

Modèle d'Interruption pour la Répartition des Tâches : Application à une simulation de colonie d'abeilles.

Thomas Alves¹, Jérémy Rivière¹, Cédric Alaux², Yves Le Conte²
Frank Singhoff^{1,4}, Thierry Duval³, Vincent Rodin¹

¹ Univ Brest, Lab-STICC, CNRS, UMR 6285, Brest, France
{prenom.nom}@univ-brest.fr

² INRAE, UR 406 Abeilles et Environnement, Avignon, France
cedric.alaux@inrae.fr, yves.le-conte@inrae.fr

³ IMT Atlantique, Lab-STICC, CNRS, UMR 6285, Brest, France
thierry.duval@imt-atlantique.fr

⁴ Groupement de Défense Sanitaire Apicole du Finistère (GDSA29), France

Résumé

Les principaux modèles d'allocation de tâches reposent sur l'hypothèse que chaque tâche est associée à un stimulus, et que l'exécution de cette tâche réduit ce stimulus. Dans cet article, nous proposons un modèle basé sur les modèles classiques "à seuil" (Fixed Threshold Models) et sur la motivation intrinsèque de l'agent, qui permet de traiter des tâches déconnectées de stimulus. Chaque agent utilise cette motivation, influencée par ses performances dans la réalisation de la tâche, pour évaluer la priorité de cette tâche et décider de l'interrompre ou de la poursuivre. Nous avons implémenté ce modèle afin de simuler la répartition des tâches dans une colonie d'abeilles simplifiée. Les premiers résultats montrent que la répartition des tâches est efficace et robuste, et que l'utilisation de la motivation intrinsèque comme mécanisme d'interruption semble l'améliorer.

Mots-clés

Simulation à base d'agents, Répartition des tâches, Auto-Organisation, Insectes sociaux.

Abstract

Task allocation model often rely on the assumption that each task is associated to a stimulus, and executing this task will lower the intensity of that stimulus. We propose here a model based on response thresholds model associated with intrinsic motivation and internal states of the agents able to handle tasks dissociated of stimulus. Each agent will estimate how important is its current task based on its results and decide whether to interrupt or pursue its current task. Preliminary work shows that task allocation is effective and that our interruption model seems to improve the overall results.

1 Introduction

La capacité des insectes sociaux à répartir leur main-d'œuvre sans aucun contrôle central est étudiée depuis plus d'une vingtaine d'années. Cette capacité d'adaptation dy-

namique à leur environnement permet aux insectes sociaux d'être résistants aux changements et de continuer à prospérer. En informatique, et plus particulièrement dans les systèmes multi-agents, les modèles de répartition des tâches adaptés des insectes sociaux sont nombreux et efficaces, utilisés par exemple en simulation [1], [2] et pour la résolution générale de problèmes [3], [4]. Les modèles de répartition des tâches reposent principalement sur deux hypothèses : 1) chaque tâche est associée à un stimulus et 2) l'exécution de cette tâche abaisse le niveau de son stimulus, alors que sans exécution ce stimulus augmente [5]. Un agent percevant ainsi un stimulus externe (dans l'environnement) ou interne (relatif à son état physique) déclenche l'exécution de la tâche associée, ce qui diminue l'intensité du stimulus et la nécessité de refaire cette même tâche. Ainsi, le stimulus perçu peut être considéré comme une représentation du travail nécessaire à la tâche.

Cependant, dans certains cas, l'exécution d'une tâche n'affecte pas l'intensité du stimulus perçu associé, ou il y a même une absence totale de lien entre une tâche et un stimulus. Dans ce cas, nous devons trouver d'autres moyens de piloter l'algorithme d'attribution des tâches. Comment un agent décide-t-il de s'engager dans une tâche ? Comment décide-t-il de continuer à exécuter la tâche en cours ou de l'interrompre ?

Dans cet article, nous décrivons un modèle capable d'attribuer à des agents des tâches sans lien direct avec les stimulus. Nous utilisons notamment la motivation intrinsèque de l'agent pour décider de continuer ou d'interrompre une tâche. Nous présentons ensuite une application de ce modèle à une colonie d'abeilles simplifiée. Dans les colonies d'abeilles, plusieurs tâches sont dissociées de tout stimulus, ce qui représente une application intéressante du modèle. Par exemple, les larves d'abeilles n'émettent aucun stimulus pour demander à être nourries, mais un stimulus général qui est plutôt une représentation de la taille du couvain. Les abeilles qui nourrissent les larves (qui endossent un rôle de nourrice) doivent donc s'appuyer sur d'autres

mécanismes pour évaluer la nécessité de la tâche, et décider s'il faut continuer à chercher des larves ou chercher d'autres tâches plus prioritaires.

2 Travaux connexes

La division du travail se produit lorsque les agents doivent décider quelle tâche exécuter dans un environnement partagé. Les sociétés d'individus (ou d'agents) doivent trouver des moyens de répartir efficacement leur main-d'œuvre entre les tâches nécessaires pour survivre et s'étendre. En informatique, le contrôle décentralisé inspiré par les insectes sociaux a été étudié pendant des années et s'est avéré efficace dans de nombreuses applications. Dans cette section, nous allons passer en revue ce qui a été fait dans le domaine des modèles de répartition des tâches.

