Macroscopic Modeling of Aggregation Experiments using Embodied Agents in Teams of Constant and Time-Varying Sizes

William Agassounon[1a], *Alcherio Martinoli*[2a], *and Kjerstin Easton*[3a]

1: Physical Sciences Inc., 20 New England Business Center, Andover, MA 01810, U.S.A.
2: Swarm-Intelligent Systems Group, Nonlinear Systems Laboratory, EPFL, 1015 Lausanne, Switzerland
3: Robotics Laboratory, California Institute of Technology, Pasadena, CA 91125, U.S.A.
E-mail: agassounon@psicorp.com, alcherio.martinoli@epfl.ch, easton@caltech.edu

Abstract

In this paper, we present discrete-time, nonspatial, macroscopic models able to capture the dynamics of collective aggregation experiments using groups of embodied agents endowed with reactive controllers. The strength of the proposed models is that they have been built up incrementally, with matching between models and embodied simulations verified at each step as new complexity was added. Precise heuristic criteria based on geometrical considerations and systematic tests with one or two embodied agents prevent the introduction of free parameters into the models. The collective aggregation experiments presented in this paper are concerned with the gathering and clustering of small objects initially scattered in an enclosed arena. Experiments were carried out with teams consisting of one to ten individuals, using groups of both constant and time-varying sizes. In the latter case, the number of active workers was controlled by a simple, fully distributed, threshold-based algorithm whose aim was to allocate an appropriate number of individuals to a time-evolving aggregation demand. To this purpose, agents exclusively used their local perception to estimate the availability of work. Results show that models can deliver both qualitatively and quantitatively correct predictions and they represent a useful tool for generalizing the dynamics of these highly stochastic, asynchronous, nonlinear systems, often outperforming intuitive reasoning. Finally, in addition to discussions of small prediction discrepancies and difficulties in generating quantitatively correct macroscopic models, we conclude the paper by reviewing the intrinsic limitations of the current modeling methodology and by proposing a few suggestions for future work.

Keywords: macroscopic modeling, swarm intelligence, swarm robotics, threshold-based allocation algorithms, collective aggregation.

[a] This work was carried out when all three authors were part of the Collective Robotics Research Group at the California Institute of Technology (see also http://swis.epfl.ch). It was supported primarily by the TRW Foundation and in part by the Engineering Research Centers Program of the American NSF under Award Number EEC-9402726. AM is currently sponsored by a Swiss NSF professorship.

1 Introduction

Swarm Intelligence (SI)[6,8] is an innovative computational and behavioral metaphor for solving distributed problems that takes its inspiration from the biological examples provided by social insects [8,9] such as ants, termites, bees, and wasps and by swarming, flocking, herding, and shoaling phenomena in vertebrates [18,35]. The abilities of such natural systems appear to transcend the abilities of the constituent individual agents. In most biological cases studied so far, the robust and capable high-level group behavior has been found to be mediated by nothing more than a small set of simple low-level interactions between individuals, and between individuals and the environment. The SI approach emphasizes self-organization, distributedness, parallelism, and exploitation of direct (peer-to-peer) or indirect (via the environment) local communication mechanisms among relatively simple agents.

The main advantages of the application of the SI approach to the control of multiple robots are four-fold: (i) scalability: the control architecture can be kept exactly the same from a few units to thousands of units; (ii) flexibility: units can be dynamically added or removed, they can be given the ability to reallocate and redistribute themselves in a self-organized way; (iii) robustness: the resulting collective system is robust in facing a priori unknown environmental and team changes not only through unit redundancy but also through an adequate balance between exploratory and exploitative behavior; (iv) individual simplicity: a simple way to obtain an adequate explorative-exploitative balance and at the same time allow for unit miniaturization and overall system cost reduction is to minimize individual complexity.

Probabilistic modeling – The main motivation for developing a modeling methodology for swarm robotic systems is that, while SI principles are appealing from scalability, robustness, and individual simplicity points of view, they do not provide us with a way to quantitatively predict the swarm performance according to a particular metric or analyze further possible optimization margins and intrinsic limitations of this approach from an engineering point of view. In other words, if we want to achieve coordinated, self-organized group behavior based on local interactions, we need to have appropriate tools for understanding how to design and control individual units so that the swarm can achieve target behaviors and levels of performance. Models allow the engineer to capture the dynamics of these nonlinear, asynchronous, potentially large-scale systems at more abstract levels, sometimes achieving even mathematical tractability. More generally, modeling is a means for saving time, enabling generalization to different robotic platforms, and estimating optimal system parameters, including control parameters and number of agents in a team.

There are fundamentally two different categories of models: microscopic and macroscopic models. While in microscopic models each robot has a separate representation, in macroscopic models, such as those presented in this paper, a single probabilistic representation (in our case a Markov chain) summarizes the whole robotic team. Therefore, macroscopic models offer a direct description of the collective group behavior and are computationally more efficient than their microscopic counterparts since, even if they must be solved numerically, their computation time is independent of the number of agents in the system.

Although for a long period early in collective robotics research there was relatively little work in modeling of multi-robot systems, recently physicists and engineers have dedicated more attention to this problem (see, for example, related work performed by Kazadi et al. [17], Lerman and Galstyan [23], and Sugawara and Sano [36,37]). Moreover, modeling methodologies for swarm robotics systems must take into account mobility, individual intelligence, intrinsic stochastic properties of the collective coordination based on SI-principles, and, potentially, several different modalities of interaction among individuals and between an individual and the environment (e.g., mechanical, electromagnetic, chemical). This extremely rich combination of system features has drastically reduced the applicability of modeling techniques developed and commonly used in other fields.

In this paper, we combine the expertise accumulated while building probabilistic microscopic models of aggregation [25,26] with that from devising macroscopic models [1,3,22,28,29,30] for various distributed manipulation experiments including aggregation. In contrast to other classical approaches adopted in robotics, the originality of our models is that instead of being derived first in highly abstracted environmental conditions, where everything is noise-free and accurate, where global information about the world is available, and then moving down to reality by relaxing certain assumptions, we proceed along the path in reverse. Much as natural system scientists do, we start from real experimental data, gradually increasing the level of abstraction of the system description. Unlike biologists and physicists, however, we engineer our artificial systems and can therefore incrementally remove or add system details from a certain description level in order to understand their influence on the metric chosen for evaluating the performance of the swarm. Therefore, in contrast to contributions which traded quantitative prediction accuracy for model simplicity [17,22], our research aims to obtain quantitatively correct predictions without free parameters through an incremental development and corresponding validation of models using less abstracted tools (sensor-based, embodied simulator or real robots).

