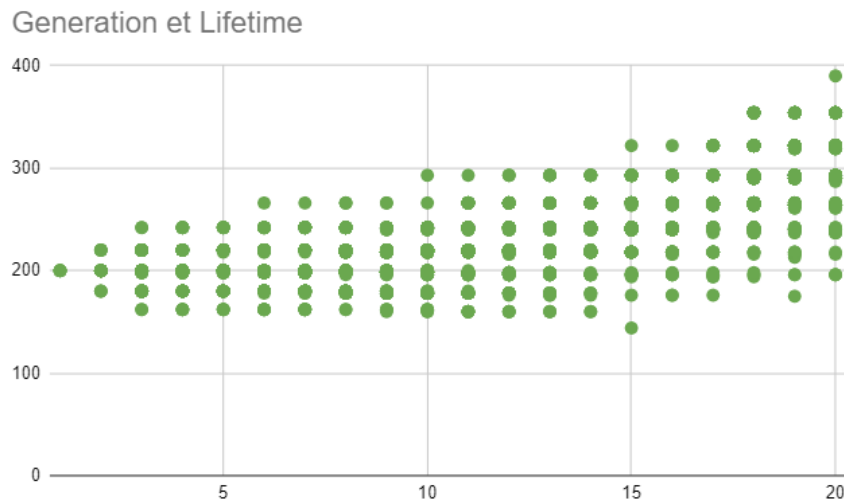
On pose l'hypothèse que le paramètre Alpha diminuera lors de l'évolution pour accorder moins d'importance à l'odeur et plus d'importance aux phéromones.

#### Screenshot de la troisième simulation



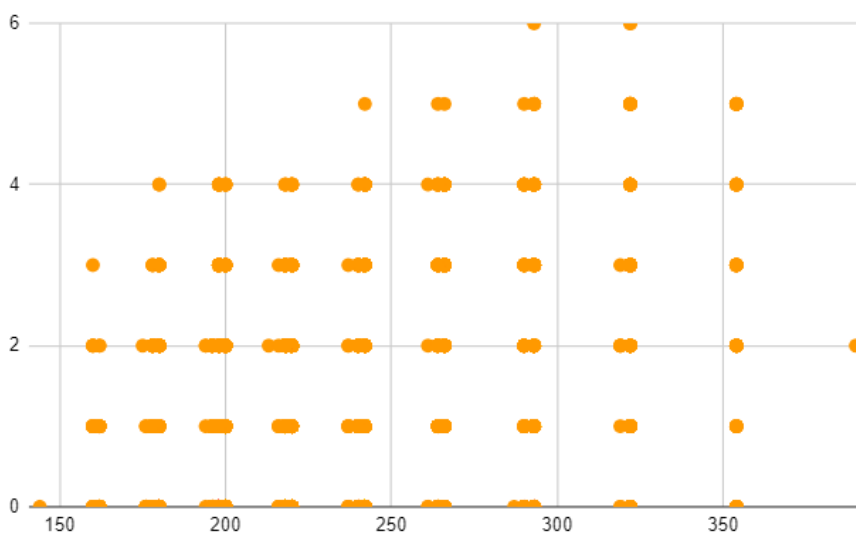
Lorsque je fais tourner la simulation, j'observe que les fourmis se jettent contre le mur de la source de nourriture ayant une plus forte odeur, mais font très vite demi tour pour aller vers l'autre en suivant les phéromones vert.

### Durée de vie selon les générations



Comme dans l'expérience précédente, on observe que la durée de vie des fourmis globale augmente au fil des générations. Cependant elle augmente moins vite, on suppose donc que d'autres paramètres ont évolué plus vite que la durée de vie.