2.0.1 "Forage for Work" [6]

Dans ce modèle, les différentes tâches que les agents doivent accomplir sont dispersées dans des zones. Les agents, en recherche active de travail, tentent d'exécuter la tâche associée à leur zone ou se déplacent de manière aléatoire. Ainsi, les zones surpeuplées "poussent" les agents vers les zones voisines offrant du travail, ce qui entraîne une division du travail. Lorsque de nouveaux agents apparaissent dans une zone spécifique et que les agents les plus âgés meurent à un certain âge, ce modèle assez simple recrée le polyéthisme d'âges : les agents du même âge effectuent globalement les mêmes tâches. Les agents nouveaux prennent les offres de travail dans leur zone, repoussant ainsi les agents plus âgés. En suivant des règles simples, un agent peut se rediriger dans d'autres zones s'il ne trouve pas de travail à faire dans sa zone actuelle. Pour que ce modèle fonctionne, les agents doivent avoir la capacité d'évaluer la nécessité de réaliser chaque tâche. Ce pré-requis, ajoutée à l'hypothèse que les tâches sont dispersées dans des zones, font que ce modèle n'est pas adapté à la problématique, et doit être affiné. Il s'agit toutefois d'un bon point de départ.

2.0.2 FTM : "Fixed Threshold Model" [5]

Le FTM est basé sur des associations entre tâches et stimulus. Chaque tâche a un score, représentant sa priorité. Un agent s'engage puis exécute la tâche ayant la priorité la plus élevée. Le score de chaque tâche est calculé à partir de l'intensité du stimulus associé perçu par l'agent, généralement à l'aide d'une fonction sigmoïde. Soit T la tâche évaluée par l'agent, $F(T)$ le score de la tâche T , x_T le stimulus associé perçu par l'agent, n un entier pour la non-linéarité de la fonction (généralement $n = 2$ [7]) et Θ_T une constante, voici comment le score est calculé :

$$F(T) = \frac{(x_T)^n}{(x_T)^n + (\Theta_T)^n} \quad (1)$$

Θ est le *biais* de la fonction, et est utilisée pour l'ajuster tel que lorsque $x_T = \Theta_T$ alors $F(T) = 0.5$. Le biais Θ est utilisé pour modifier la perception des agents : avec un biais très faible, les agents sont très sensibles au stimulus associé et s'engagent dans la tâche plus tôt que les agents avec un biais plus élevé [8].

Chaque tâche a également une probabilité d'*interruption* évaluée à chaque pas de temps. Un agent peut avoir, par exemple, 0.5% de chances d'interrompre sa tâche en cours [9]. Lorsque c'est le cas, l'agent recherche une nouvelle tâche en utilisant les scores de chaque tâche et choisit la plus élevée. L'interruption est ici complètement aléatoire et ne reflète pas l'environnement ni les capacités des agents et nous pensons que de meilleures performances peuvent être atteintes avec un mécanisme d'interruption plus élégant.

2.0.3 RTM : "Response Threshold Model", Renforcement du biais

Sur la base du FTM et de l'équation 1, différents travaux des années 90 [10], [11] ont proposé de mettre en place des mécanismes de renforcement de la valeur Θ , en modifiant la sensibilité des agents pendant l'exécution, formant ainsi efficacement des spécialistes. Cette mise à niveau du FTM est plus généralement appelée "Response Threshold Model" (RTM).

Largement utilisé pour modéliser et piloter des simulations d'insectes sociaux, le RTM repose fortement sur l'association entre tâches et stimulus. Il suppose également que l'exécution de la tâche diminue le stimulus qui lui est associé, et que ne pas exécuter une tâche augmente son stimulus associé. Dans le cas contraire, les agents exécuteraient constamment cette tâche, ou du moins même jusqu'à ce qu'elle ne soit plus prioritaire. Le stimulus doit être une représentation de la priorité de la tâche qui lui est associée. Dans cet article, nous nous intéressons aux situations dans lesquelles ces hypothèses ne sont pas vraies. Nous décrivons dans la section suivante notre modèle basé sur le RTM et un mécanisme supplémentaire pour gérer ces situations.

3 Proposition : un Modèle d'Interruption pour la Répartition des Tâches

Nous proposons dans cette section un modèle basé sur les modèles à seuil RTM dans lequel les tâches, les activités et les actions sont définies au travers d'une architecture de subsomption hiérarchique. Notre modèle repose sur deux mécanismes :

- 1) le score de chaque tâche qui n'est pas associée à un stimulus se base sur l'état physique de l'agent (les outils qu'il possède, sa physiologie, ses caractéristiques physiques, etc.)
- 2) un mécanisme d'interruption, consistant en une évaluation systématique de toutes les tâches après chaque action, et en prenant en compte la *motivation interne* de l'agent dans l'évaluation de sa tâche courante.

3.1 Tâches, Subsomption et RTM

Brooks [12] a initialement introduit les architectures de subsomption comme un moyen de contrôler des robots, et celles-ci ont maintenant beaucoup d'applications, notamment dans les systèmes multi-agents. Cette architecture est organisée en couches superposées de comportement, chacune avec des conditions d'activation. Chaque comporte-

ment activé inhibe les comportements placés en-dessous. Les couches supérieures sont prioritaires et peuvent être vues comme des comportements prioritaires. Une architecture de subsomption hiérarchique permet d'avoir des couches de comportement qui sont elles-mêmes des architectures de subsomption, avec ses propres couches et conditions d'activation. Cela permet d'améliorer l'adaptabilité et la modularité des architectures classiques, et d'adapter les comportements à une complexité croissante [13].