Macroscopic models and division of labor – Generally speaking, we are interested in understanding task allocation and labor division mechanisms exploited in social insect societies that are suitable for artificial, embedded systems such as multiple mobile robot platforms. Recently, several models have been proposed to explain these mechanisms in natural colonies [5]: some of them based on threshold responses[b] [7,38], others focus only on task-switching probabilities [34]. However, none of these publications has attempted to compare the macroscopic description of the system (which matches the experimental data) with a microscopic characterization, for instance by investigating how workers gather the information necessary to decide whether or not to switch task or to engage in a task performance. More specifically, they have not taken into consideration the partial perception in time and space of the demand and the embodiment of the agents. For instance, partial perceptions of the demand combined with real world uncertainties could strongly influence the optimal distribution of thresholds among teammates or the switching mechanism itself (e.g., probabilistic vs. deterministic). Therefore, our task as

[b] In threshold-based systems, the 'propensity' of any agent to act is governed by a response threshold. If the demand is above the agent's threshold then that agent continues to perform the task; conversely, if the demand is below its threshold, then the agent stops performing that particular task.

engineers is to use our intuition and engineering knowledge to concretely design individual robots, from software to hardware, which then allows the swarm as a whole to achieve self-organized division of labor.

Our efforts do not necessarily target the mimicking of natural, microscopic mechanisms since the available models of natural systems do not tell us implementation details at lower level and, moreover, natural and artificial hardware substrates show fundamental differences. However, we hope that this paper will not only contribute to the autonomous robotic literature but also stimulate the discussion among biologists about possible low-level mechanisms used by social insect societies for achieving division of labor.

Krieger and Billeter provide us one of the first examples of an engineering interpretation of the macroscopic allocation models presented in the social insect literature in [19]. In this experiment, the individual decision of whether or not to undertake a foraging mission was regulated by an internal threshold. A regulation of the foraging activity translated in this particular setup into a reduction of traffic jams at the nest entrance. The stimulus used for regulating the individual robot's activity was the nest energy, a software value which was assessed by an external supervisor as a function of the foraging success of the robotic colony and *globally* transmitted to all the robots. By implementing the threshold-based algorithm in this way, each individual robot must be endowed with a *different* threshold in order to achieve regulation of the overall colony activity. This in turn not only forced the experimenters to select an ad hoc distribution of thresholds but also resulted in a different exploitation of the teammates, the one endowed with the lowest threshold systematically being more active than that with the highest threshold.

In contrast with the engineering interpretation described above, in this paper we propose a threshold-based, distributed, scalable worker allocation algorithm that is based exclusively on the *local* estimation of the demand by the individuals. The individuals are all characterized by the *same* threshold but, since the agents do not perceive the demand globally but rather estimate it locally, they do not work or rest all at the same time, a behavior that would arise if the demand was broadcasted from an external supervisor.

The case study – The experiments presented in this article are the follow-up of experiments performed with real robots reported by Martinoli et al. [24,25,26]. The task is to collect small objects, referred to as "seeds", in a square arena and to gather them in a single cluster[c] using simulated Khepera robots [32] equipped with grippers and capable of distinguishing seeds from obstacles (walls, teammates) with their frontal sensors. As the robots have only local sensing capabilities and do not exploit a fully connected communication network, there is neither central nor global coordination among robots. Collective coordination is purely probabilistic and happens based on local interactions, strictly following the SI-principles mentioned above (see the experiment description in Section 2). In the experiments modeled in this paper, robots do not exploit any form of specific wireless peer-to-peer communication; stigmergic communication via the aggregation process is the only way to (indirectly) communicate. Moreover, as the robots do not have a global perception of the environment, they do not know when the task is

[c] Although these experiments are not intended to reproduce a biological system, they present several similarities with the nest cleaning and dead ants clustering performed by some ant colonies [11].

finished, motivating the need for a distributed task allocation mechanism to allow each individual to stop working based on its own estimate of the availability of work. Robots that decide to stop working can rest in a dedicated parking zone adjacent to the main arena. Allocation algorithms relying on direct peer-to-peer wireless communication have been published elsewhere [2] and are briefly mentioned in Subsection 7.2.

The specific interest for modeling this particular aggregation experiment lies in the fact that, although the control algorithms used for aggregation and individual activity regulation are both deterministic (and therefore easy to model), the whole system depends on and exploits randomness to achieve its final state, i.e., a single cluster and all the robots resting. Indeed, the stochastic nature of the robot-environment interactions combined with noisy sensory readings automatically generates probabilistic transitions between system (i.e., the robots' and environment's) states. As we will see more specifically in the next sections of the paper, first, randomness combined with irreversible elimination of isolated seeds prevents the aggregation process from getting stuck in multiple-cluster configurations, and second, randomness combined with an irreversible decision to rest achieves a self-regulation of the swarm activity which gracefully follows the evolution of the aggregation process.

Paper organization – In Section 2, we give a detailed description of the setup and of the embodied agents' control algorithms. In Section 3, we introduce the modeling methodology and we specifically address how we derived and calibrated the models presented in this paper. In Section 4, we first introduce and analyze two simple systems in which the robots search, avoid obstacles, and eventually interact with objects without, however, manipulating them. We validate model predictions with embodied simulations. In Section 5, we present the full aggregation experiment without worker allocation while in Section 6 we illustrate how the model can be extended for capturing the worker allocation mechanism. In both sections, we validate model predictions with embodied simulations. In Section 7, we discuss problems in building accurate models, strength and limitations of the modeling methodology, and briefly introduce (but do not model) possible, more efficient and/or robust variations of the worker allocation algorithm. Section 8 concludes the paper and presents a few suggestions for future work.

2 A Case Study: The Aggregation Experiment

The case study described in this paper and used for assessing the efficiency of the worker allocation algorithm is concerned with the gathering and clustering of small objects scattered around an enclosed arena. We refer to these small objects as "seeds." In most of the work done so far, for instance that performed by Holland and Melhuish [15], and more specifically that published by Martinoli et al. [25,26], the size of the working team was kept constant during the whole aggregation process. These latter experiments define the baseline for an efficiency comparison with and without the worker allocation algorithm. In this paper, we are using three primary team performance measurements: the average cluster size, the average number of clusters, and the average number of active workers in the environment.

