

Rapport SYMUA

Projet pollinisation



Guillaume CARRIERE, Youri COLERA,
Massil FERHANI, Paul GROLIER

Contents

1	Introduction	2
2	Description du modèle	2
2.1	Environnement	2
2.2	Agents	3
2.2.1	Ruche	3
2.2.2	Fleurs	3
2.2.3	Abeilles	4
2.3	Comportement du modèle	5
3	Analyse de sensibilité	5
3.1	bee-max-nectar (<i>Max Nectar Abeille</i>)	6
3.2	larvae-nectar-cost (<i>Coût Nectar Larve</i>)	7
3.3	soil-exhaustion-decay (<i>Récupération sol</i>)	9
4	Améliorations	10
4.1	Réalisme	10
4.2	Compétition	10
4.2.1	Fleurs	10
4.2.2	Abeilles	11
5	Conclusion	11

1 Introduction

Dans le cadre de ce projet de système multi-agents nous avons décidé de simuler la pollinisation de fleurs par des abeilles. Nous souhaitons en effet reproduire un système dans lequel des abeilles évolueront et se déplaceront à la recherches de fleurs à butiner.

Pour survivre les abeilles devront ramener assez de nectar à leur ruche afin de passer l'hiver. De l'autre côté les fleurs devront aussi trouver un moyen de survivre et passer l'hiver. Cela se fera par le biais de la reproduction. Une fleur butinée déposera du pollen sur une abeille. En allant visiter d'autres fleurs, l'abeille déposera le pollen dont elle est couverte sur les nouvelles fleurs leur permettant ainsi de se reproduire.

Nous tenterons à travers ce projet de répondre au problème de savoir s'il existe un équilibre, voire plusieurs, à partir duquel les abeilles et les fleurs peuvent survivre à chaque hiver systématiquement.

2 Description du modèle

Dans cette partie nous allons présenter une description complète de notre modèle, qui s'apparente à un modèle de type AGRE. Nous allons donc décrire l'environnement, ainsi que les objets et les agents qui évoluent dedans, d'abord de manière informelle, puis formellement. Enfin nous allons rapidement parler des comportements de notre modèle que nous avons constaté. L'idée générale du modèle est de tenter de simuler l'évolution d'une ruche d'abeilles et de son environnement naturel au fur et à mesure des années, dans l'objectif de répondre à notre problématique. Concrètement, nous nous sommes limités à la simulation informatique d'une ruche, des abeilles qui l'habitent, et des fleurs présentes dans son environnement naturel, et ce uniquement durant les printemps. Bien évidemment, la réalité est plus complexe puisque les abeilles interagissent avec d'autres animaux et d'autres végétaux que les fleurs, et ces points seront abordés dans la partie Améliorations.

Voici d'abord une description informelle de notre modèle :

La simulation consiste en plusieurs printemps s'exécutant à la suite. Le déroulement de chaque printemps va impacter les prochains printemps. Au début d'un printemps, la ruche crée une quantité d'abeilles pour partir récolter du nectar. Cette quantité sera proportionnelle au nectar récolté au printemps précédent. De leur côté, les fleurs sont remplies de nectar et attendent d'être butinées. Les abeilles vont alors passer tout le printemps à se déplacer et chercher les fleurs pour récupérer leur nectar, puis ramener le nectar à la ruche lorsque l'abeille est remplie, effectuant ainsi des aller-retours. Lorsqu'une abeille se dépose sur une fleur, le pollen de cette fleur se collera à l'abeille pour le reste du printemps, et inversement tout le pollen présent sur l'abeille se dépose sur la fleur. Pour essayer de coller à la réalité de la fécondation des pistils, ce dépôt n'est pas une consommation : la fleur gagne le pollen de l'abeille mais l'abeille conserve aussi le pollen. A la fin de la saison, les abeilles ramènent tout le nectar restant à la ruche et meurent (dans la réalité elles sont alors remplacées par les abeilles d'hiver, qui vont hiverner dans la ruche pendant le restant de l'année). Les fleurs quant à elles vont créer des graines proportionnellement à la quantité de pollen qu'elles auront reçues durant la saison, puis les disperser autour d'elles. Pour essayer de simuler le fait que la densité de fleurs est limitée dans un environnement naturel (notamment à cause de la compétitivité d'autres plantes ainsi que la fatigue des sols), notre environnement va empêcher les graines plantées dans des sols dit "fatigués" de pousser. Le sol sera fatigué si trop de fleurs ont poussés dessus au cours des récentes années, ce système limitera ainsi la densité des fleurs dans notre modèle.