Comme Heckel et al. l'ont montré, l'architecture de subsomption hiérarchique convient bien aux concepts de tâche, d'activité, d'action et de rôle, comme illustré sur la Figure 1. Nous donnons la définition de ces concepts (basée notamment sur [14], [15]) ici :

- une *action* est une partie ininterrompible d'un comportement qui dure un temps défini (court). Une action est dite coopérative si elle requiert un autre agent, exécutant lui-même une action donnée, pour se terminer. Nous définissons donc les actions coopératives comme des couples d'actions. Chaque action a une condition booléenne, qui lorsqu'elle est valide permet à l'agent d'exécuter l'action.
- une *activité* est un ensemble d'actions et/ou d'activités, structurée de façon hiérarchique. Les activités ont aussi une condition booléenne. Elles peuvent être vues comme les noeuds d'un arbre, les actions étant les feuilles.
- une *tâche* est un ensemble d'activités et d'actions, constituant un comportement particulier que les agents vont suivre. Chaque tâche du modèle est une implémentation du RTM, avec un seuil associé et une fonction (souvent une sigmoïde) pour calculer le score. Ce score est utilisé par l'agent pour trouver la tâche la plus prioritaire.
- un *rôle* est un ensemble *arbitraire* de tâches, d'activités ou d'actions. Ils ne suivent pas forcément la définition des tâches, mais permettent à un observateur (qui les définit, en général) de simplifier les comportements complexes qu'il observe.

3.2 Mécanisme d'Interruption et Motivation Intrinsèque

La première partie du mécanisme d'interruption consiste à utiliser la motivation intrinsèque pour aider l'agent à évaluer l'utilité de sa tâche actuelle, et à décider s'il doit la poursuivre ou en choisir une autre.

Pour les psychologues, la motivation est la source de l'action et un guide pour son exécution. Il existe deux types de motivation : extrinsèque, lorsqu'une récompense est offerte par l'environnement et intrinsèque qui n'a à voir qu'avec les besoins ou les attentes personnelles [16], comme le plaisir ou la curiosité. La motivation intrinsèque est utilisée dans l'intelligence artificielle et spécifiquement pour les systèmes d'apprentissage [17], *e.g.* pour aider ou guider les agents apprenants **baldassarre_intrinsèque_2013**. La motivation intrinsèque peut être divisée en de nombreux stimulus internes différents, tels que la faim ou la peur, qui

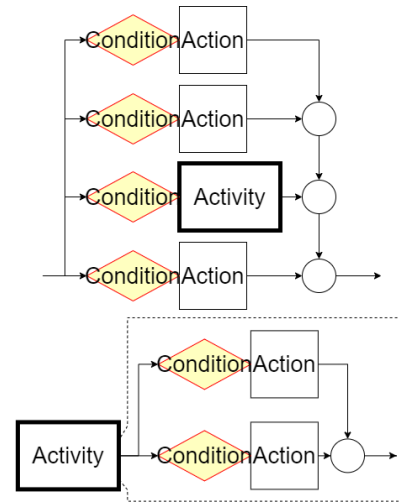


FIGURE 1 – Au-dessus, l'architecture de subsomption hiérarchique représente une tâche consistant en trois actions et une activité. En-dessous, on peut voir que cette activité est constituée de deux actions. Dans le reste de l'article, la partie la plus à droite de l'architecture ne sera plus représentée pour des raisons de simplicité.

déclenchent des comportements [18], proches de ce qui est décrit en Ethology **carbonell_multi-agent_1994**. Lorsque les stimulus internes sont classés comme des besoins, la motivation intrinsèque peut permettre une prise de décision à plus haut niveau. Selon la théorie du Flow [19], un agent en difficulté dans sa tâche ressent de l'anxiété et cherchera alors une tâche moins difficile. De la même manière, un agent qui accomplit une tâche facile s'ennuie et passera à des tâches plus difficiles [20]. L'idée de compétence apportée par Roohi et al. [21] est, pour un agent, le sentiment d'être en contrôle et capable d'accomplir sa tâche actuelle. Ainsi, un agent ayant un niveau de compétence trop faible cherche une tâche plus facile.

Dans notre modèle, les agents ne reçoivent aucune récompense de l'environnement, c'est pourquoi nous nous concentrons sur la motivation intrinsèque. Les tâches dans notre modèle peuvent être *motivées* ou non : les tâches sans stimuli représentatifs sont appelées motivées, car un agent doit utiliser sa motivation intrinsèque. En gardant l'idée de Flow à l'esprit, nos agents peuvent sentir si leur tâche motivée actuelle est utile pour l'ensemble du système. Cela peut être considéré comme une sorte de "Désir" dans le modèle Croissance-Désir-Intention **rao_bdi_1995**. Chaque fois qu'il commence une nouvelle tâche motivée, un agent fixe sa motivation à 1, comme étant pleinement motivé. Ensuite, l'agent exécute une action de cette tâche, qui peut être une **action démotivante**. Lors de l'exécution, une action démotivante diminue la motivation de l'agent. Chaque fois que l'agent a terminé une action, il calcule le score de toutes les tâches qu'il peut exécuter et remplace le score de sa tâche actuelle par sa motivation actuelle et choisit la tâche ayant le score le plus élevé, comme décrit ci-dessous. La baisse de la motivation de l'agent le rend plus réceptif à

d'autres tâches.

