Task Allocation in Foraging Robot Swarms: The Role of Information Sharing

Lenka Pitonakova[1,3], Richard Crowder[1] and Seth Bullock[2]

[1]Institute for Complex Systems Simulation and Department of Electronics and Computer Science, University of Southampton, United Kingdom
[2]Department of Computer Science, University of Bristol, Bristol, United Kingdom
[3]contact@lenkaspace.net

Abstract

Autonomous task allocation is a desirable feature of robot swarms that collect and deliver items in scenarios where congestion, caused by accumulated items or robots, can temporarily interfere with swarm behaviour. In such settings, self-regulation of workforce can prevent unnecessary energy consumption. We explore two types of self-regulation: *non-social*, where robots become idle upon experiencing congestion, and *social*, where robots broadcast information about congestion to their team mates in order to socially inhibit foraging. We show that while both types of self-regulation can lead to improved energy efficiency and increase the amount of resource collected, the speed with which information about congestion flows through a swarm affects the scalability of these algorithms.

Introduction

Congestion is an important factor that can negatively affect the performance of robot swarms (Hoff et al., 2010). Today's robotic systems, utilised in automated warehouses (D'Andrea, 2012), agriculture (Cartade et al., 2012), or in hospitals (Thiel et al., 2009), must maintain effective work schedules for individual robots in order to minimise interference between robots and save energy. Decentralised task allocation, which affords redundancy and scalability, has been proposed as a possible solution (Wawerla and Vaughan, 2010; D'Andrea, 2012) that is likely to become more important in the near future as autonomous robot swarms will increasingly be deployed in unstructured and dynamic environments. In this paper, we explore the problem faced by a robot swarm that collects items from the environment and can cope with congestion by regulating its workforce in a decentralised manner in order to save energy. Furthermore, we explore how the means by which information about congestion is obtained by the robots affects the scalability of the swarm's performance under various foraging conditions.

During foraging, congestion can either result from the size of the robot population or from the structure of the environment. For example, robots might be required to wait until an occupied resource drop-off location becomes accessible (Wawerla and Vaughan, 2010) or they might need to queue in order to leave a crowded drop off location to perform more work (Krieger and Billeter, 2000). An autonomous robot swarm should be able to sense when congestion has become a problem and adjust its workforce accordingly.

We simulate robot swarms that collect items from the environment and drop them off in a central base, from where the items are consumed at a given rate. We explore two types of workforce self-regulation: *non-social*, where robots become idle upon directly experiencing severe congestion, and *social*, where a robot will inhibit the foraging of its team mates by signalling them to become idle when the congestion that it experiences is severe. We show that both types of self-regulation can lead to significant energy savings and thus to a greater number of items collected when the energy available to the robots is limited. More importantly, we demonstrate that the speed with which information about congestion flows through a swarm, either when robots detect congestion themselves or when they exchange information with their team mates, affects the swarm's ability to respond to it appropriately. While social self-regulation results in rapid information flow and can lead to significant performance benefits in certain scenarios, it can also lead to significantly worse performance in others. On the other hand, non-social self-regulation, where information flow is slower, leads to improved energy efficiency across a greater number of foraging conditions, making it more suitable in unknown environments, although it is outperformed by social self-regulation in some cases.

The following sections provide an overview of related work and a description of our simulation and analysis methods. We then compare, across a number of experimental scenarios, the performance of our two types of self-regulated swarms with that of control swarms that do not use self-regulation. We evaluate both the amount of energy needed to collect items and the number of items collected when robot energy is limited. We conclude with a discussion of how our results relate to our previous work on information flow in swarms (Pitonakova et al., 2016) and provide examples of real-world applications where the two types of self-regulated swarms could be used.

Related Work

Route planning systems that optimise robot traffic are often used in controlled warehouse environments with small robot teams (Vivaldini et al., 2010; Mather and Hsieh, 2012). However, such approaches require a centralised controller to guide robot behaviour and are thus unsuitable for large swarms, where computation of optimal solutions becomes infeasible (Dahl et al., 2009). Furthermore, a model of the environment and of the tasks within it, which centralised planning systems rely on, can be difficult to obtain in dynamic or unstructured environments. On the other hand, decentralised decision making, where robots change their behaviour based on limited local information obtained through their sensors, is more suitable for complex tasks of this type (Hoff et al., 2010).

Decentralised robot decision making has been well studied in a number of logistic and foraging applications. For example, unmanned vehicles that need to transport items from one location to another can adjust their work time and decide to recruit others based on the number of items in pick-up locations (Wawerla and Vaughan, 2010). In behaviour-based robotics, a combination of environmental cues, such as the presence of items or other robots nearby, can trigger or inhibit foraging behaviour, leading to self-organised division of labour between robots that are idle and those that collect resources (Jones and Mataric, 2003). Alternatively, 'bucket brigading' robots can form chains of work areas and progressively transport items between two locations (Shell and Mataric, 2006; Pini et al., 2013) and even adapt the size of the work areas based on collisions with other robots in order to improve their performance (Lein and Vaughan, 2009).