2.1 Embodied simulations

In order to systematically investigate the aggregation dynamics and at the same time validate with a realistic simulator the prediction of our models, we implemented the experiment in Webots [31], a 3D, kinematic, sensor-based simulator of Khepera robots (see Figure 1). Teams of one to ten robots were simulated using Webots. The simulator computes trajectories and sensory input of the robots in an arena corresponding to a given physical setup. The resulting simulation is sufficiently faithful for the controllers to be transferred to real robots without changes and for the simulated robot behaviors to be very similar to those of the real robots, as shown in several previous papers [14,16,25,26]. It is worth noting that throughout this paper we will use the words "embodied agent" and "robot" for describing the same realistically simulated entity. The mean speed-up ratio for this experiment with five robots between Webots and real time is about 10 on a Pentium III, 933 MHz machine.

2.2 The setup

In the experiments presented in this paper, the arena consists of an inner *working* zone of 80 x 80 cm^2, in which the cluster formation takes place, and a surrounding *parking* zone of 40 cm in width, in which the robots that decide to stop working go and stay in an idle state to save energy. 20 seeds, 1.7 cm in diameter and 2.5 cm in height, are randomly scattered in the working zone at the beginning of the experiment. The picture presented in Figure 1 shows a 178 x 178 cm^2 working zone; this is chosen for clarity, as robots, seeds, working zone, and parking zone are more easily distinguishable in this picture than in a picture of an 80 x 80 cm^2 working zone.

Groups of one to ten embodied agents equipped with gripper turrets are used to collect and cluster the seeds. Agents can distinguish seeds from walls and other robots because of their thinness, simply using their six frontal proximity sensors. Furthermore, a floor-color sensor allows each agent to distinguish between the working and parking zones.

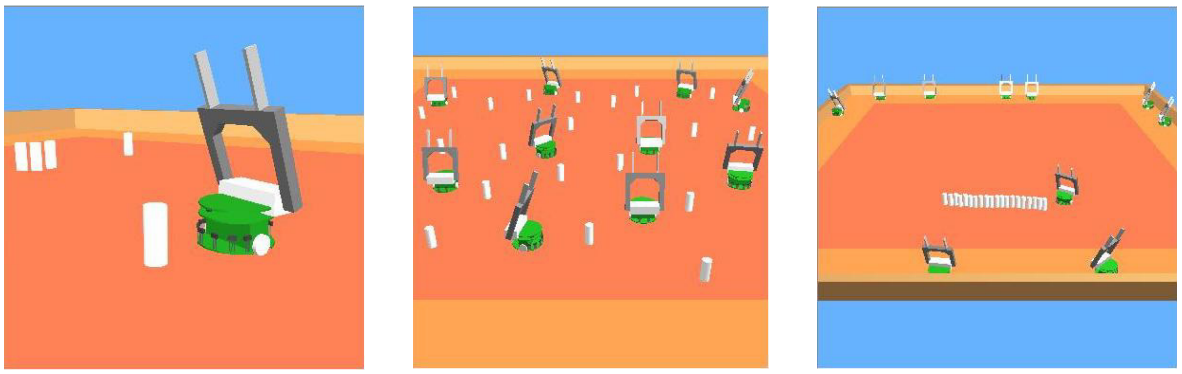

Figure 1 *Left*: close-up of a simulated robot equipped with a gripper, ready to grip a seed. *Center*: corresponding setup in the embodied simulator (10 robots, 20 seeds, 178 x 178 cm^2 arena). The inner area represents the working zone and the surrounding area is the parking/resting zone where robots that decide to stop working stay idle. *Right*: typical end of aggregation experiment, e.g., 5 hours of simulated time, in a 178 x 178 cm^2 arena.

2.3 The robot's controller

The behavior of each robot is determined by a simple hand-coded program that can be represented with a standard flow chart or a Finite State Machine (FSM), as depicted in

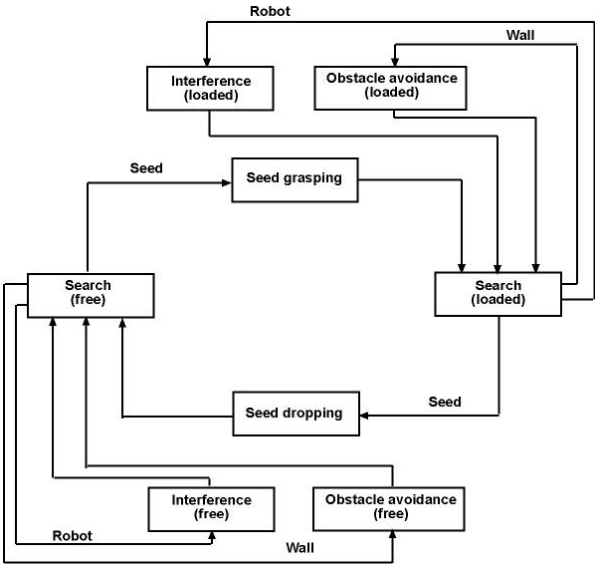

Figure 2: FSM representing the robot controller. Transitions between states are deterministically triggered by sensory measurements. This representation does not include the distributed worker allocation mechanism introduced and detailed in Section 6.

Figure 2. The behavioral granularity shown in Figure 2 is arbitrary and is chosen by the experimenter so that the FSM captures all the details of interest.

Without considering the mode-switching behavior (i.e., the worker allocation mechanism explained in detail in Section 6), we can summarize each agent's behavior with the following simple rules. In its default behavior, the agent moves straight forward within the working zone looking for seeds. When at least one of its six frontal proximity sensors is activated, the agent starts a distinguishing procedure. Basically, two cases can occur: the agent might be in front of either a large or a small object. If the agent is in front of a large object (a wall, another agent, or the long side of a cluster of seeds) the object is considered to be an obstacle and the agent avoids it with an appropriate maneuver. In the second case, the small object is considered to be a seed. If the agent is not already carrying a seed, it grasps this one with the gripper, otherwise it drops the seed it is carrying close to the one it has found; then in both cases, the agent resumes searching for seeds. With this simple individual behavior, the team is able to gather objects in clusters of increasing size. A cluster is defined as a group of seeds whose neighboring elements are separated by at most one seed diameter. We note that, since agents can identify only the two ends of a cluster as seeds (as opposed to obstacles), clusters are built in a line.

