1 Non-Spatial Model Applications

This section provides a brief run-down of some well studied swarm robot applications and the important results ascertained from them. While this list is by no means a comprehensive evaluation of every single swarm robot algorithm ever developed, the featured algorithms share a development model similar to the one described in the previous section, consisting of a PFSM and the ODE (or Difference Equation, DE) system that defines their macroscopic behavior. The final model described in this section is based on our ongoing research work on multi-robot coordination using a response threshold function.

1.1 Object Clustering

Object clustering, or aggregation, is one of the early applications for robot swarms inspired from transport techniques seen in ant colonies. It was introduced in [?] as an example of collective, non-cooperative behavior. The task involves moving objects that have been scattered over a region and collecting them in a pile without explicit communication or global positioning. It is an example of non-cooperative behavior since every agent is capable of moving objects on its own and thus no cooperation is required between individuals to successfully complete the task.

Agassounon, Kalra and Martinoli introduce worker allocation via threshold functions as a variant to the original object clustering experiment in [?, ?] and expand the results in [?]. Here, agents decide whether or not to help with the global task (stick collecting) based on their local perception of the degree of completion of the task. While earlier attempts at threshold based collective algorithms for robots suffer from scalability issues arising from explicit communication [?] between agents or the need for a centralized supervisor [?], the collaborative approach presented in [?] employs a wholly distributed strategy where agents use only local observations to make work vs. rest decisions.

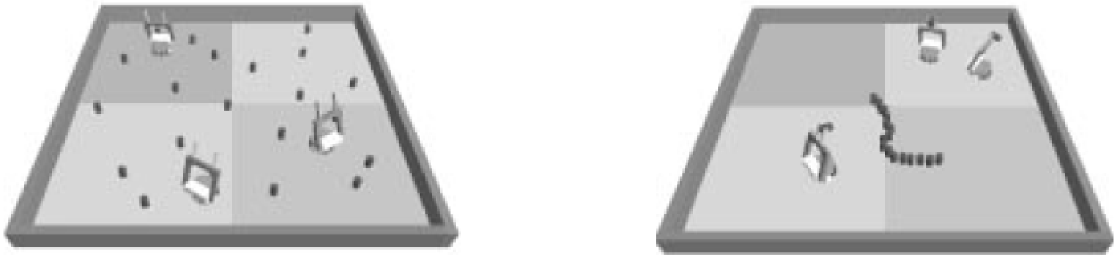


Figure 1: Webots simulation of Khepera robots performing object clustering. Left image shows the initial placement of robots/sticks in the arena while the right image shows the successful completion of the task.

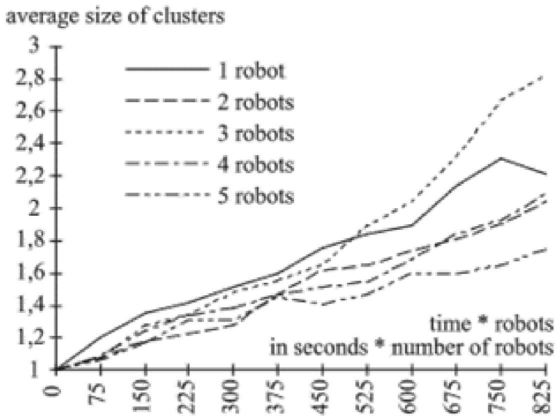


Figure 2: Results from the original object clustering task without threshold based worker allocation emphasizing the non-cooperative nature of the task, i.e. more robots do not necessarily complete the task faster.

Results

It is shown via physical experiments in [?] that adding agents to the task only speeds up the rate of collection to a certain threshold, and that overall efficiency is not necessarily maximized by adding more agents (as seen in figure 2).

For the sake of brevity, results from the threshold based worker allocation variant of this experiment have been excluded from this discussion but can be found in [?, ?, ?].