Ce qui suit est une description formelle des composants de notre modèle. Notez bien que toutes les valeurs numériques mentionnées dans cette description correspondent au modèle lorsqu'il utilise ses paramètres par défaut.

2.1 Environnement

Notre environnement est une grille de cases de dimensions 65x65. Cet environnement est déterministe et dynamique. En effet, chaque case de la grille possède un état décrit par sa *Fatigue*, une variable allant de 0 à $+\infty$, et ces états vont évoluer de manière déterministe. Les lois qui vont décrire l'évolution

de ces états sont les suivantes :

- La *Fatigue* va descendre de *Récupération sol* (par défaut 0.5) pour chaque case à chaque cycle.
- La *Fatigue* va augmenter de 1 sur une case lorsqu'un agent "fleur" est crée sur cette case.

L'environnement possède également un évènement *Nouveau Printemps* qui se réalise tous les 90 cycles, et qui est détectable par les agents.

2.2 Agents

Dans notre modèle, tous les objets sont des agents, on a $O = A$. Pour chaque type d'agent nous allons décrire ses perceptions, ses variables internes, ses actions, ses tâches, ses stimuli et éventuellement ses relations et ses tropismes.

2.2.1 Ruche

L'agent "ruche" est unique et est créé lors de l'initialisation du système, sur la case de coordonnées (0,0).

- Perception : La ruche peut percevoir l'évènement *Nouveau Printemps*.
- Variables internes :

Nectar : Nectar présent dans la ruche, peut aller de 0 à $+\infty$. Initialisé à 0.

- Actions :

Créer une abeille : La ruche crée une abeille sur sa case, au prix d'une quantité de *Nectar* égale à *Coût Nectar Larve* (par défaut 10).

- Tâches :

Créer les abeilles : La ruche exécute l'action *Créer une abeille* jusqu'à ce que sa variable *Nectar* soit égale à 0.

- Stimuli :

Nouveau printemps : Déclenché lorsque la ruche perçoit l'évènement *Nouveau Printemps*. Lance la tâche *Créer les abeilles*.

- Relations : La ruche possède une relation d'appartenance avec les abeilles, i.e. les abeilles appartiennent à la ruche.

2.2.2 Fleurs

Les agents "fleur" sont créés au nombre de 100 aléatoirement sur la grille lors de l'initialisation du système.

- Perception : Comme la ruche, une fleur peut percevoir l'évènement *Nouveau Printemps*.
- Variables internes :

Nectar : Nectar présent dans la fleur, peut aller de 0 à *Max Nectar Fleur* (par défaut 20). Initialisé à *Max Nectar Fleur*.

Pollen : Pollen présent dans la fleur, peut aller de 0 à *Max Pollen Fleur* (par défaut 20). Initialisé à 0.

- Actions :

Créer une fleur : La fleur crée une autre instance de fleur sur une case choisie aléatoirement dans un rayon de *Rayon reproduction* (par défaut 3) cases autour d'elle (sa case comprise). La création réussit uniquement si la *Fatigue* de la case choisie est inférieure à *Max Fatigue* (par défaut 2).

- Tâches :

Se reproduire : La fleur exécute l'action *Créer une fleur* un certain nombre de fois, ce nombre correspondant à la formule suivante : $\lfloor \text{Pollen} / \text{Max Pollen} \rfloor$. Ensuite, la fleur meurt et est donc supprimée.

- Stimuli :

Nouveau printemps : Déclenché lorsque la fleur perçoit l'évènement *Nouveau Printemps*. Lance la tâche *Se reproduire*.

2.2.3 Abeilles

Les agents "abeilles" sont créés au nombre de 30 sur la case de coordonnées (0, 0) lors de l'initialisation du système. Chaque abeille se déplace d'une case par cycle dans la direction de son tropisme si elle en a un, sinon sa variable *Direction* après lui avoir appliqué une variation de $\pm 10^\circ$.

- Perceptions : L'abeille peut percevoir les fleurs dans un cône devant elle (dans le sens de sa *Direction*). Ce cône est de 8 cases de hauteur et 60° d'ouverture. L'abeille peut également percevoir l'évènement *Nouveau Printemps*.
- Variables internes :

Direction : Angle décrivant l'orientation de l'agent dans l'environnement, va de 0° à 359° . Initialisé aléatoirement.

Nectar : Nectar que possède l'abeille, peut aller de 0 à *Max Nectar Abeille* (par défaut 5). Initialisé à 0.