### Score selon la durée de vie

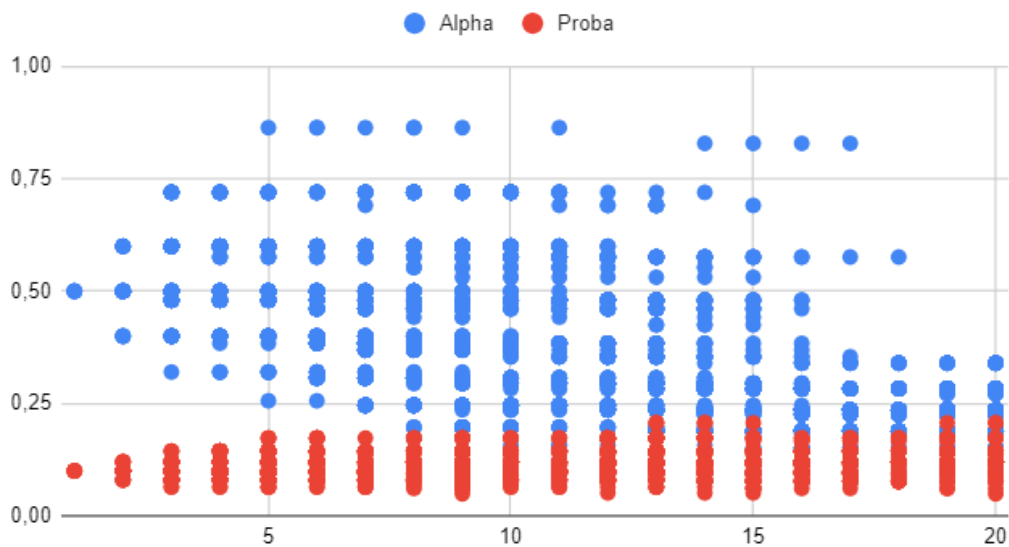


Sans surprise, plus une fourmi vit longtemps, plus elle peut rapporter de nourriture. Les fourmis vivant longtemps peuvent aussi ne rapporter aucune nourriture.

### Evolution de Alpha et Proba par génération



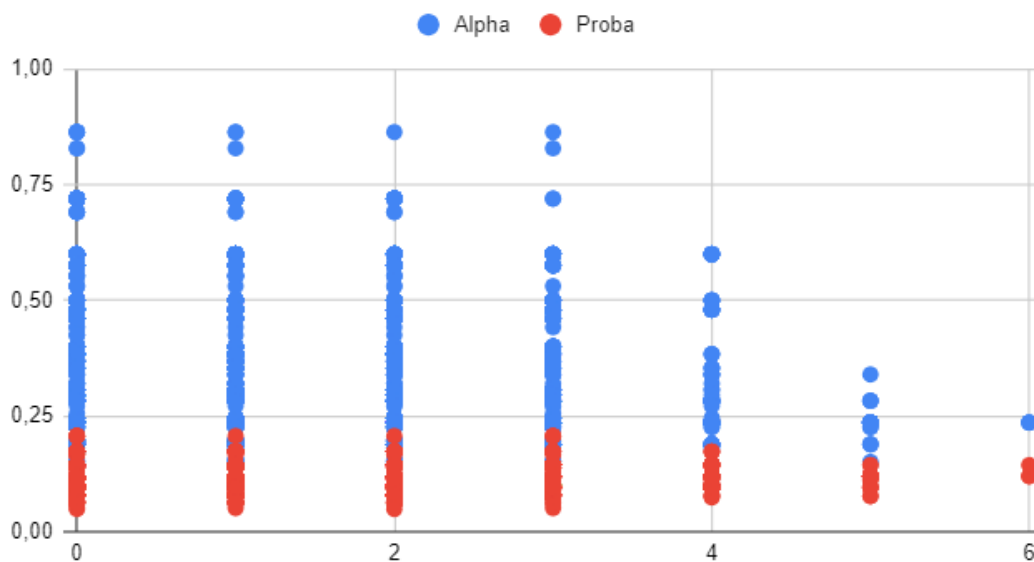
### Generation, Alpha et Proba



Ce graphique est intéressant. Contrairement à l'expérience précédente, le paramètre Alpha a diminuer et les fourmis accordant une importance plus forte pour l'odeur ont moins transmises leur gènes.