3 Macroscopic Modeling

The central idea of the probabilistic modeling methodology is to describe the experiment as a series of stochastic events with probabilities computed from the interactions' geometrical properties. The absolute location of the events in the arena is not considered in the models and the models' parameter calibration is achieved with systematic experiments using one or two embodied agents. The flowchart of the controller of the embodied agent (see Figure 2) serves as a blueprint for implementing the corresponding models, either microscopic or macroscopic. The FSM characterizing the embodied agent's controller becomes a Probabilistic Finite State Machine (PFSM) or Markov chain in the models whose state-to-state transitions depend on the interaction probabilities of a robot with other teammates and with the environment. While in microscopic models each robot is represented by its own PFSM, in macroscopic models, such as those presented in this paper, a single PFSM summarizes the whole robotic team, each of its states representing the average number of teammates in a particular state at a certain time step. In both types of models, the robots' PFSM(s) are then coupled with the environment. This coupling among robots via the environment (or in other experiments, direct peer-to-peer coupling, for example, through explicit communication) shapes the microscopic-to-macroscopic mapping, in particular, determining its linear or nonlinear properties. Moreover, in distributed manipulation experiments, the environment can be considered to be a passive, shared resource whose modifications are generated by the concurrent actions of the robots. For instance, in order to compute the arbitrary nonspatial aggregation metrics we are interested in (average cluster size, average number of clusters), we keep track of average modifications of environmental quantities in the macroscopic model.

The mean speed-up ratio between the macroscopic model (implemented in C) and embodied simulations for this experiment with five robots is approximately 5000 on a Pentium III, 933 MHz machine. This comparison is based on the time it takes to obtain a macroscopic model prediction of an experiment and to conduct a single embodied simulation, although several runs of the latter are needed to obtain statistically significant information. For instance, for each experimental result presented in this paper 30 runs were performed using Webots, corresponding to a speed-up ratio of 150,000 between the macroscopic model and the embodied simulator for predicting the average swarm performance according to one of the three aforementioned metrics.

3.1 Modeling assumptions

The current modeling methodology relies on two main, slightly overlapping assumptions: spatial uniformity and the fulfillment of Markov properties, each of them briefly described in turn.

Nonspatial models – The methodology relies on the assumption that the coverage of the arena by the groups of robots is as uniform as if the robots could hop around randomly on the surface. Therefore, robots' trajectories or specific robot spatial distributions are not considered in the current models. We also assume that the absolute position of any given object in the arena does not play a role. For instance, a seed in a given cluster has the same probability of being manipulated whether the cluster is placed in the center or in the periphery of the arena.

Semi-Markovian models - We assume that each robot's future state depends only on its present state and on how much time it has spent in that state. This assumption is correct for a reactive robot controller extended with a time-out (e.g., the threshold used in the worker allocation algorithm) or following a predetermined sequence of actions (e.g., grasping or dropping a seed) that last a certain amount of time. The robots (and the environment) in the aggregation case study clearly obey this Markov property if we assume that we are considering all robot and environment states of interest for computing the desired nonspatial metric (e.g., trajectory states - position and heading - can be neglected).

Both assumptions are valid in all the experiments presented in this paper. In Section 7, we discuss in detail the limitations of such assumptions under special experimental conditions.

3.2 Time discretization

The current probabilistic methodology generates discrete-time models. In this subsection we will explain the motivation behind this choice. At a first glance, one might think that, since the physical setup operates in continuous, real time, and robots are working in a completely asynchronous way, each of them following its own internal digital clock, the best way to obtain a faithful model would also be a continuous-time model. As usual, we would emulate time continuity in simulation by choosing a small time step combined with a standard numerical integration algorithm (e.g., Runge-Kutta). However, if we look more closely, we realize that the description of the system we are interested in (see Figure 2) is at a behavioral level, that the time accuracy used for measuring characteristic system delays (see Subsection 3.3.1) and representing the metrics of interest is not infinite (in this paper, we use 1 s), and that the fastest environmental modification which can happen (picking up or dropping a seed) requires at least 10 s. Therefore, a discrete-time description would capture accurately these characteristic time constants and delays without extra computation in between.

These are the main reasons why we believe discrete-time models are the most adequate solution for the level of description we are aiming at. Further emulation of time continuity would simply add simulation time (because macroscopic models are, in general, nonlinear, they must be solved numerically) without bringing any additional accuracy to the models' predictions.

3.3 Calibration of models' parameters

The models presented in this paper are characterized by two different categories of parameters: delays and transition probabilities. In the following two subsections, we will describe how we calculated and measured all the parameters belonging to either one or the other category.

3.3.1 Delays and discretization interval

Table 1 summarizes the values of delays used in all the models presented in this paper. The chosen discretization interval T = 1 s allows for a description of all delays

without introducing any rounding error. All the parameters presented in Table 1 were obtained by systematic experiments using 1 or 2 embodied agents.

Table 1: mean durations of the robot's maneuvers described in Subsection 2.3. Index d = dropping, p = picking up, w= wall, r = robot.

	Seed dropping and picking up	Obstacle Avoidance	Interference
Measured delay [s]	$\tau_d = \tau_p = 10$	$\tau_w = 1$	$\tau_r = 2$
Model parameter [iterations]	$T_d = T_p = 10$	$T_w = 1$	$T_r = 2$

3.3.2 Transition probabilities

Consistent with previous publications [1,3,16,22,25,26,27,28,29,30], we compute the transition probabilities from one state to another based on simple geometrical considerations and the robots' sensing and interaction capabilities (e.g., detection areas, approaching perimeters). However, instead of tightly linking space and time granularity using the smallest object appearing on the arena as we have done in the majority of the previous publications, we introduce here a computation of transition probabilities based on object encountering rates. The numerical values used for the robot speed, the approaching angle for decreasing/increasing the size of a cluster, and the different detection radii of the objects have been measured in systematic tests with one or two embodied agents and are summarized in Table 2. In Table 2, the construction (respectively, destruction) angle corresponds to the angle of the access perimeter for incrementing (respectively, decrementing) a cluster of seeds. More details are given in Martinoli et al. [25,26,27].

Table 2: parameters used in the models of the aggregation experiment. Detection distances are measured from the center of the robot to the center of the object detected.

Mean robot speed v [cm/s]	Cluster construction angle α_{inc} (n) [degrees]	Cluster destruction angle α_{dec} (n) [degrees]	Wall detection distance R_w [cm]	Seed detection distance R_s [cm]	Robot detection distance R_r [cm]	Arena size [cm^2]
8	65 if $n > 1$ 180 if $n = 1$	60 if $n > 1$ 180 if $n = 1$	6	6.4	10	80 x 80

From these experimental interaction parameters we can calculate the encountering rates used in the models: the rate at which a robot encounters the wall (γ_w), the rate at which it encounters another robot (γ_r), the rate at which it encounters a cluster of size n and decreases ($\gamma_{c_n}^{dec}$) or increases ($\gamma_{c_n}^{inc}$) the size of that cluster. To achieve this transformation we proceed as follows.

