

An Interruptible Task Allocation Model

Application to a Honey Bee Colony Simulation

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

¹ Univ Brest, Lab-STICC, CNRS, UMR 6285, Brest, France
`{firstname.surname}@univ-brest.fr`

² INRAE, UR 406 Abeilles et Environnement, Avignon, France
`cedric.alaux@inrae.fr, yves.leconte@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

Abstract. Division of labour is a key aspect of distributed systems, such as swarm robotics or multi-agent systems. Inspired by social insects known for their task allocation capabilities, most of the models rely on two assumptions: 1) each task is associated with a stimulus, and 2) the execution of this task lowers that stimulus. In short, the stimulus is a representation of the amount of work needed on a task. When these assumptions are not true, we need a mechanism to guide the agent in its decision whether to pursue or to interrupt its current task, as there is no diminishing stimulus to rely on. In this article, we propose a model based on the Response Threshold Model and a mechanism based on the agent's intrinsic motivation and internal states, allowing to take into account tasks dissociated from stimuli. Agents use their intrinsic motivation to emulate the priority of tasks not associated with any stimuli, and to decide whether to interrupt or pursue their current task. This model has been applied to simulate the division of labour within a simplified honey bee colony, associated with the constantly adapting physiology of honey bees. Preliminary results show that the task allocation is effective, robust and in some cases improved by the interruption mechanism.

Keywords: Agent-based simulation · Task Allocation · Self-Organisation.

1 Introduction

The ability of social insects to distribute their workforce without any central control has been studied for years. This self-organisation ability to dynamically adapt to its environment allows social insects to be robust to changes and still be able to thrive. In computer science, and more specifically in multi-agent systems, task allocation models adapted from social insects are numerous and effective, applied for example in agent-based simulations [3, 22] and in general problem resolution [16, 28]. These models mainly rely on two assumptions: 1) Each task is associated with a stimulus and 2) The execution of this task lowers its stimulus

[4]. Thus, internal or external (in the environment) stimulus perceived by an agent triggers the execution of the associated task, lowering the intensity of the stimulus and the probability of doing that same task again. Hence the perceived stimulus can be seen as a representation of the task work needed. However, in some cases, the execution of a task does not affect the amount of the associated perceived stimulus, or there is even a complete lack of association between a task and a stimulus. In that case, we need to find other ways to drive the task allocation algorithm. How can an agent decide whether to keep executing its current task or to interrupt it? In this article, we describe a model able to allocate tasks with no direct connection to stimuli to agents. We notably use agent’s intrinsic motivation to decide whether to interrupt a task. We then present an implementation of that model applied to a simplified honey bee colony. In honeybee colonies, there are several tasks dissociated from any stimuli, thus representing an interesting application for the model. For example, the larvae do not emit any stimuli to ask to be fed, but a general stimulus that is more a representation of the brood size. Larva-feeding bees (endorsing a nurse *role*) then have to rely on other mechanisms to evaluate the task priority and to decide whether to keep looking for larvae or seeking other jobs.

2 Related Work

Division of labour occurs when agents have to decide which task to execute in a shared environment. Societies of individuals (or agents) have to find ways to distribute their workforce effectively amongst tasks needed to thrive and survive. In computer science, decentralised control inspired by social insects has been studied for years and has proved to be effective in many applications. In this section, we oversee here what has been done in the field of task allocation models.

In the **Forage for Work** [11] model, the different tasks are scattered in zones. Agents in a given zone try to perform the zone’s associated tasks or move randomly. Thus, crowded zones “push” agents to neighbouring zones offering work, resulting in a division of work. When new agents arise in a specific zone and older agents die at a certain age, this rather simple model can recreate age polyethism: agents of the same age are globally doing the same tasks. Newborn agents are taking work offers where they spawned, effectively pushing the older ones away. Following simple rules, an agent can redirect itself in other zones if it cannot find any work to do in its current one. Here, agents have direct access to each task needs. On the other hand, the assumptions that the tasks are scattered in zones and that their stimulus is a representation of the needed work make this model not adapted to the problematic, and need to be refined.