Pollen : Pollen présent sur l'abeille, peut aller de 0 à $+\infty$. Initialisé à 0.

- Actions :

Donner le nectar: L'abeille donne tout son *Nectar* à la ruche

Prendre du nectar: L'abeille prend du *Nectar* sur une fleur, à hauteur de $\max(\text{Nectar de la fleur}, \text{Max Nectar Abeille} - \text{Nectar})$.

Déposer du pollen: L'abeille augmente le *Pollen* d'une fleur à hauteur de son propre *Pollen* (son *Pollen* ne descend pas).

Gagner du pollen: L'abeille gagne une unité de *Pollen*.

- Tâches :

Butiner : L'abeille exécute les actions *Prendre du nectar*, *Déposer du pollen* puis *Gagner du pollen* avec une fleur.

- Stimuli :

Sur fleur : Déclenché lorsque l'abeille est sur la même case qu'une fleur. Lance la tâche *Butiner* avec cette fleur.

Sur ruche : Déclenché lorsque l'abeille est sur la même case que la ruche. Exécute l'action *Donner le nectar*.

Repérer fleur : Déclenché lorsque l'abeille perçoit une fleur et que $\text{Nectar} < \text{Max Nectar Abeille}$. Tant que le stimuli est actif, adopte le tropisme *Vers fleur*.

Remplie de nectar : Déclenché lorsque *Nectar* est égal à *Max Nectar Abeille*. Tant que le stimuli est actif, adopte le tropisme *Vers ruche*.

Nouveau printemps : Déclenché lorsque l'abeille perçoit l'évènement *Nouveau Printemps*. Exécute l'action *Donner le nectar*, puis l'abeille meurt et est donc supprimée.

- Relations : L'abeille possède une relation d'appartenance avec la ruche, i.e. l'abeille appartient à la ruche.

- Tropismes :

Vers fleur : L'abeille se dirige vers la fleur qu'elle a perçue avec le stimuli *Repérer fleur*.

Vers ruche : L'abeille se dirige vers la ruche.

2.3 Comportement du modèle

Avec les paramètres par défaut nous avons constaté que le système finit au bout de quelques printemps par atteindre l'un de ces deux états stables : toutes les abeilles et les fleurs sont mortes, ou bien un équilibre s'est organisé entre le nombre de fleurs et d'abeilles. En général, cela va dépendre de la disposition initiale des fleurs, qui permettront ou non de permettre à la ruche de passer les premières saisons. Dans le second cas, un comportement émergent apparaît dans lequel on observe la création d'un disque de fleurs autour de la ruche. Cela s'explique en effet par le fait que les fleurs les plus proches de la ruche seront plus susceptibles d'être visitées et donc de se reproduire, ainsi ces dernières se reproduisent et leur nombre croît jusqu'à former un territoire précis qui nourrit la ruche. Il est intéressant de noter que la densité de fleurs dans ce périmètre est stable. En effet, elle ne peut pas augmenter car la fatigue des sols est à son maximum et ne le permet pas, mais elle ne descend pas grâce aux abeilles. Parfois, on peut aussi constater un effet de "rebond" dans lequel la croissance des fleurs et des abeilles est tellement forte qu'on aura pendant un ou deux printemps une densité de fleurs supérieure à ce qui est permis par la fatigue des sols, avant de redescendre et se stabiliser.

La forme du disque est impactée par les paramètres, par exemple la valeur des paramètres *Max Nectar Abeille* et *Max Nectar Fleur* va influencer la vitesse à laquelle les abeilles seront rassasiées en nectar et rentreront à la ruche. Cela va donc aussi directement impacter la distance parcourue par les abeilles avant de rentrer et donc le diamètre du disque de fleurs autour de la ruche une fois le modèle stabilisé.

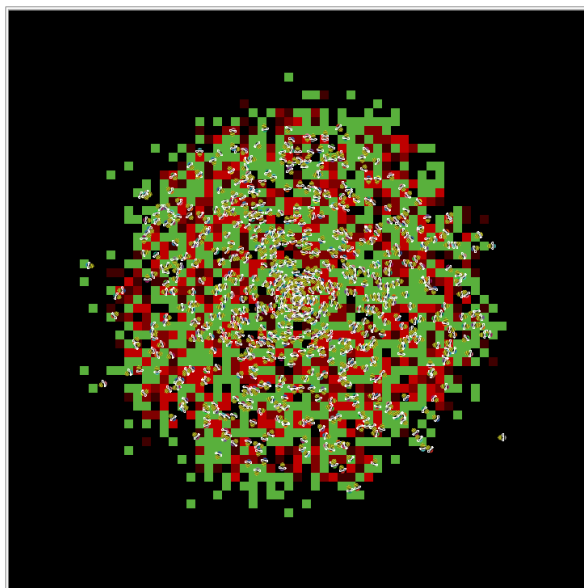


Figure 1: Disque de fleurs tel qu'observé dans NetLogo (fleurs en vert, fatigue en rouge)