3.3 Agents et Évaluation des Tâches

Chaque agent est défini par son état interne, ses comportements et ses capacités d'action, à partir desquels il peut accomplir un ensemble de tâches, associées à des seuils. La figure 2 montre la prise de décision d'un agent à chaque cycle, avec une évaluation systématique de toutes les tâches disponibles. L'agent évalue d'abord si son action en cours est réalisée ou non. Si c'est le cas, l'agent sélectionne ensuite la tâche ayant la plus haute priorité, en calculant les scores de toutes les tâches réalisables à partir du modèle RTM, et en utilisant sa motivation actuelle comme score de la tâche en cours. L'agent interroge ensuite la subsomption de la tâche sélectionnée et récupère l'action suivante à exécuter, ainsi que sa durée et l'exécute. Une fois l'action exécutée, l'ensemble du processus est répété. La réévaluation de toutes les tâches réalisables à la fin de chaque action est la clé de l'adaptabilité des agents. Les seuils des tâches évoluent dans le temps et sont représentatifs des capacités de l'agent, tandis que la subsomption prend en charge l'approche logique du comportement. Ainsi, tant que les scores des autres tâches ne dépassent pas la motivation de l'agent, ce dernier exécutera la même tâche et progressera dans sa subsomption.

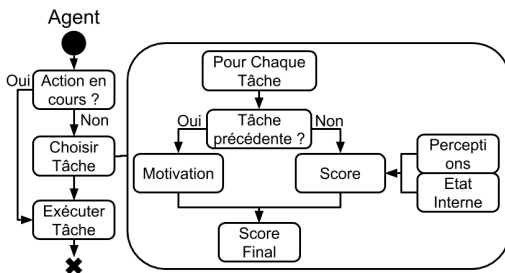


FIGURE 2 – Prise de décision de l'agent à chaque cycle.

3.4 Exemple d'application : Robotique en Essaim

Nous présentons ici un exemple d'application théorique portant sur la robotique en essaim, dans le but d'illustrer les mécanismes de notre modèle. Prenons de nombreux robots qui peuvent effectuer deux tâches différentes : la recherche d'énergie (collecte de minéraux dispersés autour de la base) et la patrouille autour de la base. Ils ont également ce que nous appelons des tâches d'entretien, comme le retour à la base pour faire le plein de carburant. L'architecture de subsomption de la tâche de recherche d'énergie est composée d'une action consistant en un mouvement aléatoire, dont la condition de déclenchement est "aucun minéral n'est en vue". C'est une action démotivante : chaque fois qu'un robot exécute cette action, il diminue sa motivation intrinsèque et le rend plus susceptible de choisir une autre tâche. En patrouille, l'action démotivante consiste à éviter un autre robot sur le chemin de la pa-

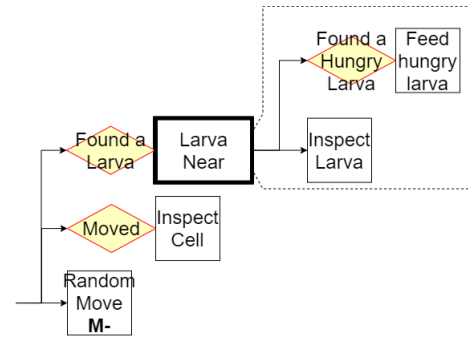


FIGURE 3 – Architecture de subsomption hiérarchique décrivant la tâche "nourrirLarve". L'action démotivante est représentée par le symbole "M-".

trouille. De cette façon, les robots patrouilleurs s'autorégulent.

Dans cet exemple, on pourrait imaginer que les robots utilisent des outils qu'ils peuvent prendre et déposer. Ces outils modifient ce qu'ils peuvent faire : ainsi, le fait de prendre un outil modifie le biais Θ pour toutes les tâches, reflétant l'état interne du robot. Par exemple, un robot qui ramasse une pioche voit son biais pour la tâche de recherche de ressource diminuer, ce qui maximise les probabilités que le robot exécute cette tâche. Une fois engagé dans la tâche de recherche de ressource, le fait de ne pas voir de champs de minéraux disponibles à récolter (non visibles ou déjà occupés par d'autres robots) diminue la motivation du robot. Avec une motivation suffisamment faible, le robot choisit une autre tâche qui peut l'obliger à changer d'outil.

4 Application : Simulation d'une colonie d'abeilles

Nous avons mis en place une version simplifiée d'une colonie d'abeilles domestiques en utilisant notre modèle. Les abeilles comptent sur de nombreux stimulus qui ne sont pas une représentation de la priorité d'une tâche. Par exemple, dans la tâche d'alimentation des larves, les larves émettent des phéromones, qu'elles aient faim ou non¹. D'autres tâches (comme la recherche de nourriture) ne sont liées à aucun stimulus externe. L'âge physiologique d'une abeille détermine la tâche clé qu'elle peut exécuter, car elle a besoin de certaines caractéristiques biologiques (glandes - c'est l'équivalent de l'"état interne" de la Fig 2 dans notre modèle).

En outre, l'âge physiologique est considéré comme un élément clé de l'auto-organisation des abeilles dans la ruche, car il peut être influencé par les échanges hormonaux et phéromonaux, comme décrit ci-dessous. L'objectif de cette

1. Les larves émettent continuellement une phéromone volatile appelée '*E-β-ocimène*', mais des travaux récents ont montré que les larves affamées en émettent davantage et attirent ainsi plus de nourrices [22]. Pourtant, on ne sait toujours pas si ce stimulus augmente l'alimentation des larves.