### Paramètres Alpha et Proba en fonction du score

#### Score, Alpha et Proba



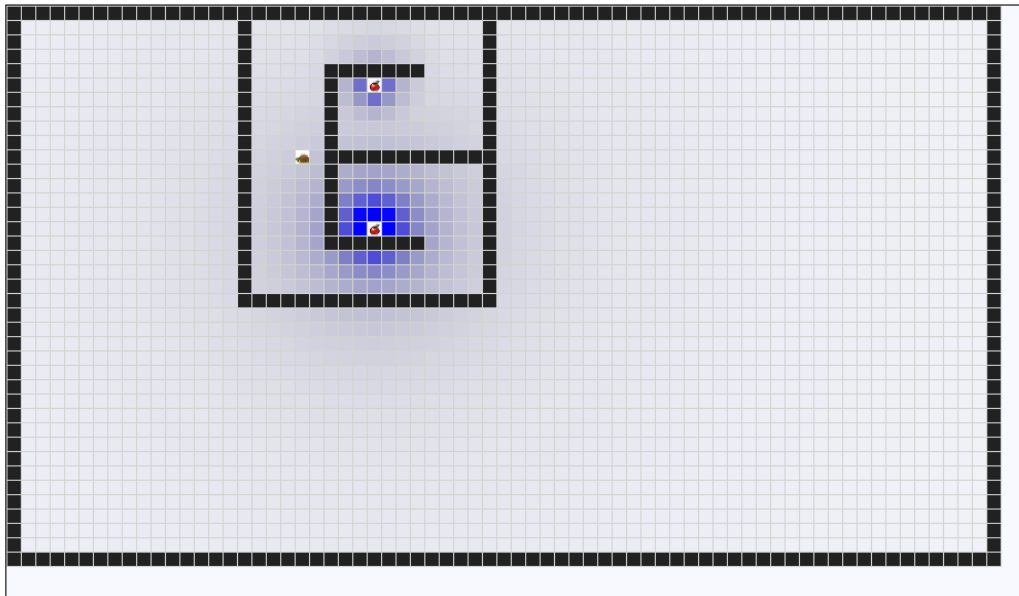
On observe ici que les fourmis ayant un paramètre Alpha plus faible ont pu rapporter plus de nourritures au nid.

Ces graphiques répondent à notre hypothèse, car les fourmis qui ont été “trompées” par l’odeur ont perdu du temps tandis que les fourmis privilégiant les phéromones sont allées plus rapidement vers la bonne source de nourriture et en ont ainsi rapporté plus.

#### 4) Quatrième simulation

Pour cette quatrième simulation, toujours avec 400 fourmis et deux sources de nourriture ayant une odeur différente, j’ai essayé de faire un petit labyrinthe pour voir leur évolution.

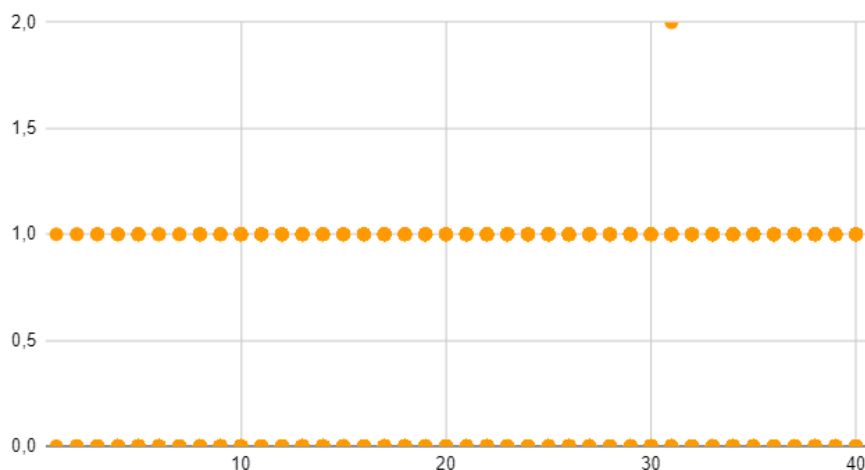
##### Capture d’écran de l’expérience



Mon hypothèse est que les fourmis seraient plus attirées par la source de nourriture du dessous et donc verraient leur attrait pour l’odeur augmenter (c’est à dire Alpha augmente) par effet de groupe.