The *Response Threshold Model* (RTM) is a self-regulatory mechanism inspired by social insects (Bonabeau et al., 1997) that has been applied in a number of simulated and real-world robot experiments. According to the model, robots alternate between *foraging* and *resting* based on some internal, environmental or social cues in order to optimise their energy consumption. For example, robots can count the number of items stored in the base and only leave to forage when the number is below a specified threshold (Yang et al., 2009). Robots can also evaluate how many items they encountered during foraging and decide to rest if the environment is not rich enough (Labella et al., 2006). By counting the number of collisions with other robots (Liu et al., 2007) or by detecting drops in their own expected performance (Dahl et al., 2009), robots can decide to rest if they estimate that congestion is beyond an acceptable level. Finally, in dynamic environments, where the number of items in the environment changes over time, robots can decrease the energy cost of collecting items by only foraging when enough items are available, estimating the state of the environment in either a centralised (Liu et al., 2007) or decentralised (Dai, 2009) manner.

Our work builds on the Response Threshold Model literature, and in particular on the work of (Liu et al., 2007) and (Dai, 2009), where robots estimated the level of congestion in order to prevent unnecessary energy consumption. However, we apply the RTM in a novel scenario, where successful foragers *recruit* other robots to the worksites that they are exploiting (see also Pitonakova et al., 2014, 2016).

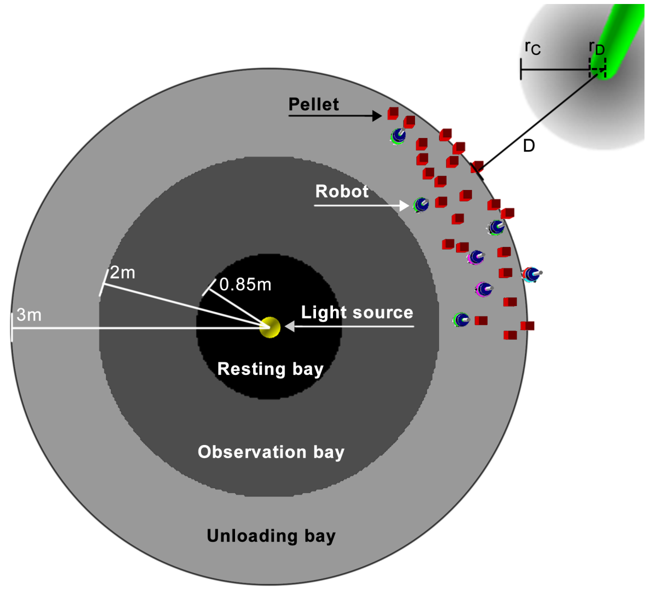

Figure 1: ARGoS simulation screenshot of the base and a deposit Dm away from the base edge. The base consists of a resting bay, an observation bay and an unloading bay. A light source is placed above the centre of the base to guide robot navigation. Pellets collected by the robots temporarily accumulate in the unloading bay, causing congestion.

Furthermore, we provide novel insights into the role played by information flow in decentralised congestion estimation. In our non-social model, congestion is estimated by each robot individually, while in the social model, robots communicate their estimates to nearby robots in order to socially inhibit foraging.

Our approach to congestion estimation is inspired by the self-regulatory behaviour of honey bees foraging for nectar (Anderson and Ratnieks, 1999; Gregson et al., 2003). When nectar is abundant, foragers may bring more nectar into the hive than the nectar-receiving bees can cope with. In order to prevent unnecessary foraging, individual foraging bees evaluate how long it takes for their nectar to be unloaded. If unloading is taking too long, a forager will *tremble dance* around the nest, inhibiting other bees from recruiting and thus reducing the number of foragers. Our social RTM uses a similar principle.

Methods

Environment

All our experiments are performed in the ARGoS simulation environment which implements realistic 3D physics and robot models (Pinciroli et al., 2012). The simulation has continuous space and updates 10 times per simulated second. A circular base with a diameter of 3 metres is situated in the centre of the experimental arena. The base is divided into three sections (Figure 1): an interior circular *resting bay* with an annular *observation bay* around it and an annular *unloading bay* around that. A light source, placed above the centre of the base, is used by robots as a reference for navigation towards and away from the base centre (as in, e.g., Krieger and Billeter, 2000; Pini et al., 2013).

Cylindrical resource deposits with radius r_D are placed outside of the base, each containing an unlimited volume of resource. In order to enable robots close to a deposit to move towards it, a colour gradient with radius r_C is present on the floor around each deposit.