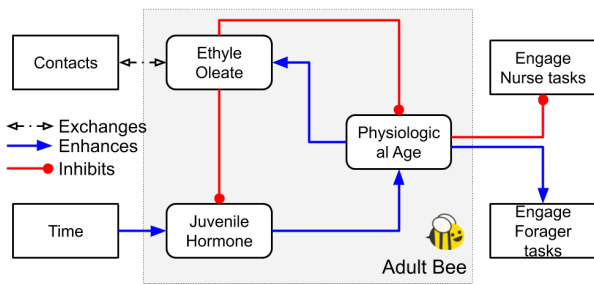


FIGURE 4 – La dynamique physiologique simplifiée de nos agents abeilles adultes. Nous pouvons voir les variables internes de l'abeille au milieu et les facteurs externes à gauche. Sur la droite, nous pouvons voir comment l'âge physiologique affecte la sélection des tâches.

application est donc double : premièrement, tester le rôle de l'âge physiologique dans la division du travail avec notre modèle et deuxièmement, évaluer l'effet du mécanisme d'interruption. Nous avons décidé de modéliser deux tâches clés au moyen de *tâches motivées* : l'alimentation des larves et la recherche de nourriture.

4.1 Tâches et Agents

Il existe deux types d'agents, les abeilles adultes et les larves. Les larves sont assez simples, elles acceptent la nourriture lorsqu'elles ont suffisamment faim, meurent lorsqu'elles ont trop faim et émettent une phéromone de contact appelée *Ethyle Oleate* (*EO*) à chaque pas de temps. Les abeilles adultes, en revanche, sont les agents qui doivent s'organiser et se répartir les deux tâches motivées suivantes : nourrir les larves et butiner à l'extérieur. Les abeilles adultes ont également quelques tâches d'entretien à effectuer, telles que se reposer ou demander/donner de la nourriture (voir tableau 1). Chacune de ces tâches a été décrite avec une architecture de subsomption hiérarchique (voir la figure 3 pour la description de la tâche "nourrir les larves").

4.2 Age Physiologique et Répartition du Travail

Dans notre modèle, les scores des tâches motivées sont calculés à partir de l'état physique actuel de l'agent (voir section 3.3). C'est particulièrement adapté à la modélisation des abeilles : l'âge physiologique d'une abeille détermine la tâche motivée qu'elle peut exécuter. Nous insistons sur la notion d'âge **physiologique**, car les abeilles peuvent *baisser* leur âge au cours de leur vie, ou *accélérer* leur vieillissement. Les abeilles plus âgées ont tendance à s'engager davantage dans des tâches de recherche de nourriture, et les jeunes abeilles sont plus susceptibles de nourrir les larves. L'âge physiologique d'une abeille est donné par la quantité d'*hormone juvénile* (*HJ*) dans son système, qui augmente naturellement avec le temps. L'hormone juvénile guide le développement physiologique des abeilles : une jeune abeille ayant une faible quantité d'hormone juvénile possède les glandes nécessaires pour nourrir les larves,

et une abeille plus âgée a la capacité de voler pour aller chercher de la nourriture ainsi que les glandes pour traiter le pollen et le nectar. Les abeilles adultes ont naturellement tendance à aller chercher leur nourriture à l'extérieur, poussées par leur augmentation interne de *HJ*, mais elles sont maintenues à l'intérieur par le couvain, la reine et les abeilles déjà butineuses émettant l'*EO* (échangé par contact) [23]. Nous avons résumé ces interactions Figure 4. Cette lutte constante de l'âge physiologique permet d'équilibrer la main-d'œuvre entre ces deux grandes tâches. Si les abeilles butineuses meurent à l'extérieur, les jeunes abeilles de l'intérieur percevront moins d'*EO* et certaines d'entre elles recommenceront à vieillir, ce qui rétablira l'équilibre. En réalité, une ouvrière peut aller butiner dès l'âge de 5 jours mais commence généralement ses activités extérieures vers 20. La perte de larves est rare : les ouvrières régulent la façon dont la reine pond des œufs en adaptant la façon dont elles la nourrissent. Lorsqu'elles perçoivent un faible apport de ressources, elles peuvent réduire la ponte de la reine. Comme nous n'avons pas modélisé la reine, la perte de larves se produira et reflètera l'efficacité de la colonie à répartir sa main-d'œuvre. De plus, le fait que les travailleurs doivent diminuer leur âge physiologique est également un événement rare. Nous ciblons spécifiquement ces cas intéressants, où la colonie est privée d'abeilles adultes plus jeunes.

4.3 Environnement et Simulateur

L'environnement est une grille hexagonale 2D de taille 30x30. Chaque cellule a un contenu (une larve, de la nourriture, etc.) et peut être occupée par une abeille adulte. Les stimulus sont gérés au niveau de la cellule, où chaque cellule contient une quantité de chaque stimulus. Les stimulus sont modélisés selon deux paramètres majeurs : la propagation (leur comportement dans l'espace) et la dissipation/évaporation (leur comportement dans le temps). Lorsqu'il émet un stimulus, un agent augmente la quantité de ce stimulus donné sur sa cellule. Ensuite, à chaque pas de temps et pour chaque stimulus, chaque cellule calcule sa nouvelle valeur en utilisant les valeurs de ses voisins et la sienne. Les paramètres que nous avons utilisés pour toutes nos simulations sont inspirés de la biologie mais calibrés pour accélérer les simulations. Les effets hormonaux et phéromonaux peuvent mettre des jours à se produire, tandis que les abeilles interagissent et errent dans la ruche en quelques secondes. Nous avons décidé d'accélérer certains effets pour obtenir des résultats dans un temps de simulation raisonnable (accélération de x4000 à x6000). Par exemple, les larves sont nourries environ toutes les 2 heures [24], mais nous avons accéléré ce temps à 2 secondes. Cette accélération introduit un biais, car les changements à long terme se situent maintenant dans la même échelle de temps que les changements rapides, mais ne changeront pas l'aspect essentiel de ce que nous démontrons.