The **Fixed Threshold Model (FTM)** [4] is based on associations between tasks and stimuli. Agent should always execute the task with the higher priority, computed from its score. Each task score is calculated from the intensity of the associated stimulus perceived by the agent, usually computed using a sigmoid. Let T be the task evaluated by the agent, $F(T)$ the score of the task T , x_T the task’s associated perceived stimulus, n an integer for the non-linearity of the

function (usually $n = 2$ [20]) and Θ_T a constant named bias used to tweak the function such as when $x_T = \Theta_T$ then $F(T) = 0.5$. The score is calculated with:

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

The bias Θ is used to alter the perception of the agents. With a given bias, agents are sensitive to the associated stimulus and engage their task earlier than higher bias agents [9]. Each task then has an interruption probability that is randomly tested at each time-step [21]. Then the agent searches a new task using the scores of each task and picks the higher one. Interruption is here completely random and does not reflect the environment nor the agents' capabilities. We believe that better performances can be reached with a more elegant interruption mechanism.

The **Response Threshold Model: Threshold Reinforcement** is based on the FTM. Different works in the 90s [26, 10] proposed to apply some reinforcement to the value Θ , changing the sensitivity of the agents inline. This upgrade of FTM is called a Response Threshold Model (RTM). Widely used to model and drive social insects simulations, it strongly relies on the association between tasks and stimuli in a one to one manner. It also assumes that the execution of the task lowers it's associated stimulus. Otherwise, the agents would execute that task forever. In this article, we are interested in situations where those assumptions are not true. We describe in the next section our model based on RTM and an added mechanism to handle those situations.

3 Proposition: An Interruptible Task Allocation Model

We propose in this section a model based on the Response Threshold Model, in which tasks, activities and actions are defined through a hierarchical subsumption architecture. Our model relies on two mechanisms: 1) the score of each task not associated with any stimuli is based on the agent's current physical state (tools in hand, physiology, physical traits etc..) and 2) An interruption mechanism consisting in: the consideration of the agent's internal motivation in the evaluation of the current task and a systematic evaluation of every available tasks after finishing an action.

3.1 Tasks, Subsumption and RTM

Subsumption architectures have been introduced by Brooks as a way to control robots [5] and are now used in plenty of fields including multi-agent systems. This architecture is organised in stacked layers of behaviour, where each activated behaviour inhibits all the behaviours below it. Topmost layers are prioritised and can be seen as reflexes. Hierarchical subsumption allows a subsumption layer to be another subsumption systems, with layers and conditions. This nesting further improves the adaptability and modality of subsumption system and scales well with increasing complexity [13]. Heckel et al. have shown that this architecture suits well the following definition of the key concepts of task, activity, action and

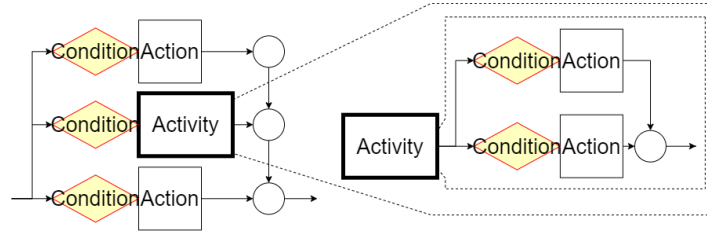


Fig. 1. The hierarchical subsumption architecture represents a task consisting of 2 actions and one activity; the latter consists of 2 other actions, seen on the right.

role (based on other’s works [6, 1]). Fig 1 shows how tasks, activities and actions are defined with a hierarchical subsumption architecture.

An **action** is an uninterruptible piece of work that lasts for a given time. Actions can be cooperative, meaning that they require another agent executing another given action to complete both. Each action has a boolean condition, stating whether it can be executed, as it is a part of a subsumption block.