1.2 The Stick Pulling Experiment

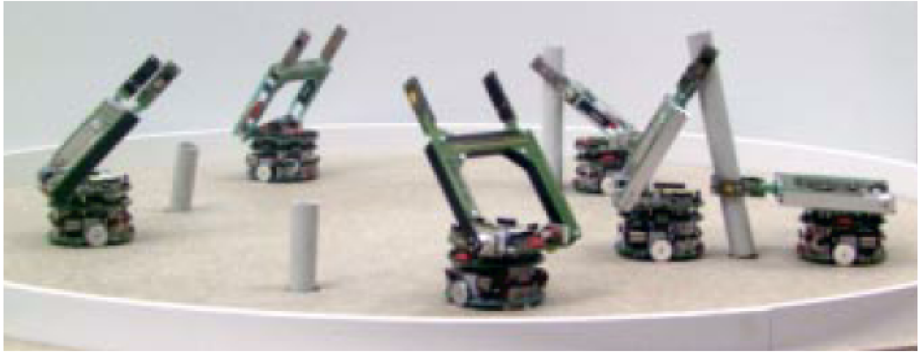


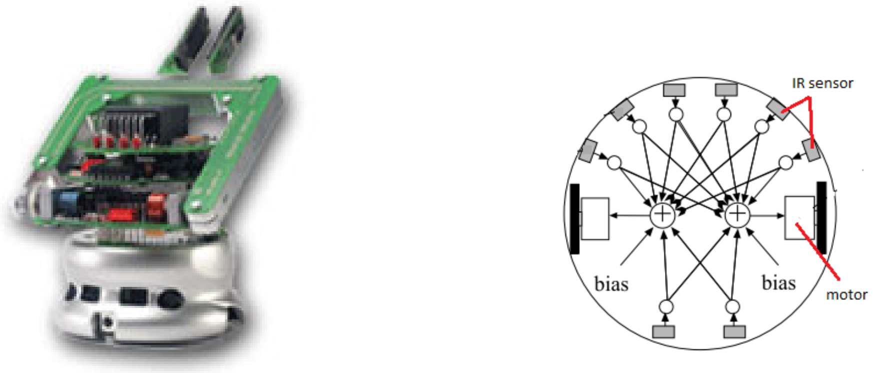
Figure 3

Introduced alongside object clustering in [?]—later extended and studied in further detail [?, ?, ?, ?, ?]—the aim of this experiment is to pull long sticks out of the ground using pairs of collaborating robots. Each stick is placed deep enough into the floor that a single robot cannot successfully pull it out. Therefore, the agent's goal is to first find a stick and then either wait at that stick if another robot is not already at it, or to help the robot currently at the stick.

The robots have an internal timer that begins to tick down as they wait at a stick. This parameter is referred to as the *gripping time* or *wait time* in the experiment. If a fellow robot does not come to help before the gripping time expires then the robot lets go of the stick it is currently at and tries to find another stick. Adjusting the gripping time for each agent non-trivially affects the overall outcome the collaborative task. Once a stick is successfully pulled out of the ground by a coalition of two robots, they perform a *success dance* and then break their coalition. Each robot then returns to the search state.

Sticks that are pulled out of the ground are replaced by a human so that the total number of sticks to pull does not change during the course of the experiment. Likewise, the number of agents is constant over course of the experiment. The goal of the experiment is to maximize the rate of collaboration between robots or, in other words, maximize the number of sticks being pulled out of the ground per unit time.

The robots used for this task are called *Khepera* robots. The technical specification of the robot are given in Table 1. The only modification made to the original hardware design for this experiment is that each robot is outfitted with a single-joint gripper arm so that it can grip and lift sticks, as seen in Figure 4a.



(a) The Khepera-II robot used in the *Stick Pulling Experiment* by Martinoli et. al.

(b) Neural net that controls sensory feedback on the Khepera-II robot for the stick pulling experiment.

Figure 4

Results

A particularly interesting insight derived from the classical stick pulling experiment and studied further in [?] is the problem of wait-time parameter optimization. As mentioned earlier, in section ??, parameter discovery and optimization are recurring tasks associated with modeling swarm systems. The sick pulling experiment provides a good practical experiment where optimization of swarm systems can be studied—the important system

Diameter	Height	Mass	Velocity
55mm	30mm	80g	0.02 to 1.0ms^{-1}
Battery Life	**CPU**	**RAM**	**EEPROM**
45min	Motorola 68331 @ 16MHz	256KB	512KB
OS	**Motors**	**Sensors**	
μKOS (RTOS)	2 DC brushed servos	8 IR proximity and ambient light	