4.4 Expérimentations et Résultats Attendus

Nous avons fait plusieurs expériences pour évaluer comment ce modèle peut s'adapter aux changements tout en

TABLE 1 – Les tâches de l’application. Certains scores de tâches sont calculés à l’aide de la sigmoïde du RTM, mais d’autres déjà mappées dans [0;1] n’ont pas besoin d’utiliser de sigmoïde. Une tâche non motivée se comporte exactement comme dans un RTM classique.

Nom de Tâche	Stimulus d’entrée	Score
Tâches d’entretien		
RestTask	Energy	1-Energy
AskFoodTask	Hunger	Hunger
GiveFoodTask	AskFoodStimulus	Sigmoid(Eq 1) avec le biais à 0.5
RandomMoveTask	-	0.2
Tâches motivées		
ForagerTask	-	Sigmoid (Eq 1) biais : 1-HJ mappé dans [0.3;1]
FeedLarva	-	Sigmoid (Eq 1) biais : HJ mappé dans [0.3;1]

TABLE 2 – Analyse statistique ANOVA (N=50). Une ligne pour chaque scénario. Les deux premières colonnes représentent l’impact sur les scores de la simulation de l’activation des effets physiologiques, avec notre mécanisme d’interruption contourné (scores PhysiologyOn/InterruptionOff - scores PhysiologyOff/InterruptionOff). Les deux dernières colonnes représentent l’impact de l’activation de notre mécanisme d’interruption tout en laissant l’effet biologique activé. Chaque impact est mesuré par le score des nourrices (combien de larves ont survécu) et le score des butineuses (combien de temps a été consacré à la recherche de nourriture).

	Ajouter la physiologie sans l’interruption		Ajouter l’interruption avec la physiologie	
	Score des Nourrices	Score des Butineuses	Score des Nourrices	Score des Butineuses
S1	+3% ($p < 0.001$)	-30% ($p < 0.001$)	<i>Non significatif</i>	-11% ($p < 0.001$)
S2	<i>Non significatif</i>	0 à 150k ($p < 0.001$)	<i>Non significatif</i>	-8% ($p < 0.001$)
S3	0% à 68% ($p < 0.001$)	<i>Non significatif</i>	+37% ($p < 0.001$)	<i>Non significatif</i>
S4	-32% ($p < 0.001$)	-29% ($p < 0.001$)	+45% ($p < 0.001$)	+35% ($p < 0.001$)
S5	+10% ($p < 0.001$)	-77% ($p < 0.001$)	<i>Non significatif</i>	-14% ($p = 0.007$)

assurant la répartition des tâches. De nombreuses expériences réalisées dans la littérature consistent à modifier la répartition initiale des tâches ou de l’environnement [9], [25]. Notre premier objectif est de nous assurer que l’équilibre final n’est pas influencé par les conditions initiales. De plus, l’équilibre doit être modifié par le ratio abeille adulte / larve. En mettant plus de larves par ouvrière, nous devrions nous attendre à avoir plus d’ouvrières consacrées à l’alimentation des larves et vice versa. Ensuite, nous évaluerons comment le mécanisme d’interruption modifie le système. Nous avons donc deux hypothèses :

- **H1** : Notre modèle avec la physiologie de l’abeille et les phéromones permet de réaliser la répartition des tâches.
- **H2** : Notre mécanisme d’interruption améliore les performances du système, ce qui améliore la répartition des tâches. Ici, un meilleur rendement signifie maximiser le temps passé à chercher de la nourriture ET le taux de survie des larves.

Nous avons créé 5 scénarios que nous utiliserons pour évaluer ces hypothèses. Chaque scénario est ensuite simulé avec différentes conditions, en contournant les effets phéromonaux pour évaluer **H1** ou/et en contournant notre mécanisme d’interruption pour évaluer **H2**. Le contournement des effets phéromonaux gèle l’âge physiologique des agents, et le contournement d’une partie du mécanisme d’interruption se fait en considérant toutes les tâches

comme non motivées :

- **Scénario 1** : Répartition initiale aléatoire des âges avec 150 abeilles adultes et 150 larves.
- **Scénario 2** : 150 adultes et 150 larves mais toutes les abeilles adultes commencent comme nouvelles.
- **Scénario 3** : 150 adultes et 150 larves, mais toutes les abeilles adultes commencent comme âgées.
- **Scénario 4** : Répartition initiale aléatoire des âges avec 150 adultes et 50 larves.
- **Scénario 5** : Répartition initiale aléatoire des âges avec 150 adultes et 300 larves.

4.5 Résultats et Analyse Statistique

Avec notre modèle complet, S1, S2 et S3 se sont terminés avec environ 60% des larves de la colonie qui se nourrissaient et environ 20% qui cherchaient de la nourriture. S4 et S5, par rapport à S1, ont changé l’équilibre. Comme nous l’avons vu précédemment, l’équilibre pour un ratio abeille adulte par larve de 1/1 (150 abeilles adultes pour 150 larves, S1) nous a donné un équilibre à environ 60% de larves en train de se nourrir. Un ratio de 1/2 (S5) l’élève à près de 90%. Un rapport de 3/1 (S4) le fait baisser à 20%. La figure 5 montre côte à côte deux graphiques comparant les différents ratios d’abeilles nourrissant les larves parmi les scénarios. Nous pouvons voir sur le graphique le plus à gauche que S1, S2 et S3 convergent vers le même équilibre, et que S4 et S5 ont respectivement un équilibre infé-