3 Analyse de sensibilité

Nous avons actuellement 9 paramètres pouvant être modifiés dans notre modèle. Nous choisirons par mesure de simplicité de ne pas toutes les faire varier dans notre analyse de sensibilité. Nous avons décidé de retenir 2 paramètres concernant les abeilles et 1 concernant la fatigue des sols. Il s'agit des paramètres *bee-max-nectar* (correspondant à *Max Nectar Abeille*), *larvae-nectar-cost* (*Coût Nectar Larve*) et *soil-exhaustion-decay* (*Récupération sol*).

On relèvera les valeurs par défaut suivantes pour nos 3 paramètres :

- bee-max-nectar : 5
- larvae-nectar-cost : 10
- soil-exhaustion-decay : 0.5

Nous allons faire varier chacune de ces 3 variables tour à tour.

On notera que ces valeurs par défaut nous donnent les observations suivantes :

- Le rayon moyen du disque de fleurs autour de la ruche est de 29.
- La probabilité de convergence du système vers l'état stable où on peut observer notre disque de fleurs est d'environ 40 %.
- Le nombre moyen d'abeilles observées lorsque le système est stable est de 2450.
- Le nombre moyen de fleurs observées lorsque le système est stable est de 1350.

(On notera également que les probabilités de convergence du système ont été calculées empiriquement à mesure de 50 observations par valeur possible des paramètres.)

3.1 bee-max-nectar (*Max Nectar Abeille*)

On distingue 2 cas lors de la variation de cette variable.

Le premier cas arrive pour $bee-max-nectar = 0$. Dans un tel cas les abeilles ne peuvent simplement pas récupérer de nectar et donc il ne pourra y avoir aucune nouvelle abeille au prochain printemps ce qui engendrera la mort de tous les agents. Le deuxième cas concerne toutes les autres valeurs que peut prendre $bee-max-nectar$. Dans ce cas-ci nous pourrions observer une convergence stable du système. On peut alors relever les valeurs stabilisées du nombre d'abeilles et du nombre de fleurs pour chaque valeur de $bee-max-nectar$ entre 1 et 10 et observer la courbe d'évolution obtenue.

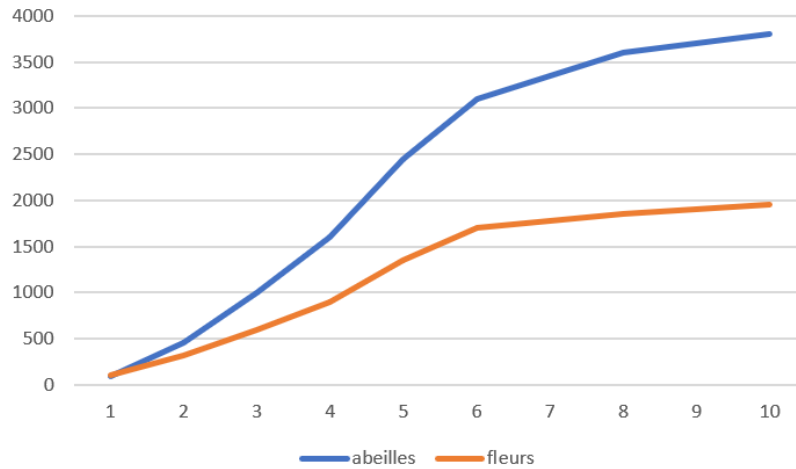


Figure 2: Évolution du nombre d'abeilles et de fleurs en fonction de $bee-max-nectar$

On observe en Fig. 2 que les courbes sont croissantes avec une allure de sigmoïde. En effet leur croissance est ralentie pour les valeurs maximales de $bee-max-nectar$ car à ce moment le rayon du disque de fleurs autour de la ruche devient trop grand par rapport à notre zone définie. Il n'y a donc presque plus de sols pouvant accueillir de nouvelles fleurs ce qui va majorer le nombre de fleurs possible et par conséquent le nombre d'abeilles aussi puisqu'il dépend du nectar donné par les fleurs.