As suggested by Lerman and Galstyan [23], and Agassounon and Martinoli [3], as the robot travels through the arena at an average speed v, it sweeps out a detection area during time interval dt. This detection area is $vw_o dt$, where w_o is the width of the field of

view of a specific sensory apparatus used by the robot to detect objects of type o (see [26] for further explanatory pictures). Hence a robot will detect an object o at a rate $\gamma_o = vw_o/A_a$, where A_a is the total area of the arena. For instance, a robot will encounter a specific seed at a rate of $\gamma_{c1} = vw_s/A_a = 2vR_s/A_a$ (and any of 20 seeds at rate $20\gamma_{c1}$) and another teammate at rate $\gamma_r = vw_r/A_a = 2vR_r/A_a$ (and any of 4 teammates at rate $4\gamma_r$). $\gamma_{c_n}^{dec}$ and $\gamma_{c_n}^{inc}$ can then be calculated from γ_{c_n} by multiplying this coefficient with the ratio defined by the destruction angle/180° and construction angle/180°, respectively (see Table 2 for numerical values).

In order to calculate the encountering rate with the surrounding walls (γ_w), we used a slightly different procedure. This is mainly due to the fact that this encountering rate is heavily influenced by shape, size, and accessibility of walls: the walls have a "stretched" detection area (in contrast to the compact, rotation-symmetric detection areas of the seeds and robots), a size much larger than any other object in the arena, and are accessible only from the arena-internal side. Thus, a better way to estimate γ_w is by dividing the average speed by the mean distance traveled before encountering a wall. The mean distance we consider is the length of the side of the arena (S) minus twice the wall detection range (R_w); mathematically, $\gamma_w = v/(S-2R_w)$.

From constant rates over the time step, it is easy to calculate the corresponding transition probabilities for the small time interval T given the corresponding encountering rate:

$$(1) \quad p_x = \int_k^{k+T} \gamma_x dt = \gamma_x T$$

4 Models of Simplified Collective Experiments

Before describing implementation details and results of the full aggregation system, we introduce two simplified scenarios as examples. The first example represents a key sub-chain used in both the second example and the full aggregation system's Markov chain, which we will describe in Sections 5 and 6. In the second example, robots are characterized by the same controller used in the aggregation experiment, with the difference that seeds are not actually manipulated when found. Consequently, in the second example, no aggregation can occur.

It is worth noting that while the sets of difference equations (DEs) describing these two simplified systems at the macroscopic level are linear, though affected by time-delays, the model of the full aggregation system at the macroscopic level will be a set of nonlinear, time-delayed DEs due to the (time-dependent) modification of the environment (e.g., seeds are moved and placed in different clusters whose sizes therefore vary over time).

In the embodied simulation presented in this section we estimated the steady-state values by averaging each variable over a 1-hour simulation time window. In all the histograms that follow, the height of the embodied simulations columns represents the mean value of the steady-state estimation while error bars represent the standard deviation over 30 runs. At the macroscopic level, of course, one run suffices since only swarm's mean values are predicted.

4.1 Search and obstacle avoidance in an empty arena

Many robotic experiments involve delays for one reason or another. A delay state may simply represent a particular behavior that the robot performs for a certain duration. In this first simplified scenario, the robot's default state is the search state. While in this state, the robot may encounter the wall or another robot with probability p_w and p_r, respectively. Upon encounter with the wall (respectively, another robot) the robot enters the wall avoidance state (respectively, interference state), staying in that state for a hand-coded duration T_w (respectively, T_r). The probabilities with which the robot leaves those states are independent of the robot's interaction with the environment[d]. As described in subsection 2.2, since the default behavioral state in the aggregation experiment is the search state, a first key sub-chain of the system is represented by the search state coupled with an obstacle avoidance state (in this experiment, walls are the only obstacles in the arena) and an interference state (teammate avoidance).

Figure 3 represents graphically the state diagram of the robot's controller and at the same time the model's PFSM. The numerical values used in this example have been derived from the values in Table 1 and Table 2 using parameters for a generic obstacle: T_w = 1 s, T_r = 2 s, and p_w and p_r are functions of the setup (for example, p_w = 0.158 and p_r = 0.157 for the wall and robot encounter rates in an arena of 80 x 80 cm^2 with a group of 10 robots). W_s, W_r, and W_w represent the average numbers of workers in the search, interference, and obstacle avoidance states, respectively.

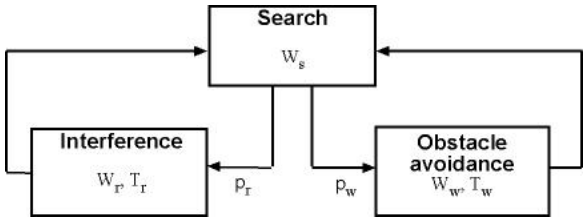

Figure 3: a simple sub-chain consisting of a search state, an interference state, and an obstacle avoidance state.

Since the state variables of the macroscopic model are represented by non-discrete quantities (real values), the PFSM of Figure 3 representing the whole swarm can be described by the following system of DEs:

$$W_s(k+1) = W_s(k) - (p_w + p_r)W_s(k) + p_w W_s(k - T_w) + p_r W_s(k - T_r) \quad (2)$$

$$W_w(k+1) = W_w(k) + p_w W_s(k) - p_w W_s(k - T_w) \quad (3)$$

[d] This is an approximation. Depending on the controller used (e.g., neural-network based, behavior-based) and the environmental density of objects, the time spent in a certain behavior may show significant variations over several situations in which the robot assumes this specific mode. A more realistic way to introduce delays in the models would be to consider not only their mean values but also, for instance, their standard deviation, both available from systematic experiments performed with one or two embodied agents.

(4) $W_r(k) = W_0 - W_s(k) - W_w(k)$

k= 0, 1 , 2, …, represents the current iteration. In general, $W_x(k)$ represents the value of the state variable W_x at time kT.

In Equations (2)-(4), W_0 represents the total number of workers in the arena while $W_s(k)$, $W_r(k)$, and $W_w(k)$ represent the number of robots in search, interference, and obstacle avoidance states at time step k, respectively. Equation (2) states that the average number of robots in the search state at iteration k decreases when one or more robots encounter either a wall or other robots and increases when some robots finish interfering with each other or finish avoiding the wall. Equation (3) represents the dynamics of the delay state of obstacle avoidance. In Equation (4) we simply exploit the conservation of the total number of robots for calculating the average number of robots in interference state. We assume that no robots exist before $k=0$ (a standard convention for this type of time-delayed DE). Mathematically speaking,

(5) $W_s(k) = W_w(k) = W_r(k) = 0 \quad \text{if} \quad k < 0$

The initial conditions for the DE system are $W(0) = [W_s(0) \quad W_w(0) \quad W_r(0)]^{\mathrm{T}} = [W_0 \quad 0 \quad 0]^{\mathrm{T}}$ (i.e., all robots are in search state at the beginning of the experiment).