We explore two types of scenarios:

- HeapN: $N \leq 4$ deposits distributed evenly around the base at a distance $D = \{5, 7, 9\}$m from the base edge. These deposits represent large heaps of resource (e.g., mineral deposits), with $r_D = 0.5$m and $r_C = 3$m.
- ScatterN: $N \geq 10$ deposits randomly distributed between $D - 5$m and $D + 5$m from the base edge. These deposits are small (e.g., litter on a street), with $r_D = 0.1$m and $r_C = 1$m.

Robots

The simulated MarXbots (Bonani et al., 2010) are circular, differentially steered robots with a diameter of 0.17m that can reach a maximum speed of 5cm/s in our simulation. The robots use infra-red sensors for obstacle avoidance and communication, colour sensors for navigation towards nearby deposits, and a light sensor for phototaxis towards the base (see Pitonakova et al., 2016, for more details). The robots are modelled as finite-state machines and can implement three types of homogeneous swarm: *control swarm*, *non-social* self-regulators, and *social* self-regulators (Figure 2).

Control swarm robots exhibit basic foraging behaviour with no self-regulation. A robot starts with a random orientation and a random position in the observation bay as an *observer*, ready to receive and follow recruitment signals. An observer moves randomly across the observation bay and avoids traveling into the unloading and resting bays. At each time step an observer can become a *scout* with scouting probability $p(S) = 10^{-3}$. A scout leaves the base and uses Lévy movement (Reynolds and Rhodes, 2009) to search for a resource deposit within 20m of the base. The robot updates its estimated location relative to the base using path

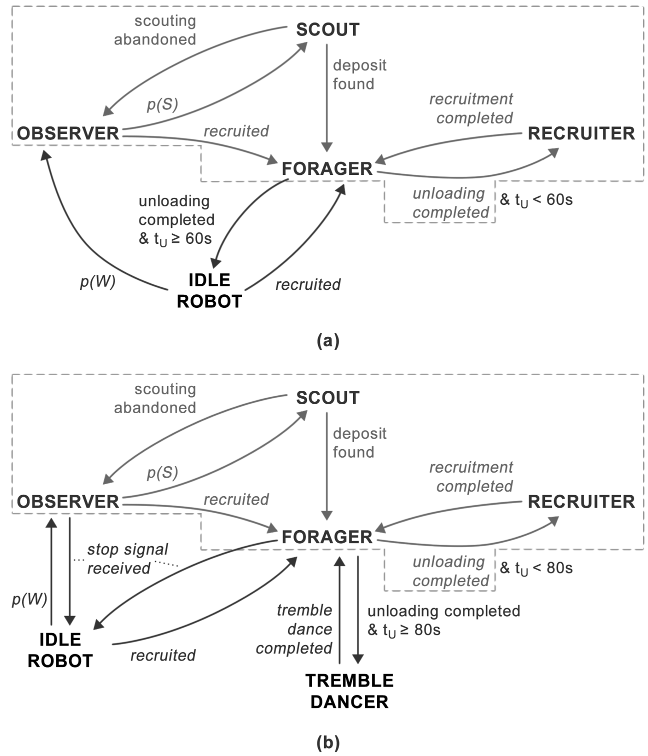

Figure 2: Finite state machine representation of the robot controller in swarms with (a) non-social self-regulation, and (b) social self-regulation. The behaviour of the control swarm controller is displayed in dashed boxes.

integration based on odometry at each time step (e.g., Lemmens et al., 2008; Gutiérrez et al., 2010). When a deposit is found, the robot loads one unit of volume of resource and returns back to the base utilising phototaxis, while keeping track of its position relative to the deposit using odometry. Odometry noise is not modelled. Any scout that cannot find a deposit within 600s returns to the base and becomes an observer.

A laden robot returning to the base drops off its load in the unloading bay in the form of four pellets of size 0.1m^3. The robots cannot push existing pellets around and thus have to avoid them in order to traverse the unloading bay. A new pellet can only be deposited when there is enough free space in front of the robot. Deposited pellets disappear from the simulation (representing their utilisation by a hypothetical unmodelled system of robots or human users) after a period of *unloading bay handling time*, t_H. When $t_H = 1$s, pellets disappear very quickly and do not cause congestion. By increasing the value of t_H, we can experiment with the level of congestion in the simulation, as more accumulated pellets make entering and leaving the base more difficult.

After depositing the pellets, the robot moves further into the base and performs recruitment for 120s, randomly moving across the base while avoiding re-entering the unloading bay. A recruiter advertises the fact that it has information about a deposit to all observers located within recruitment range of 0.6m. Deposit location is communicated to each observer in a one-to-one fashion by taking into account the local axes of the robots and their alignment relative to each other (Gutiérrez et al., 2010). The recruiter resumes foraging from the same deposit after it completes recruitment.