L'évolution du rayon peut être vue en Fig. 3.

A partir de $bee-max-nectar = 6$ le rayon devient plus grand que ce que l'environnement ne le permet. On peut aussi alors considérer qu'on ne parlera plus de rayon car la zone de fleurs autour de la ruche n'est plus un disque. Nous avons alors essayé de deviner à partir des données précédentes ce que

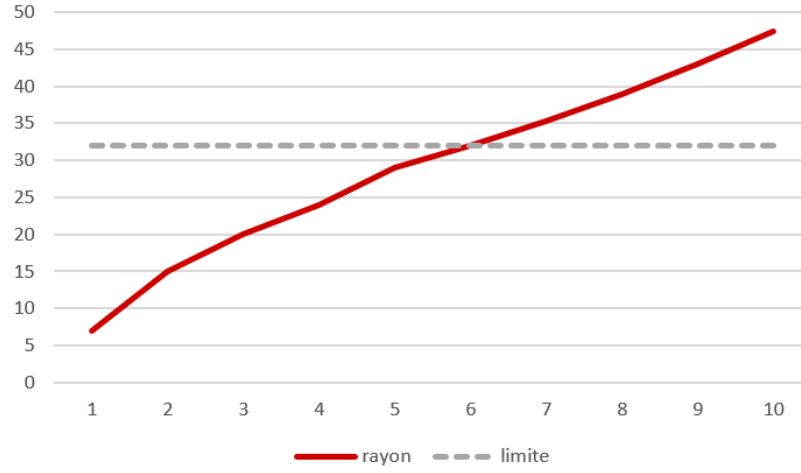


Figure 3: Évolution du rayon en fonction de *bee-max-nectar*

pourraient valoir les rayons des disques s'ils avaient pu être affichés entièrement.

On observe enfin l'évolution de la probabilité de convergence du système suivante en Fig. 4.

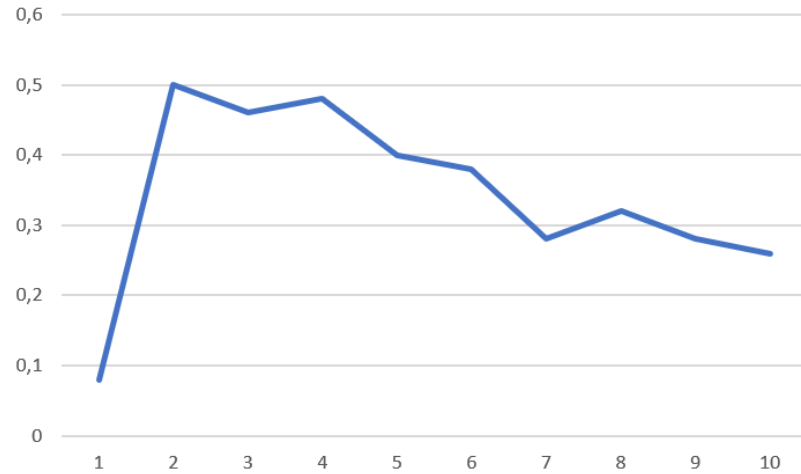


Figure 4: Évolution de la probabilité de convergence en fonction de *bee-max-nectar*

Le premier pic à 8 % s'explique par le fait que le système n'est pas assez stable pour garantir une convergence avec seulement *bee-max-nectar* = 1.

Les autres cas étant stables, on observe alors empiriquement une descente de la probabilité de convergence de 50 % jusqu'à 26 % pour la valeur maximale.

3.2 larvae-nectar-cost (*Coût Nectar Larve*)

En faisant varier ce paramètre on observe en Fig. 5 l'évolution suivante concernant le nombre d'abeilles et de fleurs dans le système lorsque celui-ci est stable.

On peut voir que plus le coût de création d'une larve est faible et plus le nombre d'abeilles sera grand. On peut d'ailleurs comparer l'allure de la courbe du nombre d'abeilles à une décroissante exponentielle tant le nombre d'abeilles sera grand pour une valeur basse de *larvae-nectar-cost*. Et c'est justement pour empêcher une situation avec trop d'abeilles que nous avons minoré ce paramètre à 5. Suivant cette même allure on verra donc que le nombre d'abeilles va décroître rapidement, et plus le paramètre sera élevé et plus la baisse de cette valeur sera faible.