An **activity** is a set of actions and/or activities building the hierarchical subsumption. Activities also have a boolean condition. Activities can be seen as nodes of a tree where actions are the leaves.

A **task** is a set of piece of work that the agents have to do. Each task of the model is an implementation of the RTM, with an associated threshold and a sigmoid function to compute a score. This score is used by the agent to find the highest priority task.

A **role** is seen as an arbitrary set of tasks, activities or actions. It does not have to match the task’s distribution and is more suited to the observer for simplifying the complex mechanisms in place.

3.2 Intrinsic Motivation as Part of the Interruption Mechanism

The first part of the interruption mechanism consists in the use of the intrinsic motivation to help the agent evaluate the usefulness of its current task, and decide whether to pursue it or to choose another one.

For psychologists, motivation is the root of action and a guide for its execution. There are two types of motivation: extrinsic, when a reward is offered by the environment and intrinsic which only has to do with personal needs or expectations [19], like fun or curiosity. Intrinsic Motivation is used in Artificial Intelligence and specifically for learning systems [23], *e.g.* to help or guide learning agents [2]. Intrinsic motivation can be split up into many different internal stimuli, such as hunger or fear, that trigger behaviours [15], close to what is described in Ethology [10]. When internal stimuli are classified as needs, intrinsic motivation can allow more high-level decision making. Based on the Flow theory by Csikszentmihalyi [8], an agent struggling in his task feels anxiety and seeks a less difficult task. In the same way, an agent completing an easy task gets bored and moves to more difficult tasks [7]. The competence idea brought by Roohi et

al. [18] is, for an agent, the feeling of being in control and able to complete its current task. As such, an agent with a competence level too low seeks an easier task.

In our model, agents do not receive any reward from the environment, so we focus on intrinsic motivation. Tasks in our model can be either *motivated* or not: tasks with no representative stimuli are called motivated, as an agent has to use its intrinsic motivation. Keeping the idea of Flow in mind, our agents can sense whether their current motivated task is useful for the overall system. This can be thought of as a sort of “Desire” in the Belief-Desire-Intention model [17]. Whenever starting a new motivated task, an agent sets its motivation to 1, as fully motivated. Then, the agent executes an action of this task, which may be a **demotivating action**. Upon execution, a demotivating action decrements the agent’s motivation. Whenever the agent has finished an action, it computes the score of every tasks it can perform and replaces the score of its current task by its current motivation and picks the higher scored task, as described below. Lowering the agent’s motivation makes it more receptive to other tasks.

3.3 Agents and Task Evaluation

Each agent is defined by its internal state, behaviours and capabilities. From these, each agent can achieve a set of tasks, associated with thresholds. Fig 2 shows the decision making of an agent at each cycle, with a systematic evaluation of every available tasks. The agent evaluates first whether its current action is done or not. If it is, the agent selects the task with the highest score. Using the subsumption architecture associated with the selected task, the agent then retrieves the next action and executes it for its full duration. Once done, the whole process is repeated. The re-evaluation of every task at each action completion is key to agents’ adaptability. The tasks’ thresholds evolve through time and are representative of the agent abilities, whereas the subsumption takes care of the logical approach.

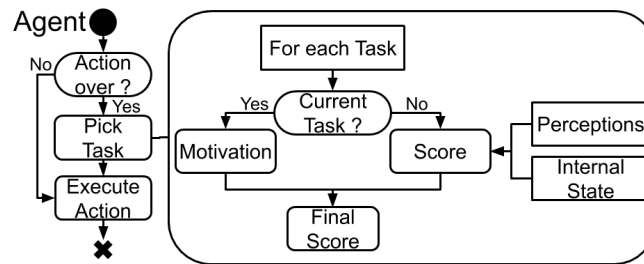


Fig. 2. Decision making of the agent at each cycle.