In *self-regulated swarms*, robots additionally measure their *pellet unloading time*, t_U, i.e., the time between entering the base and leaving the unloading bay. Robots in swarms with *non-social self-regulation* (Figure 2a) proceed to the resting bay and become *idle* after depositing pellets if congestion is severe (i.e., $t_U \geq 60$s). An idle robot consumes a negligible amount of energy (as in, e.g., Wawerla and Vaughan, 2010) and can be woken up and immediately recruited by a recruiter, i.e., by a robot that does not experience severe congestion. In order to avoid deadlocks, an idle robot can also wake up spontaneously with a waking probability $p(W) = 10^{-4}$.

Robots in swarms with *social self-regulation* (Figure 2b) do not become idle after experiencing severe congestion (i.e., when $t_U \geq 80$s), but become *tremble dancers* instead. A tremble dancer travels randomly across the observation bay and broadcasts *stop signals* to all robots within a 0.9m range for 120s, after which it leaves the base to resume foraging without recruiting. Stop signals inhibit foraging as any observer or forager that receives a stop signal moves to the resting bay and becomes idle. Stop signals also inhibit recruitment, as they cause any recruiter in range to cease recruiting and immediately leave the base to forage.

Analysis

We performed 50 simulation runs that lasted 4 simulated hours in Heap1, Heap4, Scatter10 and Scatter25 scenarios and compared the performance of all three swarm types using $N_R = 25$ and $N_R = 50$ robots. Since we are interested in efficient energy usage, we define a performance metric, *energy efficiency*, C, which represents the amount of energy a swarm spends in order to collect a unit of resource:

$$C = \frac{R}{E} \tag{1}$$

where R is the total amount of resource collected by the swarm and E is the total amount of energy expended by the swarm. It is assumed that an idle robot expends 0 units of energy per second and a robot in any other state expends 1 unit of energy per second. Since the control swarm robots are never idle, control swarms spend a total of $N_R \times (4 \times 60 \times 60) = N_R \times 14,400$ units of energy in each 4-hour experiment. We compare C values achieved by the two types of self-regulated swarms with that achieved by the control swarms in order to find out how advantageous self-regulation was in different scenarios.

We also analyse how much resource the swarms collected when energy availability was limited. During this analysis, it is assumed that all robots stop working when the swarm spends $N_R \times E'$ units of energy, where E' is the energy limit per robot. Energy limits may play a role for example in planet exploration, where robots might use a common solar-powered energy repository of a limited capacity.

Simulation Results

In the following sections, we compare the control swarms to each of the two kinds of the self-regulated swarms in terms of their energy efficiency, C, and the amount of resource they collected, R. We show that the self-regulated swarms can achieve better C in scenarios where pellets cause significant congestion. Furthermore, self-regulation leads to a higher amount of resource collected when the total energy available to the swarms is limited in such scenarios. We also discuss cases when self-regulation leads to performance deterioration, especially when social self-regulation is used.

Energy efficiency

In this section we report the performance (in terms of energy efficiency) of different swarm types in each of 48 scenarios: 2 swarm sizes (25 and 50) $\times$ 2 unloading bay handling times (5s and 20s) $\times$ 3 deposit distances (5m, 7m , and 9m) $\times$ 4 deposit distribution types (Heap1, Heap4, Scatter10 and Scatter25). In each case, we report the average performance of 50 self-regulated swarms relative to the average performance of 50 control swarms in the same scenario.

First we will summarise the performance of the control swarms, depicted in Figure 3. Their resource collection performance was more attenuated by congestion when the number of robots was large ($N_R = 50$) and when unloaded pellets did not disappear quickly from the unloading bay ($t_H = 20$s). Congestion was especially problematic in scenarios with a large number of deposits and when deposits were closer to the base. More severe performance deteriora-

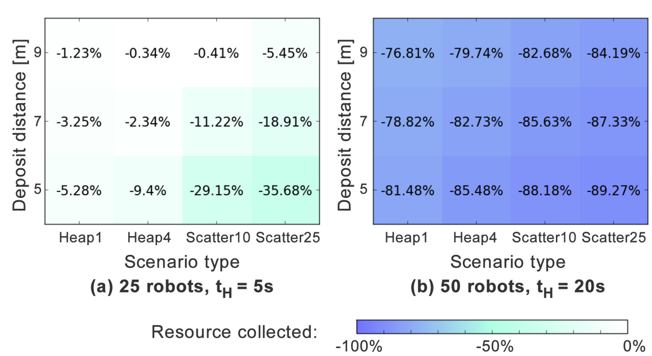

Figure 3: Resource collection performance of control swarms relative to experiments with no congestion (i.e., when $t_H = 1$s).