4.1.1 Steady state analysis

The DE system (2)-(4) is linear. Therefore, we can analyze its steady-state using a Z-transform. Equation (2) can be transformed using the right shift and left shift theorems as follows ($W_x(z)$ representing the Z-transform of $W_x(k)$).

(6) $zW_s(z) - W_s(0) = W_s(z) - (p_w + p_r)W_s(z) + p_w W_s(z)z^{-T_w} + p_r W_s(z)z^{-T_r}$

Solving for $W_s(z)$ and applying the limit theorem brings

(7) $$W_s^* = \lim_{z \to 1}(z-1)W_s(z) = \frac{W_0}{1 + p_w T_w + p_r T_r}$$

Similarly, Equations (8) and (9) follow as

(8) $$W_w^* = \lim_{z \to 1}(z-1)W_w(z) = \frac{p_w T_w W_0}{1 + p_w T_w + p_r T_r}$$

(9) $$W_r^* = \frac{p_r T_r W_0}{1 + p_w T_w + p_r T_r}$$

$W^* = \left[W_s^*, W_r^*, W_w^*\right]^{\mathrm{T}}$ represents the state vector of the DE system above in the steady-state regime.

4.1.2 Macroscopic model and embodied simulation results

Figure 4 shows a comparison between the macroscopic model predictions and embodied simulations in the steady-state regime for different state variables and team sizes.

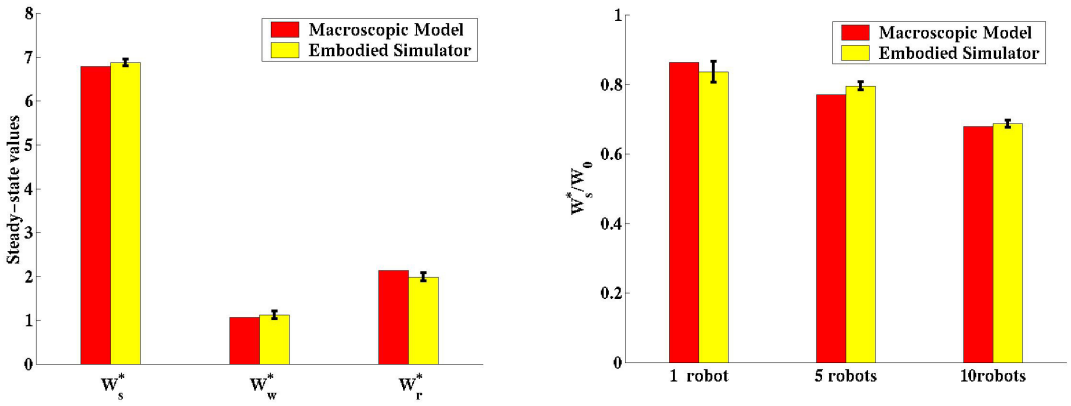

Figure 4: comparison of variables in steady-state using the macroscopic model and embodied simulations for the simple PFSM depicted in Figure 3. *Left*: W_s^*, W_w^*, and W_r^* using a team of 10 robots. *Right*: fraction of workers in search state in the steady-state regime (W_s^*/W_0) for different team sizes.

Figure 4 illustrates good agreement between the macroscopic model predictions and the simulation results for all steady-state variables used in this simple system as well as for different team sizes. While this system is linear and, therefore, the mapping between a microscopic and a macroscopic description is straightforward, this simple example illustrates that with an accurate calibration of the model's parameters, trajectories can be neglected in order to calculate the average number of robots in a given state. Minor discrepancies between the model predictions and the mean values of embodied simulation are due to the difficulties in measuring the model's parameters with systematic experiments and limited number of runs (i.e., 30) carried out with the embodied simulator. These discrepancies between models and realistic simulations on steady-state values will be discussed further in Section 7.

4.2 Distributed manipulation without aggregation

In this subsection, we compare the macroscopic model predictions and embodied simulations for a slightly more complex experiment involving seed detection, identification, and "quasi-manipulation". In this experiment, the environment contains a single seed which is not actually manipulated by the robots. In other words, a free (respectively, quasi-loaded) robot only mimics a seed-picking-up (respectively, seed-dropping) maneuver upon encounter with the seed. Hence, no seed-clustering takes place here and the initial position of the seed remains the same during the whole experiment.

This scenario is similar to the second phase of the full aggregation experiment (see Section 5) where there is a single cluster of seeds and the remaining manipulation sites

are the two tips of the cluster. A first difference between the two scenarios is that in a cluster only the two seeds at the extremities can be successfully manipulated if approached from the appropriate angle (α_{inc} or α_{dec}), whereas in a single-seed scenario, any angle of approach works. A second difference is that the side of the cluster is avoided as if it were a wall. Despite these minor differences, this simplified scenario provides insight into the steady-state of the full aggregation experiment (i.e., to evaluate the numbers of robots in different states) and helps test the validity of the assumption made for calculating the average size of the single cluster at steady-state (i.e., the number of free and loaded robots in search state is predominant over the numbers of robots in all other states).

The robot controller consists of exactly the same states used in the full aggregation experiment (compare Figure 5 and Figure 7). The only difference is that a robot, instead of actually picking up or dropping off the seed, simply emulates these operations leaving the seed where it has been placed initially. The controller is characterized by two different robot states, *free* (W_f) and quasi-*loaded* (W_l). While in both experiments a robot is free when its gripper is open and no seed is carried, in this experiment a robot is quasi-loaded when its gripper is simply closed but no seed is being carried. In contrast, in the full aggregation experiment the robot is loaded when it is actually carrying a seed. Further states of quasi-*dropping* (W_d) and quasi-*picking-up* (W_p) represent the emulation of seed-picking-up and seed-dropping operations. Since a robot cannot switch from a free to a quasi-loaded state or vice versa by simply avoiding a wall or another robot but must do so exclusively by performing either a quasi-seed-picking-up or quasi-seed-dropping maneuver, we must introduce separate states for obstacle avoidance and interference actions performed in either quasi-loaded or free situations. Hence, we distinguish the *free* robots in *wall* and *robot* avoidance states (e.g., W_{fw} and W_{fr}, respectively) from the quasi-*loaded* robots in *wall* and *robot* avoidance states (e.g., W_{lw} and W_{lr}, respectively).