3.4 Swarm Robotics Example

We present here a theoretical example with swarm robotics to illustrate the mechanisms of our model. The example consists of many robots that can per-

form two different tasks: foraging (gathering minerals spread around the base), and patrolling around the base. They also have what we call upkeep tasks, such as returning to the base to refuel. The subsumption architecture of the foraging task includes a random movement when no mineral is in sight. This action is a demotivating action: each time a robot executes this action, it lowers its intrinsic motivation and makes it more susceptible to choose another task. When patrolling, the demotivating action is the avoidance of another patrolling bot on the patrol path. This way, patrolling bots auto-regulate themselves.

Robots could use tools that they can pick up and drop-down. Tools modify what robots can do, so picking up a tool changes the bias for all tasks, reflecting the bot internal state. For example, a bot picking up a pickaxe sees its bias for the foraging task drop-down. Once engaged in the foraging task, not seeing any available mineral fields to harvest (not in sight or already occupied by other bots) lowers the robot’s motivation. With a low enough motivation, the robot chooses another task that may require it to change tool.

4 Application: Simulation of a Honey Bee Colony

We implemented a simplified version of a honey bee colony using our model. Bees (are thought to) rely on many stimuli that are not a representation of the priority of a task⁵. For example, in the feeding larva task, larvae emit pheromones no matter how hungry they are⁶. Other tasks (like foraging) are not linked to any external stimuli. The physiological age of a bee determines what key task it can execute, as it needs some biological features (glands - this is equivalent to the “Internal State” of Fig 2 in our model). Moreover, the physiological age is thought to be a key component of the self-organisation of bees within in the colony, as it can be influenced by hormonal and pheromonal exchanges, as described below. The purpose of this application is then twofold: first, test the role of the physiological age in the division of labour with our model and second, to evaluate the effect of the interruption mechanism. We decided to model two key tasks by the mean of *motivated tasks*: feeding larvae and foraging.

4.1 Agents and Tasks

There are two kinds of agents, adult bees and larvae. Larvae accept food when hungry enough, die when too hungry and emit a contact pheromone called *Ethyle Oleate (EO)* at each time-step. Adult bees, on the other hand, are the agents that have to organise themselves using the *Juvenile Hormone (JH)* in their body to execute the following two motivated tasks: feed the larvae and forage

⁵ The source code of the implementation (java), the table of parameters used in the experiments and the scripts (python) used to conduct the statistical analysis (with JASP) can be found on GitHub: <https://github.com/Kwarthys/BeeKeeper>.

⁶ Larvae continuously emit a volatile pheromone called “*E-β-ocimene*”, but recent work has shown that hungry larvae emit more of it and thus attract more workers [12]. Yet, it is still unclear if this stimulus increases the feeding of the larvae.

outside. Adult bees also have few upkeep tasks to perform, such as resting or asking/searching for food (see Table 1). Each of these tasks has been described by a hierarchical subsumption architecture (see Figure 3 for an example).

Table 1. Tasks executed by our adult bee agents. Some tasks are computed using the RTM sigmoid (Eq 1), but others already mapped in $[0;1]$ do not need the use of a sigmoid. A non-motivated task behaves exactly as in a classical RTM.

Task Name	Input Stimulus	Score
Upkeep tasks		
RestTask	Energy	1-Energy
AskFoodTask	Hunger	Hunger
GiveFoodTask	AskFoodStimulus	Sigmoid with bias at 0.5
RandomMoveTask	-	0.2
Motivated tasks		
ForagerTask	-	Sigmoid with bias: 1-JH remapped in $[0.3;1]$
FeedLarva	-	Sigmoid with bias: JH remapped in $[0.3;1]$

In our model, the scores of the motivated tasks are computed from the current physical state of the agent (see section 3.3). It is particularly true when modelling honeybees: the physiological age of a bee determines what motivated task it can execute. We are insisting on the **physiological** age, as bees can *lower* their age during their lifetime, or *accelerate* their ageing. The physiological age of a bee is given by the amount of (JH) in its system, that naturally increases with time. JH guides the physiological development of honeybees: a young bee with low amounts of JH has the required glands to feed the larvae, and an older bee has the ability to fly to go forage and the glands to process pollen and nectar [24]. Adult bees naturally tend to go foraging outside, pushed by their increasing internal JH , but are kept nursing by the brood, the queen and already foraging bees’ EO (exchanged by contact) [14]. We summed up those interactions Figure 4. This constant fight of physiological age can nicely balance the workforce between those two major tasks. If foraging bees die outside, less EO would be perceived by the young bees inside and some of them would start to age again, restoring the balance.