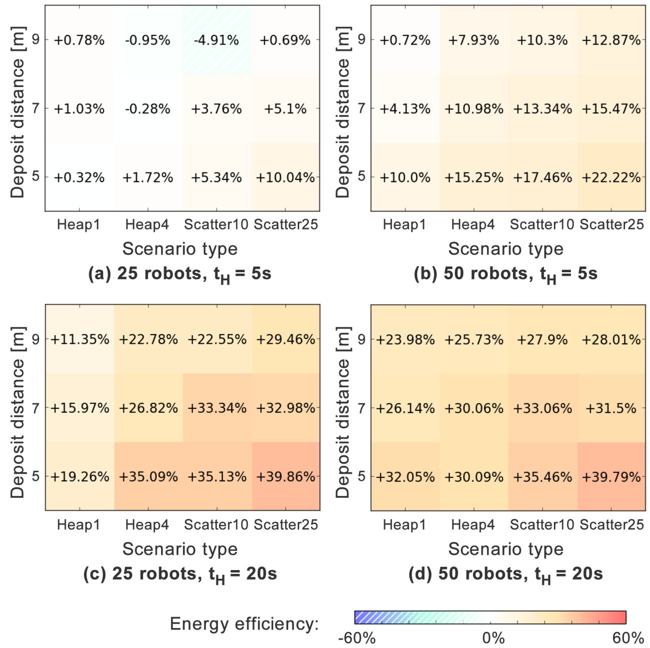

Figure 4: Performance of non-social self-regulated swarms relative to control swarms.

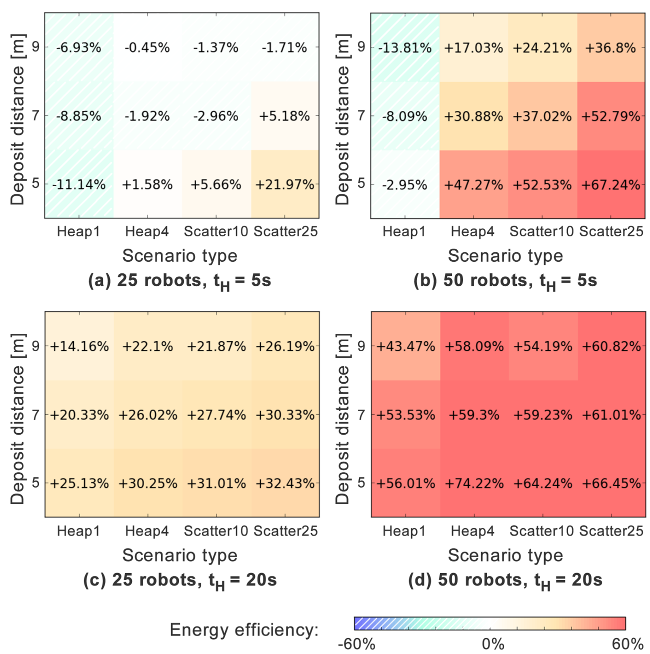

Figure 5: Performance of social self-regulated swarms relative to control swarms.

tion was measured in the Scatter scenarios, where multiple foraging locations were exploited at the same time, causing fast pellet accumulation around the whole unloading bay.

Evaluating the performance of *non-social* self-regulated swarms relative to that of control swarms (Figure 4) indicates that non-social self-regulators tended to enjoy more of an advantage when control swarms were more affected by congestion. Consequently, where congestion was very mild (e.g., a small number of robots foraging for a few heaped deposits distributed far from a base that handles unloaded deposits quickly), the performance of non-social swarms and control swarms was very similar ($\approx \pm 1\%$ difference), and control swarms even enjoyed a 5% advantage in the mildest Scatter10 scenario. However, where congestion tended to be severe (e.g., a large number of robots foraging for many scattered deposits distributed near to a base that handles unloaded deposits slowly), the performance of non-social swarms was considerably greater than that of control swarms (up to $\approx +40\%$ in the most extreme Scatter25 environments).

The performance of *social* self-regulatory swarms relative to the control swarms follows a similar but more complicated pattern (Figure 5). Again, where congestion tended to be severe, the performance of social swarms was better than that of control swarms (up to $\approx +67\%$ in the most extreme Scatter25 environments). Moreover, in these scenarios, social swarms did even better than non-social swarms, achieving an advantage over the control swarms that was often between 20% and 40% larger than that achieved by non-social swarms. Conversely, in scenarios where congestion was very mild, the performance of social swarms was *worse* than that of control swarms and non-social swarms by as much as -14%.