##### Score de chaque fourmis en fonction des générations

Generation et Score

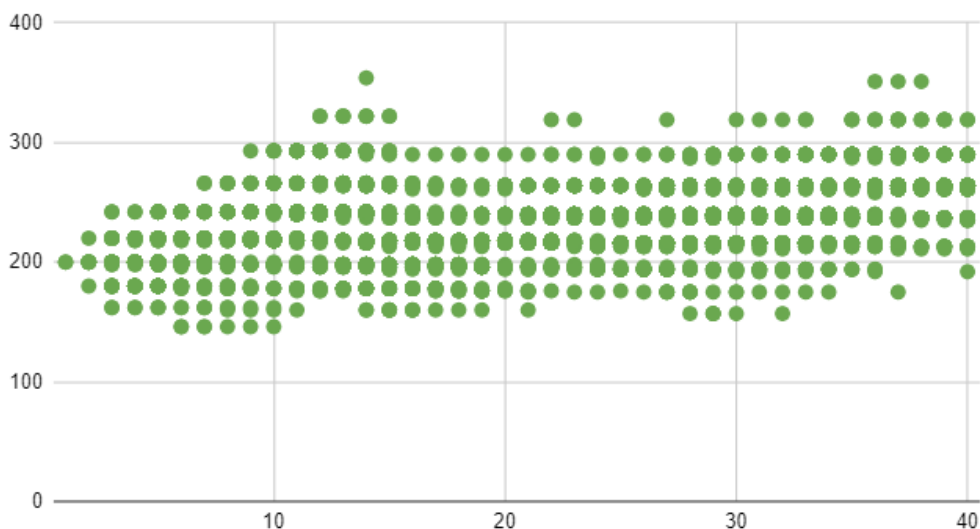


Malheureusement, après 40 générations les fourmis n'ont pas réussi à ramener plus d'une nourriture à la fois (sauf une fourmis lors de la génération 31). La fonction d'aptitude était donc obsolète pour cette expérience.

Peut être que le chemin d'accès à la nourriture était trop long par rapport à leur durée de vie pour en rapporter 2 lors d'une génération.

#### Durée de vie des fourmis en fonction de la génération

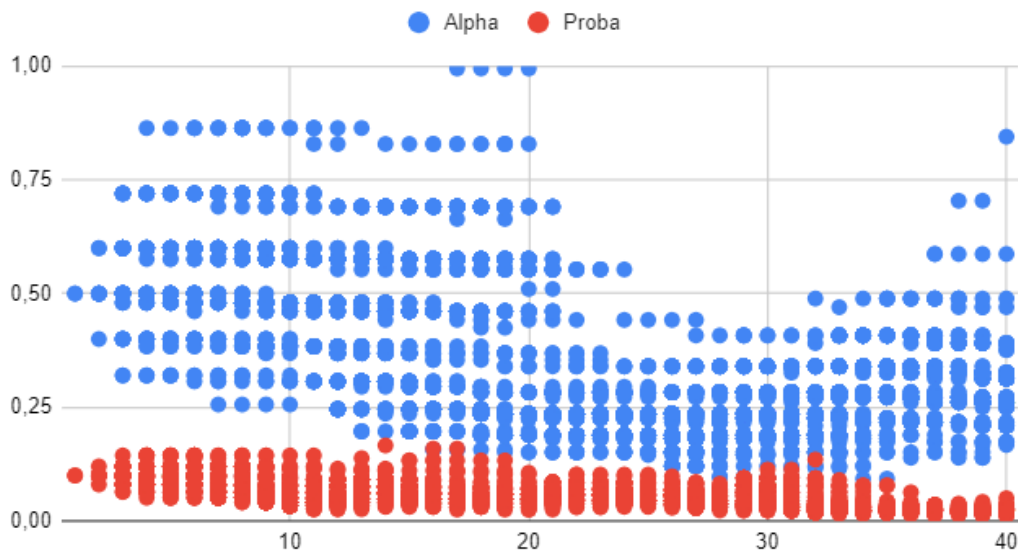
Generation et Lifetime



La durée des fourmis a tout de même augmentée mais pas suffisamment pour pouvoir rapporter plus d'une nourriture chacune.