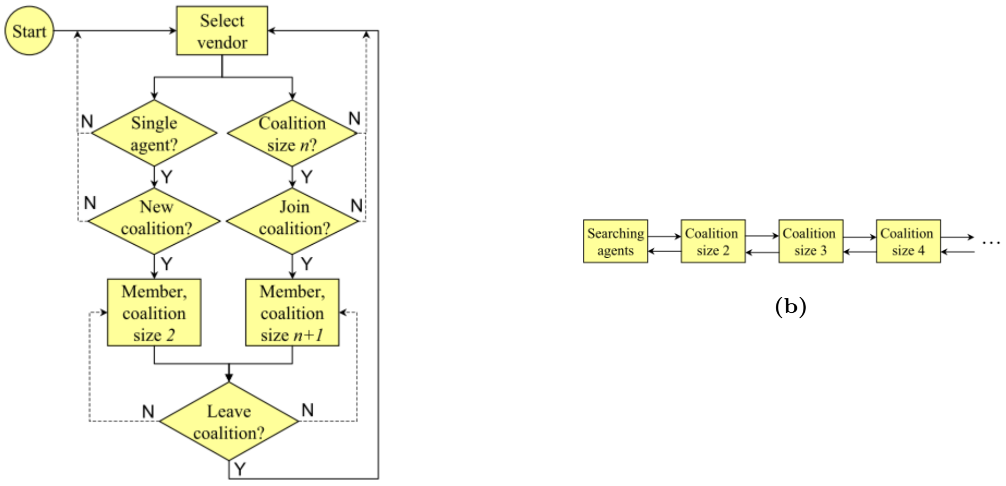
Table 1

parameter being gripping-time of the agents.

It is found that for the case where two robots are needed to pull out a stick, an optimum wait-time exists for maximizing the rate of sticks pulled when the number of robots is less than or equal to the number of sticks in the arena. Intuitively, if we have more robots than sticks, the best strategy for a robot at a stick is to just wait forever since there will be at least one robot still searching at all times. On the other hand, if there are fewer (or equal) robots than sticks, the strategy of waiting forever could cause a deadlock situation in the arena where every robot is at an individual stick waiting for a partner to arrive.

Comment 1 : Add gripping time optimization figure here.

1.3 Coalition Formation



(a)

(b)

Figure 5: The flowchart and associated FSM used to describe the robot controller for coalition formation experiments are discussed by Lerman et al.

A practical swarm optimization strategy for e-commerce applications was introduced by Lerman et al. in [**?**, **?**, **?**]. Given a set of sellers for a specific product, the goal of the agent (or buyer) is to minimize cost per unit by forming coalitions with other buyers to get group discounts.

A coalition between two or more buyers is formed if they are all purchasing the same product from the same seller. Agents are free to join or leave a coalition on their own accord but entire coalitions are not allowed to switch sellers. Discounts given to buyers are directly proportional to the size of the buying coalition. Due to storage or other constraints, sellers only keep a certain amount of the product for sale which, in turn, restricts coalition sizes. A product is kept on sale for a pre-determined period of time buy a seller after which all transactions are completed and the product is taken off the market. The flowchart and FSM for the process described above is seen in figure 5.

Results

This application has been mainly used to illustrate the use of differential equation, macro-level models for studying multi-agent systems[**?**, **?**]. The analysis provided explains how even a "greedy" agent based approach can lead to maximal utility gains for the entire system. This is a recurring property of many swarm systems—decisions are often made by an individual agent based only on local information but the effect of the decision is felt globally throughout the system.

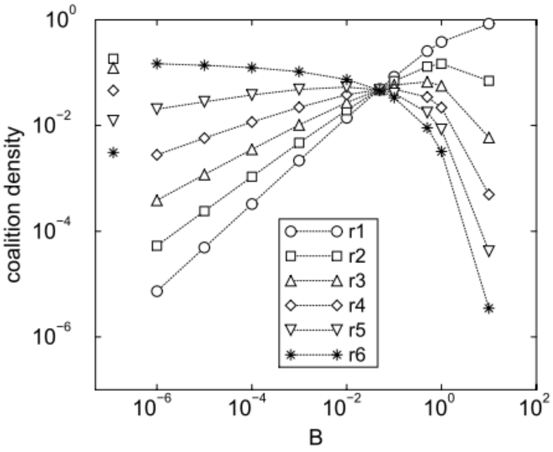


Figure 6: Plot showing utility gain in the coalition experiment with group sizes ranging from 1 to 6 per coalition and varying attachment/detachment rates