In general, there were two factors that affected the advantage of self-regulation: the amount of congestion in the base and the distribution of deposits in the environment. For instance, self-regulation was most advantageous in Scatter scenarios when deposits were close to the base (i.e., when the control swarms experienced high congestion due to short trips between the base and the deposits), and, more importantly, when foraging effort could be refocussed in a new direction once a particular part of the unloading bay became congested. On the other hand, self-regulation was not as effective in the Heap1 scenarios, where all resources were concentrated in a single location. Robots in the self-regulated swarms could still become idle when pellets accumulated, but recruitment could only take place again when the foraging robots measured a low unloading time, i.e., when enough of the pellets that had been unloaded in the part of the unloading bay between the deposit heap and the resting bay had disappeared. This was a particular problem for the swarms with social self-regulation, where the information about congestion spread quickly through the swarm, causing a majority of the robots to become idle. Unlike in non-social swarms, the number of foraging robots was often very low and it took the social swarms a long time to recover from inactivity.

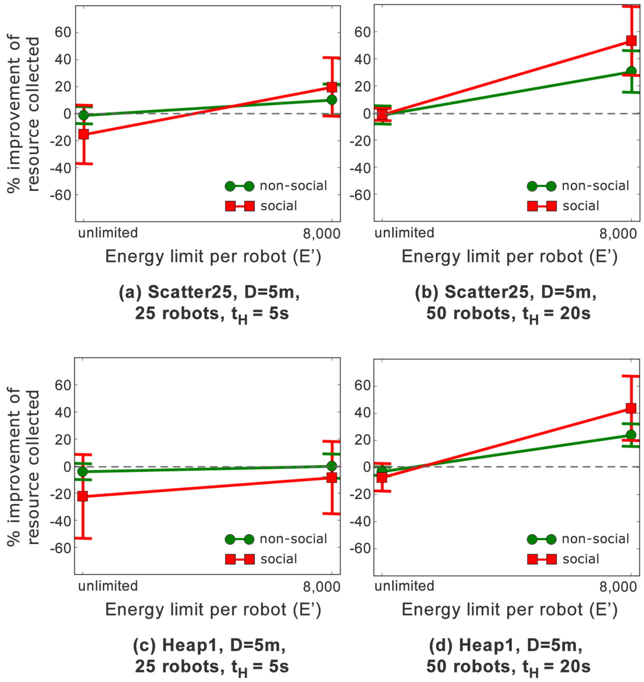

Figure 6: Resource collection performance of self-regulated swarms relative to control swarms under unlimited and limited energy conditions for scenarios with (a, c) mild congestion, and (b, d) severe congestion.

Resource collection

In this section we report the average performance (now in terms of total amount of resource collected) of the self-regulated swarms relative to the average performance of the control swarms. As in the previous section, we consider cases with mild congestion ($N_R = 25, t_H = 5s$) and severe congestion ($N_R = 50, t_H = 20s$). We first consider experiments where the total energy available to the swarms was unlimited. We then report on experiments with energy limitation, where robots ceased foraging as soon as their swarm had consumed $N_R \times E'$ units of energy, where E' was the energy limit per robot. Figure 6 depicts our results.

Although both types of self-regulated swarms were often more energy efficient than control swarms, when energy was unlimited they did not tend to collect more resource than the control swarms. Non-social self-regulators tended to collect a similar quantity to control swarms, whereas social self-regulators collected less resource than control swarms when congestion was mild (Figure 6a,c)

When swarm energy was limited, non-social self-regulators tended to either collect significantly more resource than control swarms (when congestion was severe), or roughly the same amount as control swarms (when congestions was mild). For instance, when E' was set to 8000 and when a large number of robots foraged from a base that handled unloaded deposits slowly in the Scatter25 scenario, non-social swarms foraged up to 30% more resource relative to control swarms (Figure 6b). In experiments where congestion was mild, the advantage of non-social swarms was less pronounced. For instance, when a small number of robots foraged from a base that handled unloaded deposits quickly, non-social self-regulated swarms only collected up to 10% more resource than control swarms (Figure 6a).

Social self-regulation again led to more extreme variation in performance when the swarm energy was limited. When congestion was severe, social self-regulators tended to collect significantly more resource than either control swarms or non-social self-regulators (Figure 6b and 6d). Whereas when congestion was mild, they collected roughly the same amount as control swarms and social self-regulators (Figure 6a and 6c). For instance, in Scatter25, social-self-regulators collected approximately 55% more resource than the control swarms when $E' = 8000$ (Figure 6b). On the other hand, in Heap1, where the robots could not spread their foraging effort to other directions once a particular part of the unloading bay became congested, social self-regulators collected on average 10% *less* resource than the control swarms when congestion was mild (Figure 6c). In both cases, variation in performance within a scenario was higher for social swarms.

When the value of E' was higher or lower than 8000, the relative performance of both self-regulated swarms decreased linearly but was never lower than when the swarm energy was unlimited.