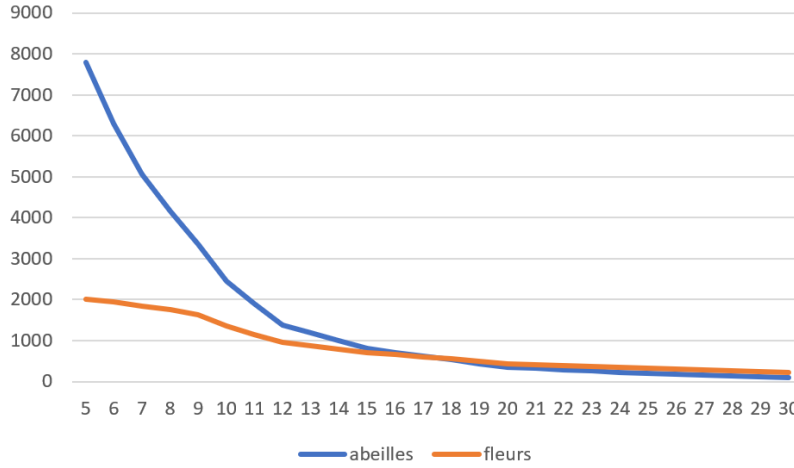


Figure 5: Évolution du nombre d'abeilles et de fleurs en fonction de *larvae-nectar-cost*

Concernant l'évolution du nombre de fleurs, celui-ci s'apparente plutôt à une décroissance sigmoïde. En effet on peut voir que pour des valeurs basses du paramètre le nombre de fleurs restera élevé et il sera bas pour des valeurs hautes du paramètre avec un changement noté surtout pour des valeurs moyennes de *larvae-nectar-cost*.

Concernant l'évolution du rayon du disque de fleurs autour de la ruche on observe en Fig. 6 un comportement assez similaire à celui du nombre d'abeilles et de fleurs.

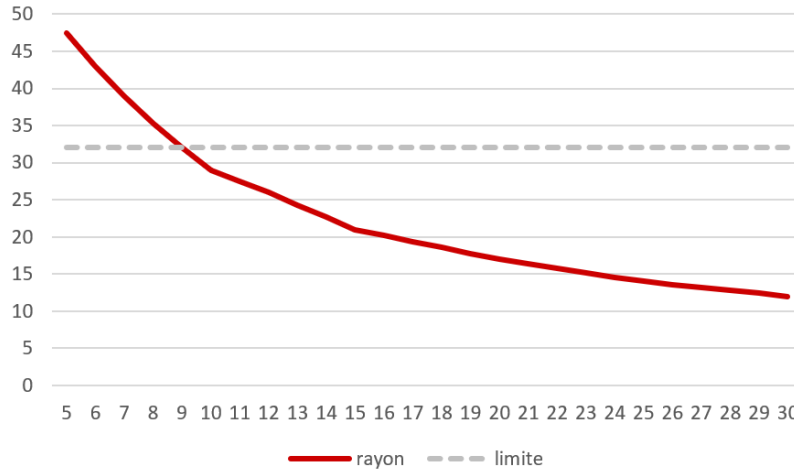


Figure 6: Évolution du rayon en fonction de *larvae-nectar-cost*

On peut voir là aussi une courbe à l'allure décroissante. On notera comme pour *bee-max-nectar* une limite à partir de laquelle le rayon devient trop grand et pour laquelle notre zone prédéfinie ne nous permet plus d'afficher entièrement le disque de fleurs. Cette limite semble se fixer à 9 pour *larvae-nectar-cost*. En dessous de cette limite la zone affichée ne pourra donc plus être considérée comme un disque. Comme pour *bee-max-nectar* on a essayé de déterminer ce que pourraient être ces rayons à partir des informations dont nous disposons.

Le dernier point concerne l'évolution de la probabilité de convergence du système. On observe alors empiriquement en Fig. 7 la courbe suivante.

Cette courbe décroissante nous fait comprendre que plus le coût d'une larve est élevé et plus les chances