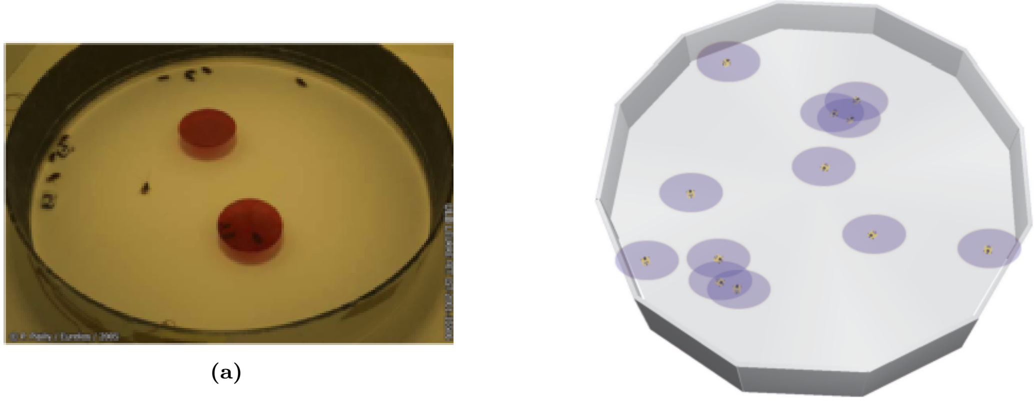
Figure 6 above shows the results obtained from solving the macro-level equations associated with this experiment for coalition sizes of 1–6 robots. Discussion of this graph in [**?**] suggests that there is only a small utility gain in the case where agents are allowed to form coalitions but not leave them. However, there is a very large utility gain in the case where agents can both, form and leave coalitions at will. The caveat is that attachment and detachment rate must be kept very low. For large detachment rates, there is again virtually no utility gain, as the system is composed mainly of unaffiliated agents.

Utility gain is a measure of profit per agent computed by,

$$G = Np - \sum_{n=1}^{m} p_n n r_n$$

where N is the total number of agents, p_n is the coalition price dependent on the coalition size ($p_n = p - \Delta p(n-1)$, $\Delta p =$ price decrement, $p =$ base price), and r_n is the number of coalitions of size n.

1.4 Cockroach Aggregation



(a)

(b)

Figure 7

In [**?**], Jeason et al. showed that a species of cockroach, *Blattella germanica*, exhibited clustering mechanics to form groups using only local information and without knowledge of the global structure of the environment or group sizes. An agent based, biological model is developed and described in the above paper where simple action rules are extrapolated from watching real cockroach aggregation behavior in a closed environment under different lighting conditions.

The data collected from real cockroach aggregation inspired the development of a robotic model for aggregation, as seen in figure 8 [**?**]. The probabilities for leaving and joining individual clusters are based exclusively on perceived cluster sizes and the time elapsed since joining a cluster. Global information such as actual cluster sizes are not available to any individual agent. Experiments using miniature *Alice* robots were conducted where the robot controller was developed using observed cockroach behavior in [**?**].

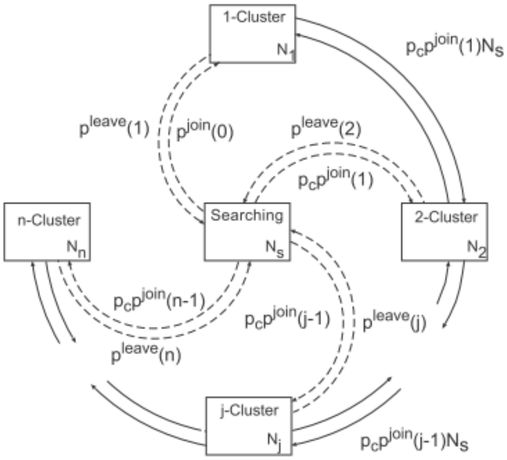


Figure 8

Results

In [**?**] the authors theorize that the size of clusters formed by robots can be controlled simply by adjusting their communication range. Specifically, "For the system with dynamics found in [. . . eqns representing the robot controller as seen in figure 8], the system behavior can change between dispersion and aggregation simply by tuning p_c [encountering probability], as long as the inequalities for $p^{join}(j)$ and $p^{leave}(j)$ [probability to join/leave a cluster, respectively] are satisfied."