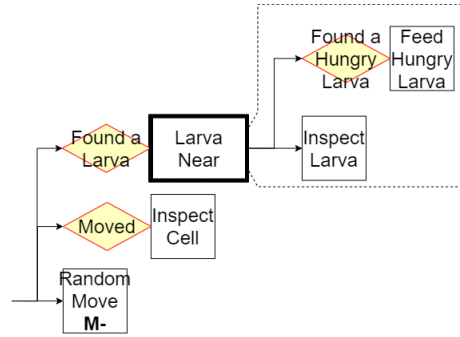


Fig. 3. Hierarchical subsumption describing the “feedLarvae” task. The demotivating action is represented with a “M-”

In nature, a worker can go forage as soon as it is 5 days old but generally starts outside activities around 20. Larvae loss is rare: workers regulate how the queen lay eggs by regulating how they feed her. When they perceive a low resource input, they can almost stop the queen from laying eggs. As we did not model the queen, larva loss will occur and will reflect the efficiency of the colony to allocate its workforce. Moreover, workers having to decrease their physiological age is also a rare event. We specifically target those interesting cases, where the colony is deprived of younger adult bees, and, as we did not model a life cycle (no birth, no death, no egg laid), we certainly won't see age polyethism, and thus, bees "classical" life of starting nurse and dying foraging [25].

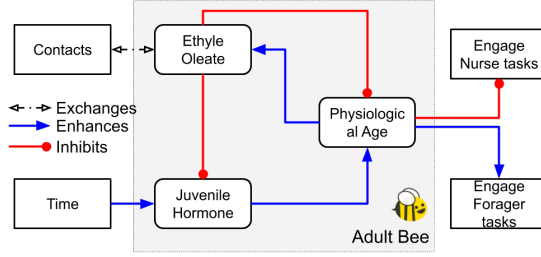


Fig. 4. Physiological dynamics of our simplified adult bee agents. We can see the internal bee variables in the middle and external factors on the left. On the right, we can see how physiological age affects task selection.

4.2 Environment and Simulator

The environment is a 2D hexagonal 30x30 grid. Each cell has a content (a larva, food, etc..) and may have an adult bee on it. Stimuli are managed at a cell level, where each cell holds an amount of each stimulus. Stimuli are modelled as two major parameters: propagation (their behaviour through space) and dissipation/evaporation (their behaviour through time). When emitting a stimulus, an agent raises the amount of that given stimulus on its cell. Then, at each time-step and for each stimulus, each cell computes its new value using its neighbours' values and its own. The parameters we used for all of our simulations are inspired by biology but calibrated to accelerate the simulations. Hormonal and pheromonal effects can take days to occur, while bees interact and wander through the hive in seconds. We decided to accelerate some effects to obtain results in reasonable simulation time (ranges from x4000 to x6000). For example, larvae are fed about every 2 hours [27], but we accelerated that to 2 seconds. This acceleration brings up a bias, as long term changes are now in the same time scale as quick changes, but won't change the core aspect of what we are demonstrating.