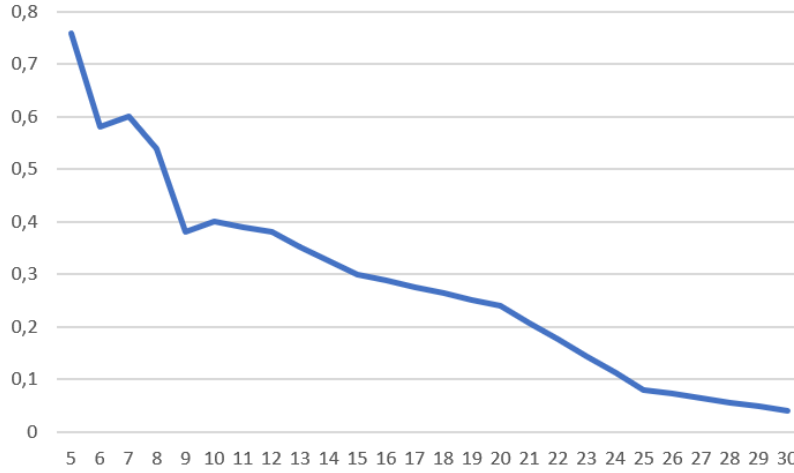


Figure 7: Évolution de la probabilité de convergence en fonction de *larvae-nectar-cost*

du système de converger sont faibles. En effet, on a un nombre d'abeilles et de fleurs fixe donné pour débiter nos simulations. Si ces valeurs sont trop faibles alors il n'y aura pas assez d'abeilles pour ramener suffisamment de nectar à la ruche au premiers printemps, ou il n'y aura pas assez de fleurs pour fournir assez de nectar. Ces valeurs faibles placent dès le début le système dans un état instable duquel il aura peu de chances de se sortir.

3.3 soil-exhaustion-decay (*Récupération sol*)

On note tout d'abord que la modification de cette variable n'impacte en rien la probabilité de converger du système ainsi que le rayon du disque de fleurs formé autour de la ruche une fois le système stabilisé. Les seules modifications concernent le nombre d'abeilles et le nombre de fleurs. On observe alors 3 cas distincts selon les valeurs.

Le premier cas arrive pour *soil-exhaustion-decay* = 0.0. Dans un tel cas le sol ne peut pas se régénérer et donc il est condamné à ne plus pouvoir faire pousser de fleurs au bout d'un certain temps. Le hasard des déplacements des abeilles permettra dans certains cas d'étendre le rayon de reproduction des fleurs mais il arrivera un moment où tous les agents finiront par mourir.

Le second cas arrive pour *soil-exhaustion-decay* = 0.1. Dans ce cas on peut observer une convergence du système mais nous n'aurons pas de point d'équilibre stable. Le système sera très instable imitant un effet yo-yo sur le nombre de fleurs, d'abeilles et de sols fatigués.

Le troisième cas concerne toutes les autres valeurs que peut prendre *soil-exhaustion-decay*. Dans ce cas-ci nous pourrions observer une convergence du système, qui plus est stable. On peut alors relever les valeurs stabilisées du nombre d'abeilles et du nombre de fleurs pour chaque valeur de *soil-exhaustion-decay* entre 0.2 et 1.0 et observer en Fig. 8 la courbe d'évolution obtenue.

On peut noter que cette courbe sera majorée par les valeurs obtenues pour *soil-exhaustion-decay* = 1.0 car le sol ne peut gagner que 1.0 de fatigue par cycle (lorsqu'une fleur germe) donc toute valeur supérieure ou égale à 1.0 suffira à régénérer complètement un sol.

On remarquera également qu'à partir de 0.9 le sol récupère si vite que cela ne fait plus de différence avec 1.0. Le nombre d'abeilles et de fleurs résultant est donc le même.

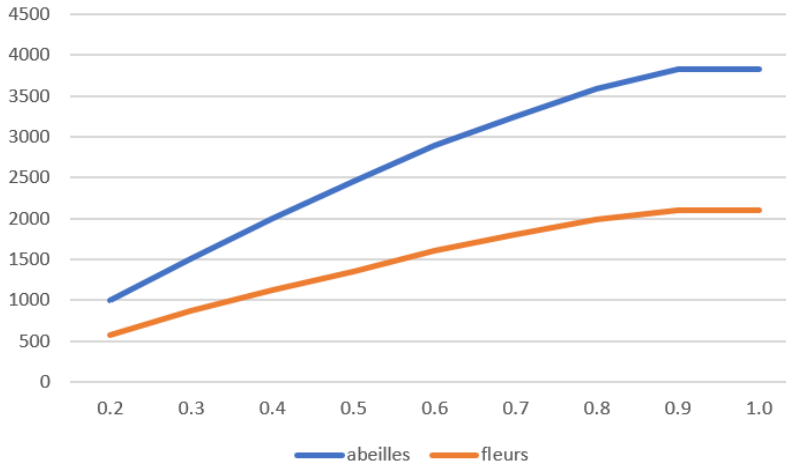


Figure 8: Évolution du nombre d'abeilles et de fleurs en fonction de *soil-exhaustion-decay*