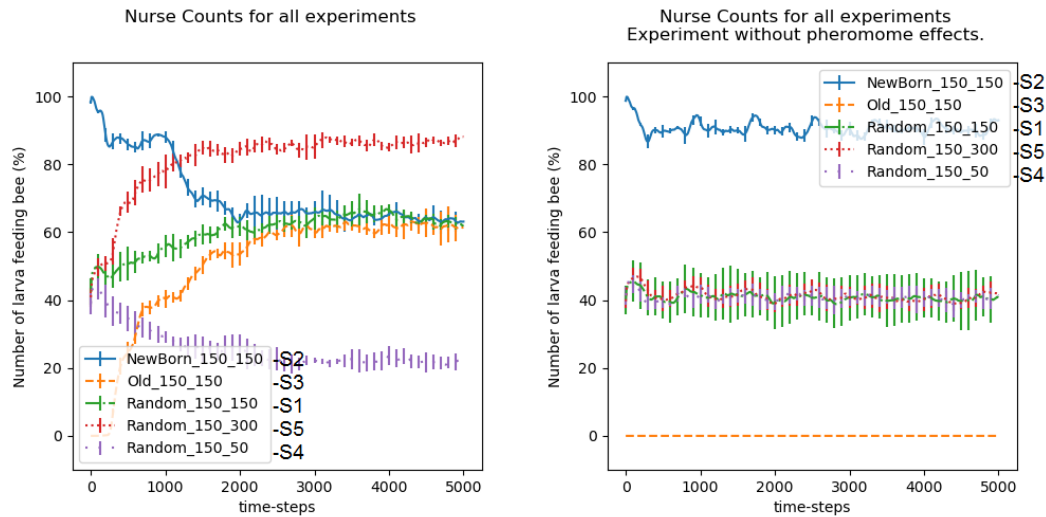


FIGURE 5 – Voici les différents ratios de d’abeilles nourrices de toutes les expériences (scénarios) dont nous avons parlé, avec et sans le contournement de la physiologie.

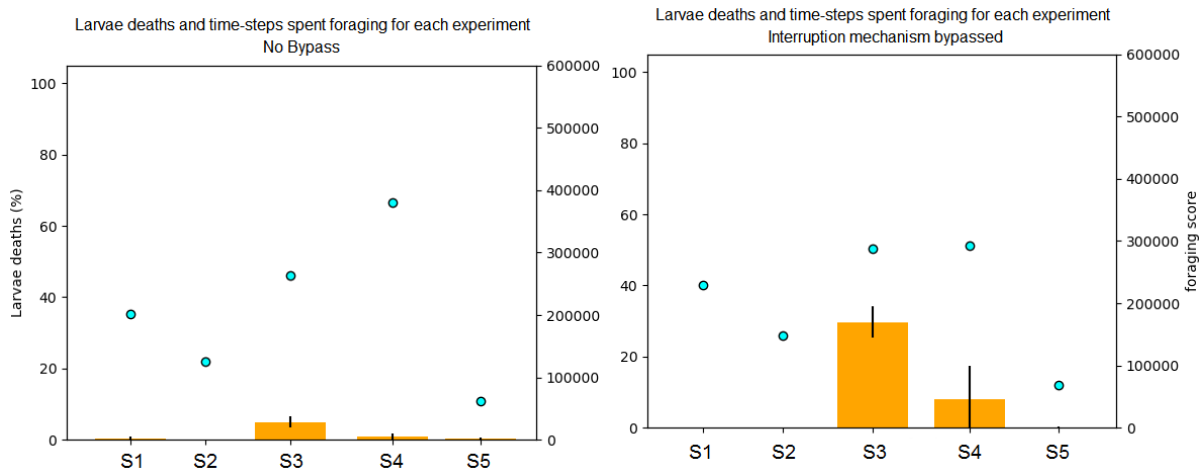


FIGURE 6 – Ces graphiques montrent la moyenne des décès de larves et le temps passé à butiner pour chaque expérience répétée 5 fois. Les graphiques de gauche et de droite représentent les expériences respectivement avec et sans mécanisme d’interruption.

rieur et supérieur aux trois premiers. Dans le graphique de droite, où la physiologie de l’abeille a été court-circuitée, nous voyons que le système ne s’adapte pas aux scénarios. Une analyse ANOVA (Physiologie x Interruption, N=50) a montré que dans certains scénarios, notre mécanisme d’interruption fait que le système obtient des scores plus faibles, alors que dans d’autres, c’est le contraire : Le tableau 2 donne un aperçu de ces changements statistiquement significatifs, et le lecteur trouvera sur la figure 6 une comparaison graphique du contournement de l’interruption. D’une part, S1, S2 et S5 sont légèrement affectés négativement par le mécanisme d’interruption (recherche de nourriture 10% moins efficace) alors que les soins aux larves sont aussi efficaces. D’autre part, dans S3 et S4, le mécanisme d’interruption a amélioré les scores de manière significative (entre 35 et 45%).

4.6 Discussions

Nous avons démontré que **H1** est vrai en changeant la condition de départ et le ratio abeille par larve. Cependant, l’analyse statistique que nous avons fournie ne nous permet d’affirmer que **H2** est vrai que pour S3 et S4, où la plus grande adaptabilité est requise. La vitesse à laquelle une tâche démotivante entraîne un changement de tâche est un paramètre clé. Des travaux supplémentaires doivent être effectués pour le calibrer, mais les résultats préliminaires montrent qu’un changement de tâche plus rapide (jusqu’à un certain point) entraîne de meilleurs scores globaux, au détriment de la stabilité individuelle. Nous tenons également à souligner que la réévaluation systématique de toutes les tâches après l’achèvement de chaque action ne pouvait pas être contournée.