4.3 Experiments and expected results

Our first concern is to be sure that the final equilibrium is not influenced by the initial conditions. Moreover, equilibrium should be altered by changing the bee per larva ratio. By putting more larva per worker, we expect to have more worker dedicated to larvae feeding and vice versa. Then we will assess how the interruption mechanism alters the system. We want to assess two hypotheses: **H1:** Our model with bee physiology and pheromones can achieve task allocation and

H2: Our interruption mechanism makes the system perform better, enhancing the task allocation. Here, performing better means maximising the time spent foraging AND the larvae survival rate. We created 5 scenarios (S1 to S5) with different initial conditions that we use to assess those hypotheses. Each scenario is then simulated by bypassing the pheromonal effects (the agents’ physiological age is frozen) to assess **H1** or/and bypassing our interruption mechanism (all the tasks are considered as non-motivated) to assess **H2**.

- **S1:** Random age distribution with 150 adult bees and 150 larvae.
- **S2:** 150 adults and 150 larvae but all adult bees start as newborn.
- **S3:** 150 adults and 150 larvae but all adult bees start as old.
- **S4:** Random age distribution with 150 adults and 50 larvae.
- **S5:** Random age distribution with 150 adults and 300 larvae.

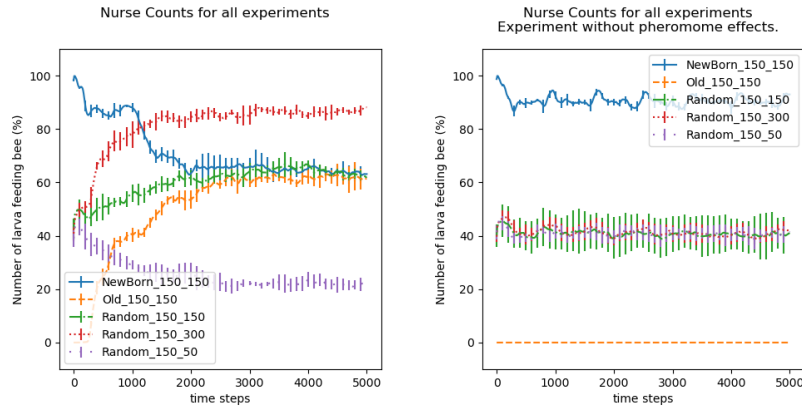


Fig. 5. Here are the different ratios of larva feeding bees of all the experiments (scenarios) we discussed, with and without the Physiology bypass. Each scenario is referred as “InitialDistribution_numberOfBees_NumberOfLarvae”.

4.4 Results and Statistical Analysis

With no bypass, S1, S2 and S3 ended with roughly 60% of the colony feeding larvae and about 20% foraging. S4 and S5, when compared to S1, changed the equilibrium. As seen before, the equilibrium for a ratio of adult bee per larva of 1/1 (150 adult bees for 150 larvae, S1) gave us equilibrium at around 60% feeding larvae. A ratio of 1/2 (S5) raises it to almost 90%. A ratio of (S4) 3/1 lowers it to 20%. Figure 5 shows side by side two graphs comparing the different ratios of feeding larva bees amongst the scenarios. We can see on the leftmost graph that S1, S2 and S3 are converging to the same equilibrium, and S4 and S5 have respectively lower and higher equilibrium than the first three. In the rightmost graph, where bee’s physiology has been bypassed, we see that the system does not adapt to the scenarios. An ANOVA analysis (Physiology x Interruption, N=50) showed that in some scenarios our interruption mechanism

makes the system achieves smaller scores, whereas in others it is the opposite: Table 2 shows an overview of those statistically relevant changes, and the reader will find on Figure 6 a graphical comparison of the interruption bypass. On the one hand, S1, S2 and S5 are slightly negatively impacted by the interruption mechanism (10% less effective foraging) where the brood care is as effective. On the other hand, in S3 and S4, the interruption mechanism improved the scores by a significant amount.

Table 2. ANOVA analysis, one row for each scenario. The left part holds the impact on the simulation scores of turning on the pheromonal effects, with our interruption mechanism bypassed (computed by scores PhysiologyOn/InterruptionOff - scores PhysiologyOff/InterruptionOff). The right part represents the impact of turning on our interruption mechanism while leaving the pheromonal effects on. Each impact is measured with the nursing score (how much larvae survived) and the foraging score (how many time-steps have been spent foraging).