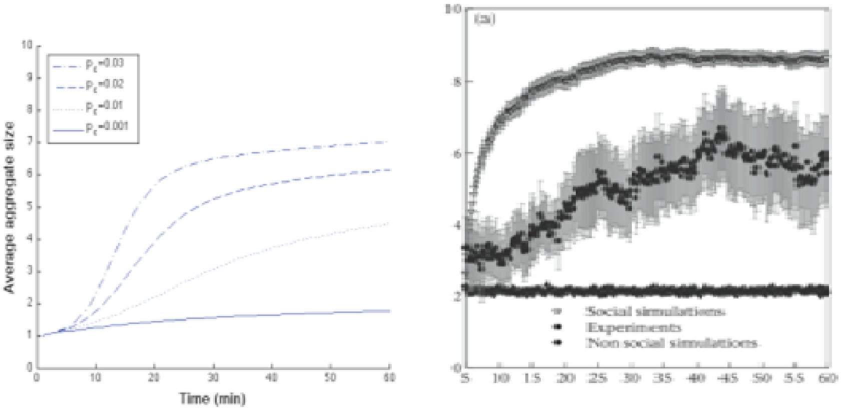


Figure 9

Figure 9 shows plots of average cluster sizes observed over time for both, the robotic experiment (left) as well as the experiment involving real cockroaches (right). The robot experiment contains a range of plots for p_c between 0.001 and 0.03. We can easily see that the cluster sizes can be made to match the real cockroach experiment just by adjusting the value of encountering probability, which is in turn a function of the communication range of the robots. Conversely, there is the interesting parameter estimation/optimization problem of finding the right p_c to get the desired global behavior (cluster size) from the swarm.

1.5 Collective Inspection Of Regular Structures

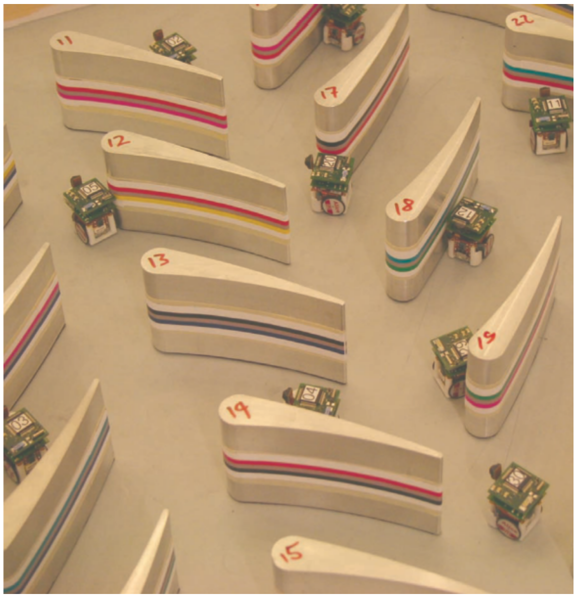


Figure 10

A practical application for swarm systems is proposed by Correll and Martinoli in[**?**, **?**]. The inspection of airplane jet turbine blades has, so far, been performed manually using borescopes. This procedure is both time and cost intensive and is subject to human error. Automating the task has proven difficult since the shielded, complex, narrow structure of a turbine imposes not only strong miniaturization constraints on the design but also prevents the use of any traditional global positioning and communication system. Furthermore, a limited on-board energy budget might prevent computation of a sophisticated deliberative planning strategy and dramatically narrows the sensor and communication range of our robots. [**?**]

These constraints on the inspection task make it well suited for a miniature robot swarm/multi-agent approach. Since jet turbine blades are regular, repeating structures they provide a good inspection target for robot vision systems.

Results

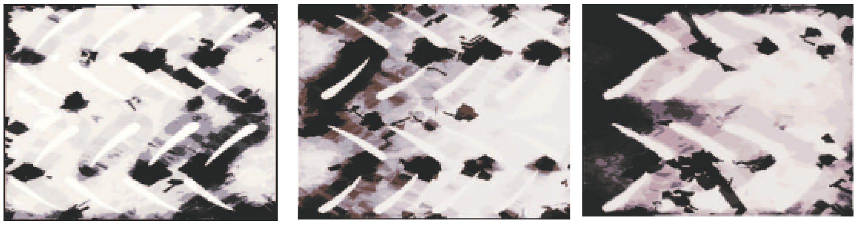


Figure 11

Simulation results seen in figure 11 show that the probability distribution (agent dispersion and coverage pattern) is non-uniform both, due to the shape of the blades, and border-following algorithm used by robots. It is evident that simply rotating the orientation of the blades results in a different coverage pattern of the arena, by the robots.