4 Améliorations

4.1 Réalisme

La principale critique qui puisse être apportée au modèle serait le manque de réalisme. D'abord, l'initialisation pourrait être retravaillée, puisque le nombre de fleurs et abeilles initial est généralement très différents de leur nombre dans un état de modèle stable. Ensuite, il pourrait être intéressant d'augmenter la taille de l'environnement, pour éviter des situations où le territoire des abeilles ne peut plus se développer par manque de place. Enfin, beaucoup de paramètres rentrent en jeu dans l'évolution d'une ruche dans la nature et tous n'ont pas pu être simulés. Ainsi il faudrait essayer de se documenter afin de paramétrer notre modèle de manière à ce qu'il soit le plus réaliste possible. Cela demande d'obtenir des informations telles que les nombres (en moyenne) d'abeille dans une ruche, de fleurs que les abeilles d'une ruche vont polliniser en un printemps, de la quantité de fleurs pollinisées par une abeille en un jour etc. En obtenant ces informations il serait alors possible de calibrer les paramètres de notre modèle afin de nous rapprocher de la réalité. Enfin il existe probablement d'autres comportements et phénomènes complexes que nous n'avons pas implémentés, par exemple les habitudes de mouvement des abeilles (ces dernières possèdent tout un système de vol permettant de communiquer entre elles pour se déplacer efficacement).

4.2 Compétition

Un autre point qui peut être relevé, est le fait qu'un environnement naturel se compose rarement d'un seul type de fleur et d'une seule ruche. En pratique on a plusieurs ruches d'abeilles couvrant un large territoire dans lequel on peut trouver une variété très diverse de différentes fleurs, avec chacune leur stratégies. Notre modèle pourrait donc être amélioré en introduisant de la compétition entre plusieurs espèces de fleurs et d'abeilles. Cela permettrait également, en plus de rendre notre modèle plus réaliste, de répondre à une nouvelle problématique. Cette nouvelle problématique serait de déterminer la "stratégie gagnante", permettant à une espèce de fleurs ou d'abeilles parmi plusieurs de dominer son environnement.

4.2.1 Fleurs

Pour ajouter cette particularité aux fleurs, il suffirait de rajouter à notre modèle plusieurs stratégies pour les fleurs. Chaque stratégie ferait varier les différents paramètres de la fleur, tels que le nectar qu'elle contient, le coût de reproduction ou encore la quantité de fatigue qu'elle produit en germant. De plus il serait possible de rajouter des comportements particuliers, par exemple faire en sorte qu'une espèce de fleur ne fleurisse que vers le début ou vers la fin du printemps, ou bien rajouter des paramètres

supplémentaires comme la capacité d'attraction de la fleur (correspondant à ses couleurs etc.).

4.2.2 Abeilles

Pour ajouter cette particularité aux abeilles, il est nécessaire d'implémenter la possibilité de faire coexister plusieurs ruches. En effet, si on veut être réaliste toutes les abeilles d'une ruche doivent être identiques, et simuler plusieurs espèces d'abeilles demandera donc d'avoir plusieurs ruches. Ainsi il sera possible de faire varier les paramètres des abeilles tels que le coût de production d'une abeille, la vitesse ou encore la capacité de stockage de nectar d'une abeille. Il serait également possible de faire varier les comportements mentionnés plus tôt tels que les habitudes de déplacements ainsi que la communication entre les abeilles. Éventuellement, il serait intéressant d'implémenter des interactions directes entre les abeilles, par exemple des conflits. Enfin, le fait de simuler plusieurs ruches rajoute la possibilité de faire varier leur position, ce qui permettrait par exemple d'étudier quelle stratégie est la plus efficace dans un contexte où il y a une forte/faible densité de ruche, voire construire des scénarios particuliers dans lesquels certaines zones de l'environnement comportent une seule espèce d'abeille etc.

5 Conclusion

Ce projet avait pour objectif de créer un état stable censé représenter l'équilibre naturel se formant entre les ruches d'abeilles et leur environnement, et de l'étudier en fonction de paramètres permettant de symboliser un contexte réel. Au vu de nos résultats nous considérons que cet objectif est majoritairement rempli et nous sommes satisfaits. Cependant, beaucoup d'améliorations pour obtenir un modèle plus proche de la réalité pourraient être rajoutées, notamment afin d'étudier la compétitivité naturelle des espèces. Selon nous, notre travail pourrait donc former une bonne *proof of concept* pour un projet de plus grande envergure, avec des perspectives de recherche en biologie et en zoologie.