	Without interruption		With interruption	
	Nursing Score	Foraging Score	Nursing Score	Foraging Score
S1	+3% ($p < 0.001$)	-30% ($p < 0.001$)	Not significant	-11% ($p < 0.001$)
S2	Not significant	0 to 150k ($p < 0.001$)	Not significant	-8% ($p < 0.001$)
S3	0% to 68% ($p < 0.001$)	Not significant	+37% ($p < 0.001$)	Not significant
S4	-32% ($p < 0.001$)	-29% ($p < 0.001$)	+45% ($p < 0.001$)	+35% ($p < 0.001$)
S5	+10% ($p < 0.001$)	-77% ($p < 0.001$)	Not significant	-14% ($p = 0.007$)

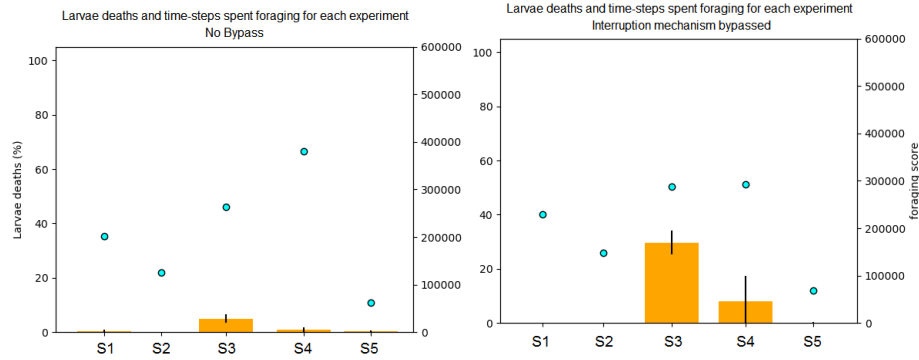


Fig. 6. These graphs show the mean deaths and time-steps spent foraging for each experiment repeated 5 times. The bars show the larvae death rate. The dots shows the number of time steps spent foraging by the colony. The left and right graphs represent the experiments respectively with and without interruption mechanism.

4.5 Discussions

We demonstrated that **H1** is true by changing the starting condition and the bee per larva ratio. However, the statistical analysis we provided only allows us to

say that **H2** is true for S3 and S4, where most adaptability is required. The speed at which a demotivating task will cause a task switch is a key parameter. Further work has to be done to calibrate it, but preliminary results show that a faster (up to a point) task switch causes overall better scores, at the cost of individual stability. We would also like to point out that the systematic re-evaluation of all tasks after each action completion could not be bypassed.

5 Conclusion and Perspectives

We proposed here a model based on Response Threshold Model, internal state of the agents and intrinsic motivation to perform task allocation in an environment where not all the tasks have stimuli representing their priority. Modelling the tasks with hierarchical subsumptions and discriminating them using an RTM approach allows us to drive the behaviour of our agents. Tasks contain demotivating actions that represent frustrating behaviours, such as looking for something. Executing a demotivating action lowers the agent's motivation to pursue its current task, making him more sensitive to other tasks. Whenever an agent changes his current task to a motivated task, it resets its motivation. We experimentally showed that our model can reach equilibrium, respond to changing conditions and preliminary work shows that it enhances the effectiveness of the system. Our implementation of this model for a honey bee simplified colony shows a division of labour capabilities. Equilibrium is reached and does not depend on initial conditions, but depends on the adult bee per larva ratio. We intend to enhance the current implementation by adding complexity toward an implementation as close as possible of how honey bee colonies work, further testing the model and its capabilities. We will then compare the behaviour of this implementation with real honey bee colony observations. In parallel we are also implementing the swarm robotics example discussed in section 3.4 to further assess the modality of our proposition.