#### Evolution de Alpha et Proba en fonction des générations

## Generation, Alpha et Proba

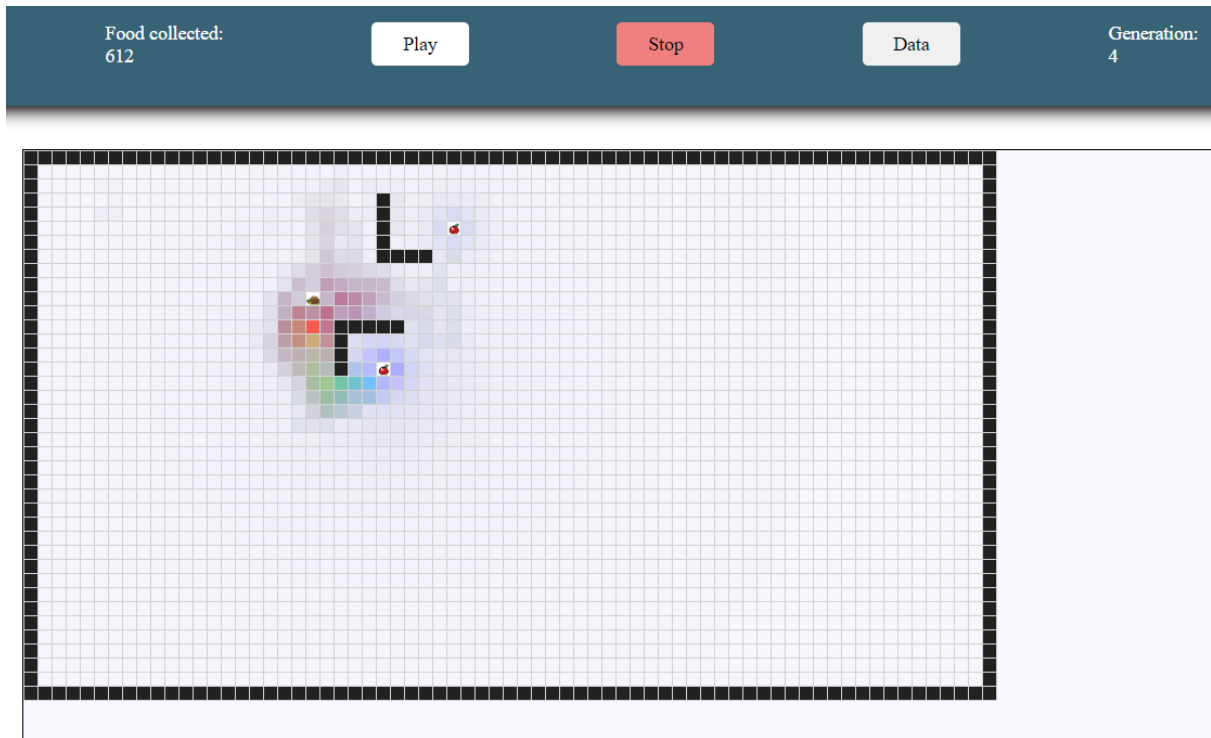


L'évolution des paramètres à l'air chaotique. Alpha a diminué, jusqu'à la 30eme génération puis augmenté à nouveau. Nous ne pouvons pas en tirer de conclusion évidente.