5 Conclusion et Perspectives

Nous avons proposé ici un modèle basé sur le modèle RTM, l'état interne des agents et la motivation intrinsèque pour effectuer la répartition des tâches dans un environnement où toutes les tâches ne sont pas associées à des stimulus représentant leur priorité. Modéliser les tâches avec une architecture de subsomption hiérarchiques et les discriminer en utilisant une approche RTM nous permet de piloter le comportement de nos agents. Les tâches contiennent des actions démotivantes qui représentent des comportements frustrants, comme la recherche de quelque chose. L'exécution d'une action démotivante diminue la motivation de l'agent à poursuivre sa tâche actuelle, le rendant plus sensible à d'autres tâches. Lors du choix d'une nouvelle tâche, si la tâche actuelle de l'agent est motivée, son score RTM est remplacé par la motivation actuelle de l'agent. Chaque fois qu'un agent change sa tâche actuelle pour une tâche motivée, sa motivation est ramenée à 1, comme étant pleinement motivé. Nous avons montré expérimentalement que notre modèle peut atteindre l'équilibre, répondre à des conditions changeantes et les travaux préliminaires montrent qu'il améliore l'efficacité du système. Notre mise en œuvre de ce modèle pour une colonie d'abeilles simplifiée montre des capacités de répartition du travail. L'équilibre est atteint et ne dépend pas des conditions initiales, mais dépend du ratio abeille par larve. Nous avons l'intention d'améliorer la mise en œuvre actuelle en ajoutant de la complexité vers une mise en œuvre aussi proche que ce que nous comprenons actuellement sur le fonctionnement des colonies d'abeilles, en testant davantage le modèle et ses capacités. Nous comparerons ensuite le comportement de cette mise en œuvre avec les observations réelles de colonies d'abeilles. En parallèle, nous mettons également en œuvre l'exemple d'essaim de robot discuté dans la section 3.4 pour évaluer la modalité de notre proposition.

Les codes sources de l'implémentation (java) et des scripts (python) utilisés pour l'analyse des résultats (with JASP) sont sur GitHub : <https://github.com/Kwarthys/BeeKeeper>.

Références

- [1] S. N. BESHES et J. H. FEWELL, "Models of division of labor in social insects.", p. 413-440, 2001.
- [2] T. SCHMICKL et I. KARSAI, "Integral feedback control is at the core of task allocation and resilience of insect societies", 2018.
- [3] R. E. MORLEY et C. SCHELBERG, "An analysis of a plant-specific dynamic scheduler", 1993.
- [4] P. ZAHADAT et T. SCHMICKL, "Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition", 2016.
- [5] E. BONABEAU, M. DORIGO et G. THERAULAZ, *From Natural to Artificial Swarm Intelligence*. 1999.
- [6] N. R. FRANKS et C. TOFTS, "Foraging for work : how tasks allocate workers", p. 470-472, 1994.
- [7] T. SCHMICKL et K. CRAILSHEIM, "TaskSelSim : a model of the self-organization of the division of labour in honeybees", t. 14, n° 2, p. 101-125, 2008.
- [8] A. DORNHAUS, F. KLÜGL, F. PUPPE et J. TAUTZ, "Task Selection in Honeybees - Experiments Using Multi-Agent Simulation", p. 13,
- [9] T. SCHMICKL et K. CRAILSHEIM, "Analysing honeybees' division of labour in broodcare by a multi-agent model", 2008.
- [10] G. THERAULAZ, E. BONABEAU et J.-L. DENEUBOURG, "Response threshold reinforcements and division of labour in insect societies", 1998.
- [11] A. DROGOUL et J. FERBER, "Multi-agent simulation as a tool for modeling societies : Application to social differentiation in ant colonies", in, 1994.
- [12] R. BROOKS, "A robust layered control system for a mobile robot", p. 14-23, 1986.
- [13] F. W. P. HECKEL, G. M. YOUNGBLOOD et N. S. KETKAR, "Representational complexity of reactive agents", 2010, p. 257-264.
- [14] A. CAMPBELL et A. S. WU, "Multi-agent role allocation : issues, approaches, and multiple perspectives", 2011.
- [15] C. ANDERSON et N. R. FRANKS, "Teamwork in Animals, Robots, and Humans", 2003.
- [16] R. M. RYAN et E. L. DECI, "Intrinsic and Extrinsic Motivations : Classic Definitions and New Directions", p. 54-67, 2000.
- [17] J. SCHMIDHUBER, "Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990-2010)", p. 230-247, 2010.
- [18] P. MAES, "The agent network architecture (ANA)", 1991.
- [19] M. CSIKSZENTMIHALYI, *Finding flow : The psychology of engagement with everyday life*. 1997.
- [20] M. CORNUDELLA, P. VAN EECKE et R. van TRIJP, "How Intrinsic Motivation can Speed Up Language Emergence", 2015, p. 571-578.
- [21] S. ROOHI, J. TAKATALO, C. GUCKELSBERGER et P. HÄMÄLÄINEN, "Review of Intrinsic Motivation in Simulation-based Game Testing", 2018, p. 1-13.
- [22] X. J. HE et AL., "Starving honey bee (*Apis mellifera*) larvae signal pheromonally to worker bees", 2016.
- [23] Y. LE CONTE, A. MOHAMMEDI et G. E. ROBINSON, "Primer effects of a brood pheromone on honeybee behavioural development", 2001.
- [24] M. L. WINSTON, *The biology of the honey bee*. 1991.
- [25] T. D. SEELEY, *The wisdom of the hive : the social physiology of honey bee colonies*. 1995.