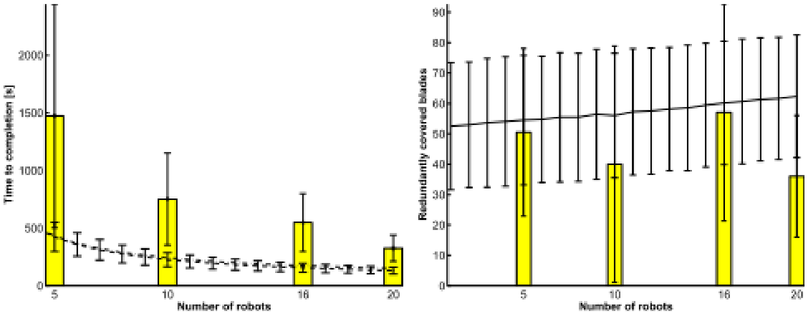


Figure 12

As a result of the inherent spatial nature of the experiment, it is found that both micro-level simulations and macroscopic equations derived for this setup (using the non-spatial methods we have discussed so far) only prove to be qualitatively correct for small robot teams (see figure 12). A more detailed analysis of the model vs. real experiment mismatch is provided by the authors in [?, ?]. We shall see a spatial modeling based approach for the same experiment in the next section.

1.6 Multi-Robot Collaboration Using A Response Threshold Function

The work by Lerman et. al [?] reinforces results obtained for stick pulling in [?, ?, ?]. It also extends the original experiment to include the case where more than two robots can be used to pull sticks of variable lengths out of the ground. We build upon this work to introduce a more general algorithm for multi-agent collaboration based on sigmoid response threshold functions, motived from task allocation in insects.

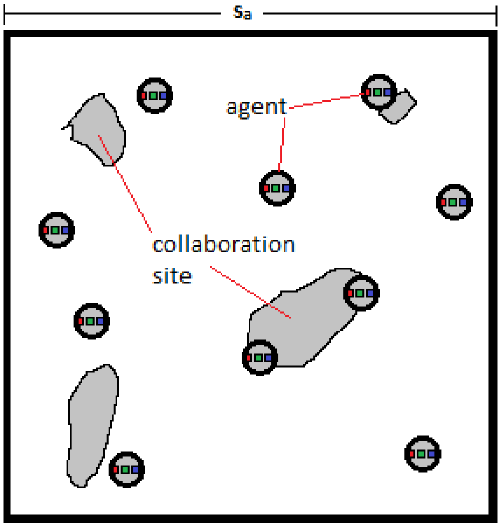


Figure 13: A visual representation of the collaboration experiment being studied. Objects in the image are not to scale.

All the collaborative tasks discussed so far are characterized by a binary outcome—success or failure to perform the group task assigned, e.g. pulling a stick out of the ground, or failing to do so. We wish to extend this model to include a more general case of collaboration tasks that involve only approximate number of robots for successful collaboration. Examples of such tasks include collective transport [?, ?], pattern recognition [?], real-time mapping of oil spills [?], determining coverage area of forest fires [?] aerial swarm surveillance, and many others that require a subset of a swarm to coalesce to tackle a task collaboratively. It is important to note that we do not study any particular task in detail but instead, outline a general approach to modeling scenarios with the aforementioned properties. In such cases, so long as we have at least one robot, the task may be accomplished with varying degrees of success. The result is now an accuracy function of the number of robots collaborating on the task, among other environmental factors.

We consider a generic collaboration task with M uniformly distributed collaboration sites within a flat arena with side, s_a. A swarm of individually simple robots is deployed within the arena, also uniformly and at random as shown in Figure 13.

The number of robots being used per experiment varies, as we discuss results for a number of different scenarios. The collaboration sites in the arena can be of various sizes and configurations—the only value that is of importance to the model is the total area of all collaboration sites within the arena.

The objective of each agent in the robot swarm is to find a collaboration site in the arena and perform a collective task with other agents at that site. Once this collaboration is complete, the robot detaches itself from its current collaboration site and repeats the search-

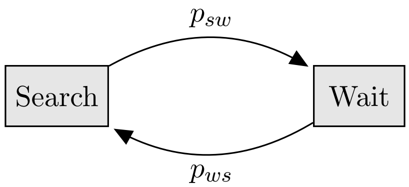


Figure 14: Robot controller used to drive group collaboration. p_{sw} is the probability of going from the search state to the wait state, while p_{ws} is the probability of going to wait to search.

wait cycle. The precise details of the collective task are not important for the purpose of understanding the coordination mechanism, and we assume collaboration to happen instantaneously for the purpose of the model studied in this paper. The robot controller that describes individual agent behavior for this swarm system is shown in Figure 14.