Discussion and Conclusions

We have shown that swarms can regulate their foraging activity effectively on the basis of locally perceived levels of congestion. The solution presented in this paper extends previous studies of the Response Threshold Model (RTM) (e.g., Liu et al., 2007; Dahl et al., 2009; Yang et al., 2009), applying it for the first time to foraging swarms that use recruitment and investigating the effect of information sharing during decentralised congestion estimation.

We compared three types of swarms: control swarms with no self-regulation, swarms with non-social self-regulation (where robots become idle when they directly sense severe congestion), and swarms with social self-regulation (where robots instruct their team mates to become idle when they detect severe congestion). The swarms were assessed across a number of experimental scenarios, where we varied the number of deposits (N_D), deposit distance from the base (D), the number of robots (N_R), and the time it took for accumulated material to be consumed at the base (t_H). We evaluated the performance of the swarms in terms of energy efficiency, C, and showed that C can be improved through self-regulation especially in environments where the collected material accumulates in the base quickly (because D is small, or N_R or t_H are large) or where the swarms can exploit multiple foraging directions simultaneously (i.e., because N_D is large). Additionally, we demonstrated that self-regulated swarms collect more resources than control swarms when the energy supply available to the robots is limited.

There were notable differences in how swarms with non-social and social self-regulation performed in the various experimental scenarios. By comparison with control swarm behaviour, non-social self-regulation led to mediocre performance improvements or equivalent levels of performance in some scenarios. On the other hand, social self-regulation achieved large improvements over control swarms in scenarios where pellets accumulated quickly, but were also outperformed by control swarms in scenarios where congestion was not as severe or where all resources were concentrated in a single location. In our work on information flow in foraging swarms that use recruitment (Pitonakova et al., 2016), we argued that *fast* information flow can lead to pathological states of a whole swarm that prevent the swarm from responding to changes in the environment. Furthermore, we demonstrated that while swarms with fast information flow tend to perform extremely well in a limited number of environments but perform poorly in others, swarms with *slow* information flow tend to perform well across a broad spectrum of scenarios. In this paper, we extend this argument to scenarios involving congestion. Information flow was slower in swarms of non-social self-regulators which relied on their own local perception alone, and it was faster in swarms of social self-regulators which communicated information about congestion to one another. As was the case in (Pitonakova et al., 2016), slow information flow led to behaviour suitable for a larger number of experimental scenarios, while fast information flow caused more extreme variation in performance meaning it was only appropriate in a restricted set of scenarios.

Consequently, robots inspired by our social self-regulated swarms could be applied effectively in appropriate well-defined foraging or logistic tasks, for example to deliver items between various locations in warehouses and hospitals, or to collect crops. In these scenarios, the relevant task parameters (swarm size, processing time of collected items, etc.) are known upfront. However, if we were to employ robot swarms in an unknown or more variable environment, e.g., work sites on different planets or underwater, we would need to take into account the fact that while fast information flow can lead to beneficially fast response times, it can also cause significantly suboptimal performance under certain conditions. In such applications, self-regulation that is more subtle and occurs in a more localised fashion would be more suitable, not because of the ability of the swarms to perform work faster or more efficiently, but because such collective behaviour is more scalable. It might also be advantageous to create an adaptive algorithm, where robots alter their own self-regulatory behaviours (for example their willingness to exchange information with others, their waking up probability, etc.), in order to achieve a level of information flow within the swarm that varies dynamically in a way that is appropriate to the swarm's current environment.

Acknowledgements: This work was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1). All source code and data supporting this study are openly available from the University of Southampton repository at http://eprints.soton.ac.uk/386728/

References

Anderson, C. and Ratnieks, F. L. W. (1999). Worker allocation in insect societies: Coordination of nectar foragers and nectar receivers in honey bee (*Apis mellifera*) colonies. *Behavioral Ecology and Sociobiology*, 46(2):73–81.

Bonabeau, E., Sobkowski, A., Theraulaz, G., and Deneubourg, J.-L. (1997). Adaptive task allocation inspired by a model of division of labor in social insects. In Lundh, D., Olsson, B., and Narayanan, A., editors, *Biocomputing and Emergent Computation: Proceedings of BCEC97*, pages 36–45, London. World Scientific Publishing.

Bonani, M., Longchamp, V., Magnenat S., Philippe, R., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., and Mondada, F. (2010). The MarXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 4187 – 4193, Piscataway, NJ. IEEE Press.

Cartade, P., Lenain, R., Thuilot, B., Benet, B., and Berducat, M. (2012). Motion control of a heterogeneous fleet of mobile robots: Formation control for achieving agriculture task. In *Proceedings of the International Conference on Agricultural Engineering (CIGR-AgEng '12)*.

Dahl, R. S., Mataric, M. J., and Sukhatme, G. S. (2009). Multi-robot task allocation through vacancy chain scheduling. *Robotics and Autonomous Systems*, 57:674–687.