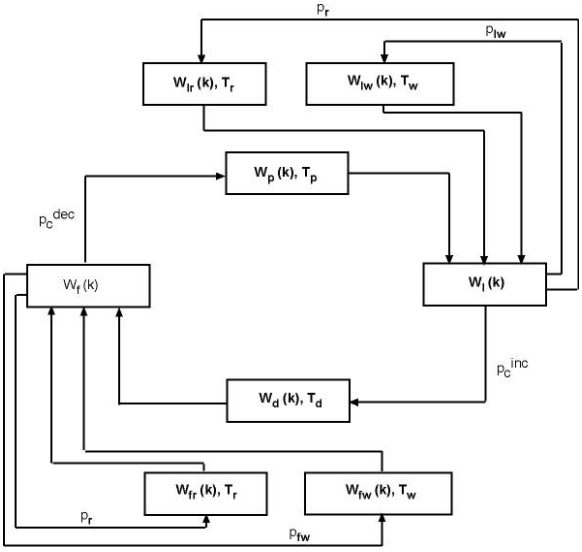

Figure 5: the PFSM representing the 'quasi-manipulation' experiment in a simple scenario with a single seed. The numerical values used in this model were derived from Table 1 and Table 2.

The following system of DEs represents the macroscopic model for this Markov-chain:

$$W_f(k+1) = W_f(k) - [p_{fw}(k) + p_r + p_c^{dec}(k)]W_f(k) + p_{fw}(k-T_w)W_f(k-T_w) + p_r W_f(k-T_r) + p_c^{inc}(k-T_d)W_l(k-T_d) \quad (10)$$

$$W_l(k+1) = W_l(k) - [p_{lw}(k) + p_r + p_c^{inc}(k)]W_l(k) + p_{lw}(k-T_w)W_l(k-T_w) + p_r W_l(k-T_r) + p_c^{dec}(k-T_p)W_f(k-T_p) \quad (11)$$

$$W_p(k+1) = W_p(k) + p_c^{dec}(k)W_f(k) - p_c^{dec}(k-T_p)W_f(k-T_p) \quad (12)$$

$$W_d(k+1) = W_d(k) + p_c^{inc}(k)W_l(k) - p_c^{inc}(k-T_d)W_l(k-T_d) \quad (13)$$

$$W_{fw}(k+1) = W_{fw}(k) + p_{fw}(k)W_f(k) - p_{fw}(k-T_w)W_f(k-T_w) \quad (14)$$

$$W_{fr}(k+1) = W_{fr}(k) + p_r W_f(k) - p_r W_f(k-T_r) \quad (15)$$

$$W_{lw}(k+1) = W_{lw}(k) + p_{lw}(k)W_l(k) - p_{lw}(k-T_w)W_l(k-T_w) \quad (16)$$

$$W_{lr}(k) = W_0 - W_f(k) - W_l(k) - W_p(k) - W_d(k) - W_{fw}(k) - W_{fr}(k) - W_{lw}(k) \quad (17)$$

This set of equations can be interpreted in a way similar to that done for Equations (2)-(5). For instance, Equation (10) tells us that the number of free robots decreases when free robots start an interference, obstacle avoidance, or seed quasi-manipulation maneuver and increases when one of these maneuvers is finished (after its characteristic delay) or when a quasi-loaded robot finishes its quasi-dropping operation and resumes search as free robot. In the last equation we exploit again the conservation of the total number of robots in the environment to calculate the number of quasi-loaded robots in interference state.

Since in this experiment we use a single cluster consisting of one seed and no incrementing or decrementing operation can take place due to the emulation of manipulation, the probabilities of interacting with the cluster remain constant over time and can be expressed as:

$$p_c^{inc}(k) = p_c^{inc}(k-T_d) = p_c^{dec}(k) = p_c^{dec}(k-T_p) = p_{c_1} \quad (18)$$

In the same way, obstacles are represented exclusively by static walls, therefore the probability of encountering an obstacle remains constant and identical for both the free and quasi-loaded robots, i.e.,

$$p_{fw}(k) = p_{fw}(k-T_w) = p_{lw}(k) = p_{lw}(k-T_w) = p_w \quad (19)$$

The robotic team size is constant during the experiment. Therefore, the probability of encountering a teammate (i.e., p_r) is constant as well.

Notice that Equations (14)-(17) are similar to Equations (2)-(5). As a matter of fact, the only difference is that the default states of the two sub-chains involving the variables

W_{fr}, W_{fw}, W_{lr}, and W_{lw} are those describing free (W_f) and quasi-loaded (W_l) robots in search instead of a single class of searching robots, as in the PFSM of Subsection 4.1. The two sub-chains can be seen in the lower left and upper right corner of Figure 5.

The initial conditions for the DE system are $W(0) = [W_f(0) \;\; W_l(0) \;\; W_p(0) \;\; W_d(0) \;\; W_{fr}(0) \;\; W_{fw}(0) \;\; W_{lr}(0) \;\; W_{lw}(0)]^{\mathrm{T}} = [W_0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^{\mathrm{T}}$ (i.e., all robots are in free-search state at the beginning of the experiment).

4.2.1 Steady-state analysis

The same methodology as that used in Subsection 4.1 (i.e., Z-transform and limit theorem) can be used here for finding the values of state variables in stationary regime. However, by analyzing Figure 5 more carefully, we notice that the PFSM is symmetric. Moreover, since T_p and T_d are identical (see Table 1) the steady-state values of the variables in symmetric states must be the same. In other words, $W_f^* = W_l^*$, $W_p^* = W_d^*$, $W_{fr}^* = W_{lr}^*$, and $W_{fw}^* = W_{lw}^*$. Solving the DE system above, we obtain

(20) $$W_p^* = W_d^* = p_{c_1} T_p W_f^*$$

(21) $$W_{fw}^* = W_{lw}^* = p_w T_w W_f^*$$

(22) $$W_{fr}^* = W_{lr}^* = p_r T_r W_f^*$$

Finally, by using the conservation of the total number of robots it follows that

(23) $$W_f^* = W_l^* = \frac{W_0 / 2}{1 + p_w T_w + p_r T_r + p_{c_1} T_p}$$

4.2.2 Macroscopic Model and embodied simulation results

Figure 6 shows a comparison between macroscopic model predictions and embodied simulations for different state variables and swarm sizes. Due to the symmetric properties of the PFSM in Figure 5, the other state variables can be derived via Equations (20)-(23) and achieve a similar distribution to that plotted in this histogram when measured with Webots. We notice good agreement between the mathematical model predictions and the embodied simulations. In particular, the results presented in Figure 6 confirm that, due to the short time lapses needed to perform seed-dropping/-picking operations and obstacle (wall and robots) avoidance maneuvers, the number of (free and quasi-loaded) robots in search state is predominant over those in the other states in the steady-state regime. This assumption is used in the steady-state analysis of the full aggregation experiment in Section 5. Minor discrepancies between models and realistic simulations on steady-state values will be discussed further in Section 7.