The population dynamics equations and the threshold function used for the model are described below. The associated PFSM is seen in figure 14

$$p_{ws}(\sigma) = \frac{1}{1 + e^{\theta(-\sigma+\tau)}} \tag{1}$$

$$\begin{align} N_s'(t) =& N_w(t)p_{ws}(\sigma) - N_s(t)p_{sw} \tag{2} \\ N_w(t) =& N_0 - N_s(t) \tag{3} \\ N_s(0) =& N_0 \tag{4} \\ N_w(0) =& 0 \tag{5} \end{align}$$

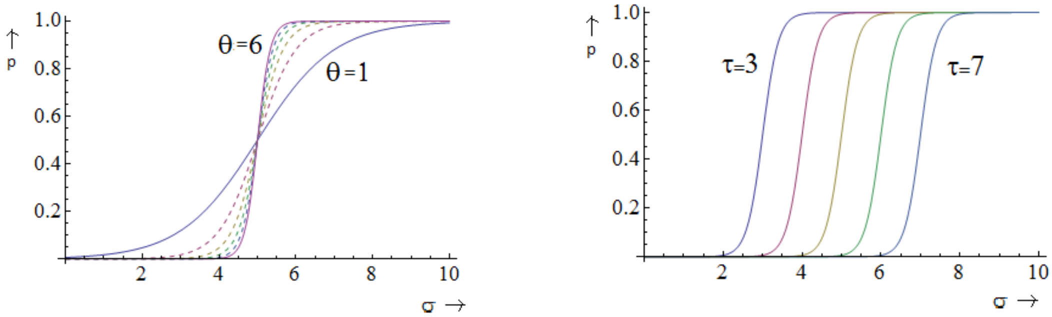
In equation (1) control parameter θ controls the slope of the sigmoid function, while τ controls its offset along the σ axis, as seen in Figure 15. The input-σ to the probability function p_{ws} within an individual robot controller is the number of agents currently at the same collaboration site as the robot. But group sizes at collaboration sites are not tracked in the macro-model and so we re-define σ as the average group size at time t, i.e. $\sigma(t) = N_w(t)/M$. Within the context of the swarming task, τ is the desired collaboration threshold value, while θ controls the variance or degree of flexibility we want the group sizes to have.

Results

We numerically solve the rate equations with the given initial condition that all robots start off in the search state((2) to (5)), for $N_s(t)$, $N_w(t)$ and $R_c(t)$, from $t = [0, 200]$. Here, $R_c(t)$ is not a state variable of the robot controller but is an aggregated value that measures the collaboration rate of the system at each time step and is mathematically described by

$$R_c(t) = N_w(t)p_{ws}(\sigma) \tag{6}$$

i.e., the number of robots going from the wait state to the search state at time t.



(a) Increasing θ in the sigmoid function increases it's slope at the point $\sigma^* = \tau$, $p_{ws}(\sigma^*) = 0.5$.

(b) Changing the τ parameter in the sigmoid function, offsets the curve along the σ axis.

Figure 15

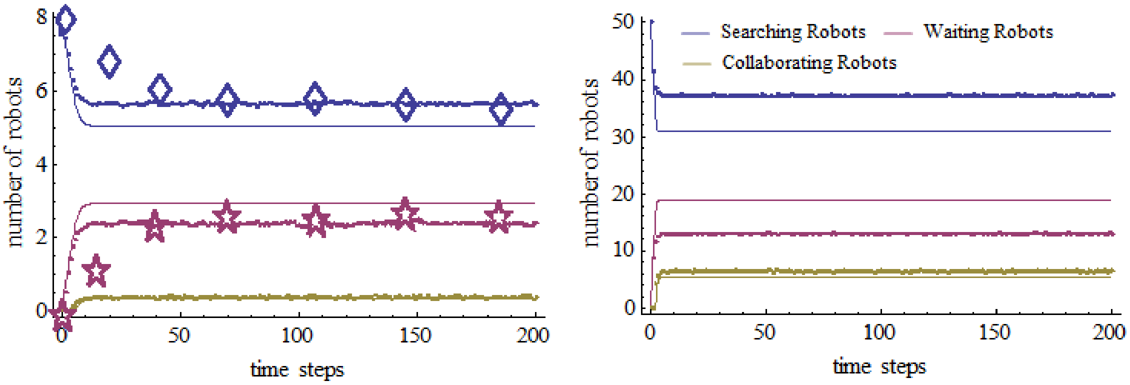
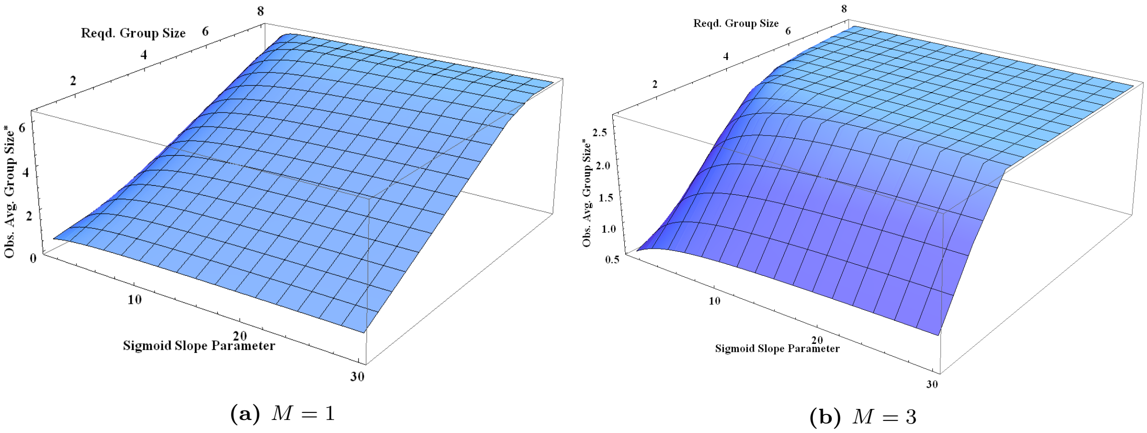