References

1. Anderson, C., Franks, N.R.: Teamwork in animals, robots, and humans. *Advances in the Study of Behavior* (2003)
2. Baldassarre, G., Mirolli, M.: *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer Berlin Heidelberg (2013)
3. Beshers, S.N., Fewell, J.H.: Models of division of labor in social insects. *Annual review of entomology* (2001)
4. Bonabeau, E., Dorigo, M., Theraulaz, G.: *From Natural to Artificial Swarm Intelligence*. Oxford University Press, Inc. (1999)
5. Brooks, R.: A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation* (1986)
6. Campbell, A., Wu, A.S.: Multi-agent role allocation: issues, approaches, and multiple perspectives. *Autonomous Agents and Multi-Agent Systems* (2011)
7. Cornudella, M., Van Eecke, P., van Trijp, R.: How intrinsic motivation can speed up language emergence. In: *European Conference on Artificial Life*. The MIT Press (2015)

8. Csikszentmihalyi, M.: Finding flow: The psychology of engagement with everyday life. Basic Books (1997)
9. Dornhaus, A., Klügl, F., Puppe, F., Tautz, J.: Task selection in honeybees - experiments using multi-agent simulation. In: Proc. of GWAL (1998)
10. Drogoul, A., Ferber, J.: Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In: Castelfranchi, C., Werner, E. (eds.) Artificial Social Systems. Springer Berlin Heidelberg (1994)
11. Franks, N.R., Tofts, C.: Foraging for work: how tasks allocate workers. *Animal Behaviour* (1994)
12. He, X.J., et al.: Starving honey bee (*apis mellifera*) larvae signal pheromonally to worker bees. *Scientific Reports* (2016)
13. Heckel, F.W.P., Youngblood, G.M., Ketkar, N.S.: Representational complexity of reactive agents. In: Proc. of the IEEE Conference on Computational Intelligence and Games. IEEE (2010)
14. Le Conte, Y., Mohammadi, A., Robinson, G.E.: Primer effects of a brood pheromone on honeybee behavioural development. *Proceedings of the Royal Society of London. Series B: Biological Sciences* (2001)
15. Maes, P.: The agent network architecture (ANA). *ACM SIGART Bulletin* (1991)
16. Morley, R.E., Schelberg, C.: An analysis of a plant-specific dynamic scheduler. In: Proc. of the NSF Workshop on Dynamic Scheduling (1993)
17. Rao, A.S., Georgeff, M.P.: BDI agents: From theory to practice. In: International Conference on Multiagent Systems (1995)
18. Roohi, S., Takatalo, J., Guckelsberger, C., Hämäläinen, P.: Review of intrinsic motivation in simulation-based game testing. In: Proc. of the CHI Conference on Human Factors in Computing Systems - CHI '18. ACM Press (2018)
19. Ryan, R.M., Deci, E.L.: Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology* (2000)
20. Schmickl, T., Crailsheim, K.: TaskSelSim: a model of the self-organization of the division of labour in honeybees. *Mathematical and Computer Modelling of Dynamical Systems* (2008)
21. Schmickl, T., Crailsheim, K.: Analysing honeybees' division of labour in broodcare by a multi-agent model (2008)
22. Schmickl, T., Karsai, I.: Integral feedback control is at the core of task allocation and resilience of insect societies. *Proc. of the National Academy of Sciences* (2018)
23. Schmidhuber, J.: Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* (2010)
24. Seeley, T.D.: The wisdom of the hive: the social physiology of honey bee colonies. Harvard University Press (1995)
25. Seeley, T.D., Kolmes, S.A.: Age polyethism for hive duties in honey bees - illusion or reality? *Ethology* (1991)
26. Theraulaz, G., Bonabeau, E., Deneubourg, J.L.: Response threshold reinforcements and division of labour in insect societies. *Proceedings of the Royal Society B: Biological Sciences* (1998)
27. Winston, M.L.: The Biology of the Honey Bee. Harvard University Press (1991)
28. Zahadat, P., Schmickl, T.: Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition. *Adaptive Behavior* (2016)