### 5) Cinquième simulation

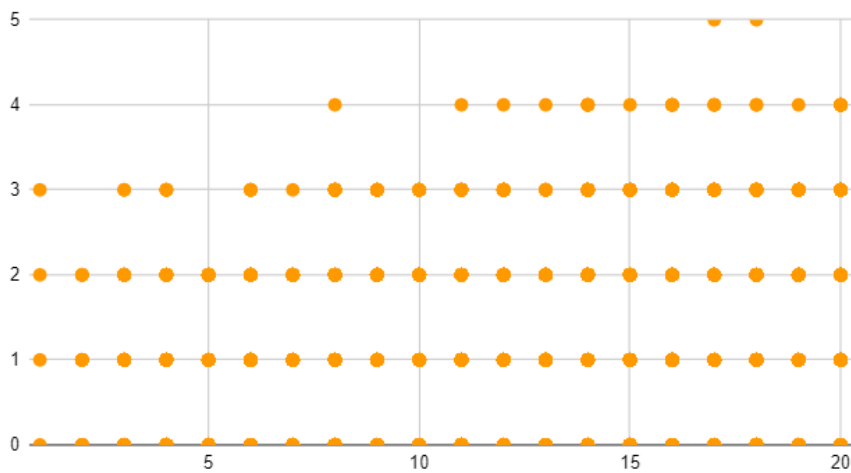
Pour cette dernière simulation, j'ai placé 2 sources de nourriture dont celle du dessus avec une odeur 2x plus forte et plus proche du nid que celle du dessous. J'ai ajouté des obstacles à contourner pour les deux nourritures.

Capture d'écran de la simulation



### Score par fourmis en fonction des générations

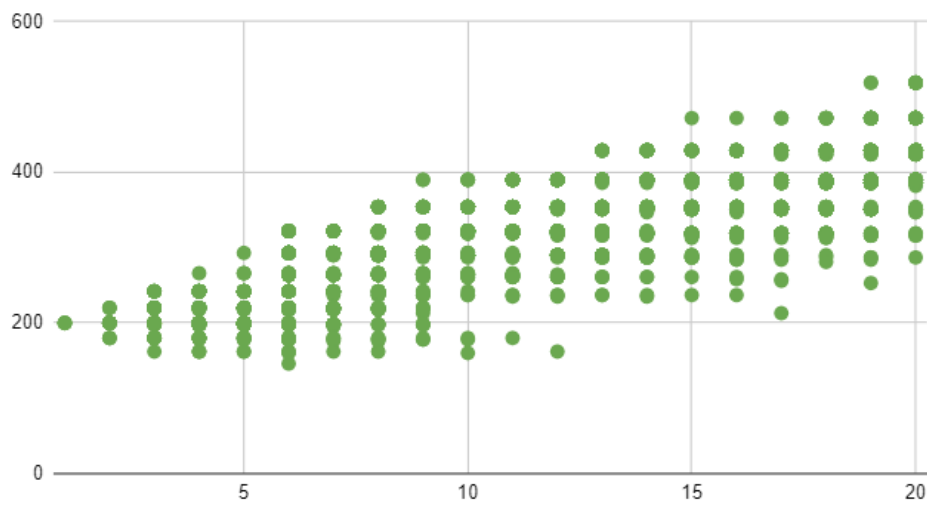
Generation et Score



Cette fois-ci l'expérience n'est pas un échec car le score des meilleurs fourmis de chaque génération a évolué.