Figure 16: Macro-level numerical solutions and micro-level Gillespie simulation data plots for the proposed collaboration model. Left: $N_0 = 8$, $M = 2$, $\theta = 4$, and $\tau = 2$. The solid symbols indicate average values of 5 runs of the physics based simulation for robots in search and wait states. Right: $N_0 = 50$, $M = 5$, $\theta = 5$, and $\tau = 4$. Data plotted for 200 seconds of simulation time and 100 runs each.

Numerical solutions for the rate equation are shown in Figure 16, along with data from the microscopic simulation for the same quantities. The two plots show that the system approaches a steady state distribution of agents for each of the two states. We perform the macro-level solution and the micro-level experiments with two sets of system parameters. The left plot in Figure 16 shows results from experiments conducted using 8 robots, 2 collaboration sites, expected group sizes $\tau = 2$, and the sigmoid slope parameter $\theta = 4$. The right plot in Figure 16 shows results using $N_0 = 100$ robots, $M = 5$ collaboration sites, $\tau = 5$ expected group size, and $\theta = 4$. The data points for the microscopic model are averaged over a hundred runs per point.

Figure 17 showcases the macro-level behavior of the swarm system when we sweep across θ and τ, numerically solving the rate equations. As the slope of the sigmoid at σ^*



(a) $M = 1$

(b) $M = 3$

Figure 17: Numerical solutions for the rate equations (2) to (5) with $N_0 = 8$ and $t = [0, 200]s$ as we sweep the sigmoid control parameters, θ and τ.

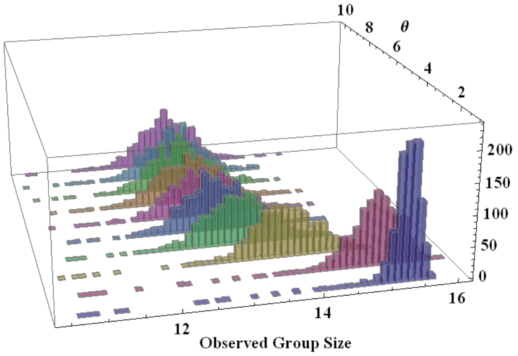


Figure 18: Relative distribution of robot group sizes as θ is varied, while keeping all other parameters constant.

gets higher, we expect to see system behavior and average group sizes resembling that as if a step function was used instead of the sigmoid probability function, i.e. for $\theta \gg 1$,

$$p_{ws}(\sigma) \approx \Theta(\sigma) = \begin{cases} 1 & : \sigma \geq \tau \\ 0 & : \text{o/w} \end{cases}$$

The sloped portion of the graphs reinforces this assertion that setting higher values for θ does indeed result in average observed group sizes being close to the requested collaboration threshold using the τ parameter. As the slope parameter, θ, is reduced, we see a marked drop in the correspondence between observed vs. requested group sizes. We also see that increasing θ past a certain point results in no major observed vs. requested group size shift.

Finally, we are interested in studying the distribution of group sizes during the course of the experiment, as the sigmoid slope parameter is varied. Figure 18 shows a 3D histogram of observed group sizes for increasing θ values, from 1 to 10, over 200 seconds of Gillespie simulation time, using a 100 robots, 5 collaboration sites, and a threshold value of 20. All values a averaged over 100 simulation runs.