Dai, H. (2009). Adaptive control in swarm robotic systems. *The Hilltop Review*, 3(1):54–67.

D'Andrea, R. (2012). Guest editorial: A revolution in the warehouse: A retrospective on Kiva Systems and the grand challenges ahead. *IEEE Transactions on Automation Science and Engineering*, 9(4):638–639.

Gregson, A. M., Hart, A. G., Holcombe, M., and Ratnieks, F. L. (2003). Partial nectar loads as a cause of multiple nectar transfer in the honey bee (*Apis mellifera*): A simulation model. *Journal of Theoretical Biology*, 222(1):1–8.

Gutiérrez, Á., Campo, A., Monasterio-Huelin, F., Magdalena, L., and Dorigo, M. (2010). Collective decision-making based on social odometry. *Neural Computing and Applications*, 19(6):807–823.

Hoff, N., Sagoff, A., Wood, R. J., and Nagpal, R. (2010). Two foraging algorithms for robot swarms using only local communication. In *IEEE International Conference on Robotics and Biomimetics (ROBIO 2010)*, pages 123–130, Piscataway, NJ. IEEE Press.

Jones, C. and Mataric, M. J. (2003). Adaptive division of labor in large-scale minimalist multi-robot systems. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, pages 1969 – 1974, vol. 2, Piscataway, NJ. IEEE Press.

Krieger, M. J. B. and Billeter, J.-B. (2000). The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1-2):65–84.

Labella, T. H., Dorigo, M., and Deneubourg, J.-L. (2006). Division of labour in a group of robots inspired by ants' foraging behaviour. *ACM Transactions on Autonomous and Adaptive Systems*, 1(1):4–25.

Lein, A. and Vaughan, R. T. (2009). Adapting to non-uniform resource distributions in robotic swarm foraging through work-site relocation. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*, pages 601–606, Piscataway, NJ. IEEE Press.

Lemmens, N., de Jong, S., Tuyls, K., and Nowe, A. (2008). Bee behaviour in multi-agent systems. In Tuyls, K., Nowe, A., Guessoum, Z., and Kudenko, D., editors, *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, volume 4865 of *Lecture Notes in Computer Science*, pages 145–156. Springer, Berlin.

Liu, W., Winfield, A. F., Sa, J., Chen, J., and Dou, L. (2007). Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior*, 15(3):289–305.

Mather, T. W. and Hsieh, M. A. (2012). Ensemble modeling and control for congestion management in automated warehouses. In *Proceedings of the 2012 IEEE International Conference on Automation Science and Engineering (CASE 2012)*, pages 390–395, Piscataway, NJ. IEEE Press.

Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.

Pini, G., Brutschy, A., Pinciroli, C., Dorigo, M., and Birattari, M. (2013). Autonomous task partitioning in robot foraging: An approach based on cost estimation. *Adaptive Behavior*, 21(2):118–136.

Pitonakova, L., Crowder, R., and Bullock, S. (2014). Understanding the role of recruitment in collective robot foraging. In Lipson, H., Sayama, H., Rieffel, J., Risi, S., and Doursat, R., editors, *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE 14)*, pages 264–271, Cambridge, MA. MIT Press.

Pitonakova, L., Crowder, R., and Bullock, S. (2016). Information flow principles for plasticity in foraging robot swarms. *Swarm Intelligence*, pages 1–31, DOI: 10.1007/s11721–016–0118–1.

Reynolds, A. M. and Rhodes, C. J. (2009). The Lévy flight paradigm: Random search patterns and mechanisms. *Ecology*, 90(4):877–887.

Shell, D. A. and Mataric, M. J. (2006). On foraging strategies for large-scale multi-robot systems. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, pages 2717 – 2723, Piscataway, NJ. IEEE Press.

Thiel, S., Häbe, D., and Block, M. (2009). Co-operative robot teams in a hospital environment. In *Proceedings of the 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2009)*, volume 2, pages 843–847, Piscataway, NJ. IEEE Press.

Vivaldini, K., Galdames, J. P. M., B., P. T., Sobral, R. M., Araujo, R. C., Becker, M., and Caurin, G. A. P. (2010). Automatic routing system for intelligent warehouses. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010)*, pages 93–98, Piscataway, NJ. IEEE Press.

Wawerla, J. and Vaughan, R. T. (2010). A fast and frugal method for team-task allocation in a multi-robot transportation system. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010)*, pages 1432–1437, Piscataway, NJ. IEEE Press.

Yang, Y., Zhou, C., and Tian, Y. (2009). Swarm robots task allocation based on response threshold model. In *Proceedings of the 4th International Conference on Autonomous Robots and Agents (ICARA 2009)*, pages 171–176, Piscataway, NJ. IEEE Press.