### Durée de vie en fonction des générations

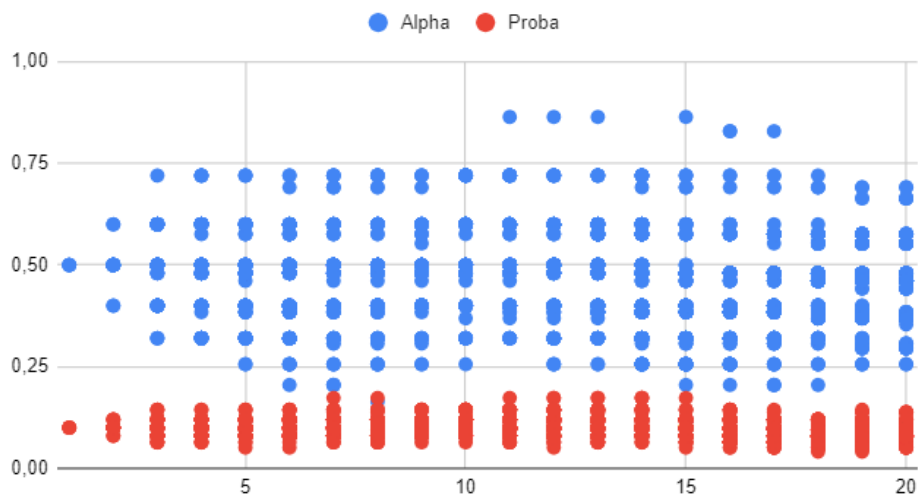
### Generation et Lifetime



Comme pour les autres expériences, la durée de vie globale a augmenté progressivement.

### Evolution de Alpha et Proba en fonction des générations

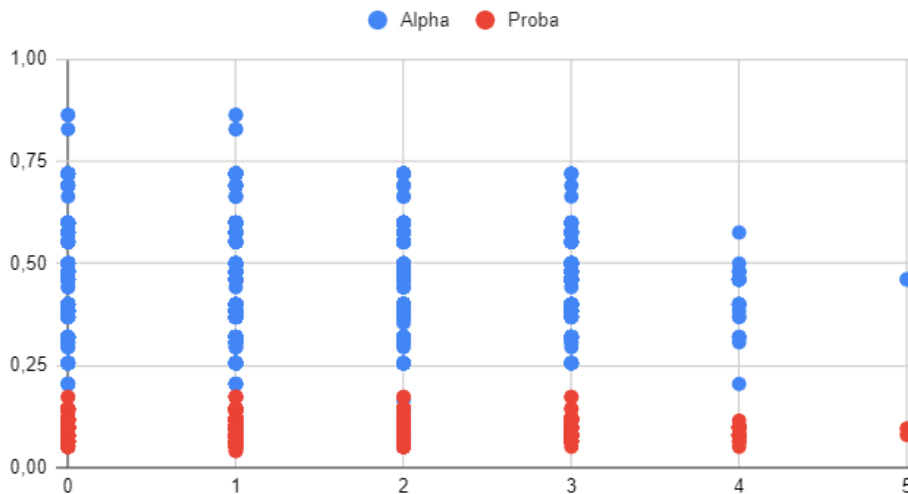
#### Generation, Alpha et Proba



Malgré que la source de nourriture la plus proche soit la plus odorante, cette expérience ne montre pas une évolution significative du paramètre Alpha.

### Alpha et Proba en fonction du Score

Score, Alpha et Proba



On observe une convergence de Alpha vers la moyenne lorsque le score est plus élevé.

Comme pour la première expérience, ces graphiques montrent que l'importance de l'odeur semble aussi pertinente pour les fourmis que l'importance des phéromones.

## V) Conclusion

En conclusion, nous pouvons dire que les algorithmes génétiques sont efficaces pour optimiser les paramètres des agents dans un environnement donné.

Pour cette simulation de fourmis, lorsque le terrain est sans obstacle l'odeur et les phéromones sont d'une importance équivalente pour optimiser la quantité de nourriture apportée au nid par fourmis.

Lorsqu'il y a des obstacles, suivre les phéromones se montre plus efficace. Le paramètre Alpha des fourmis diminue en conséquence.

D'après ces expériences, je n'ai pas trouvé de configuration de terrain permettant de rendre plus efficace l'odeur plutôt que les phéromones. On peut émettre l'hypothèse que parce que les fourmis se dirigent seulement à droite, gauche et tout droit sans faire demi-tour, l'odeur est moins pertinente à suivre, mais ce n'est pas démontré.

Globalement, la durée de vie augmente toujours quelque soit le terrain.

De plus, il est toujours plus efficace pour les fourmis de suivre l'odeur ou les phéromones plutôt que de se déplacer aléatoirement dans le terrain. (Le paramètre Proba reste faible.)