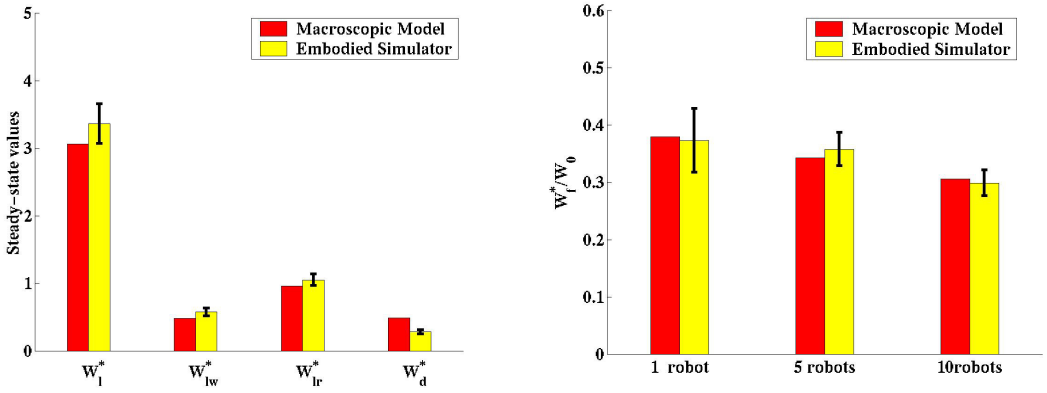

Figure 6: comparison of variables in steady-state using the macroscopic model and embodied simulations for the PFSM of Figure 5. *Left*: subset of state variables (W_l^*, W_{lw}^*, W_{lr}^*, and W_d^*) using a team of 10 robots. *Right*: normalized steady-state value of the mean number of free robots (W_f^*/W_0) for different team sizes.

5 Aggregation Experiment without Worker Allocation

In most previous aggregation experiments [4,15,25,26], the size of the working team was kept constant during the whole process. Furthermore, the experiments were monitored and terminated when a single cluster arose in the arena. Thus, the (destructive) effect of the lack of a mechanism to allow the robots to stop working was never clearly studied. Here we do not end the experiment when the robots create a single cluster of seeds, instead we let the experiment run for 10 hours. The outcome presented in Subsection 5.3 justifies the need for a distributed mechanism that enables each robot to switch off when the aggregation task is finished. In concordance with previous works in cluster formation and aggregation [1,4,25,26], we are using two primary team performance measurements: the average cluster size and the average number of clusters. Additionally, since in some of our experiments the working team size is time-varying, we introduced a metric computing the average number of active workers in the environment.

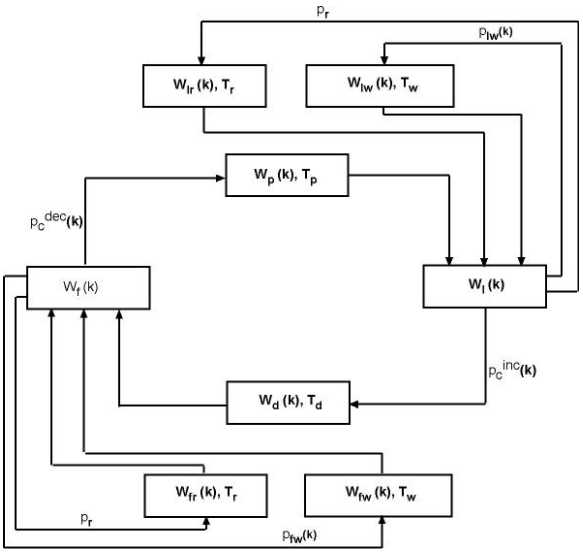

Figure 7: PFSM representing the whole robotic team performing the aggregation task.

Figure 7 shows a PFSM representing the whole robotic team performing the aggregation task. This representation does not include the distributed worker allocation mechanism introduced and detailed in Section 6. Notice the time-varying coefficients characterizing several transition probabilities (p_{lw}, p_{fw}, p_c^{inc}, and p_c^{dec}) in this PFSM: they are the result of the aggregation process. As a matter of fact, the PFSM of Figure 7 has exactly the same structure as that of Figure 5 but, while that of Figure 5 has constant transition probabilities between all its states (as no aggregation process takes place), that of Figure 7 has time-varying coefficients, which reflect environmental modifications due to the aggregation process. In other words, the robots' controller is essentially the same in both experiments; therefore, the resulting PFSMs' structures are the same. However, the transition probabilities are different because they are conditioned by the coupling of the robot's PFSM with the environment, which is static in the former experiment and dynamic (via the active manipulation of the robots) in this case.

Notice that the environment could also be represented with its own PFSM whose states are represented by clusters of different sizes, as we did in [25]. In such a representation, the representation of the environmental PFSM would have time-varying transition probabilities that would be functions of the mean current availability of the workers in one state or the other in the robotic PFSM, as we will see in the next subsection (see Equation (29) in particular).

5.1 Macroscopic model

The macroscopic model for this aggregation experiment is represented again by Equations (10)-(17). However, here W_f refers to *free* robots, W_l to *loaded* robots, W_p to robots *picking* up seeds, and W_d to robots *dropping* off seeds (as opposed to the free, quasi-loaded, quasi-picking-up, and quasi-dropping robots of Subsection 4.2).

As mentioned above, the major difference between the two experiments is that the probabilities with which the robots encounter obstacles and seeds to manipulate are functions of time in the aggregation experiment rather than being time invariant. Concretely, the aggregation process modifies wall-encountering probabilities since the sides of clusters represent obstacles robots must avoid and, of course, reduces the number of available seeds to manipulate. Due to actual manipulation, Equation (18) is transformed as follows:

$$p_c^{inc}(k) = \sum_{n=1}^{M_0} p_{c_n}^{inc} N_n(k) \quad (24)$$

$$p_c^{inc}(k - T_d) = \sum_{n=1}^{M_0} p_{c_n}^{inc} N_n(k - T_d) \quad (25)$$

$$p_c^{dec}(k) = \sum_{n=1}^{M_0} p_{c_n}^{dec} N_n(k) \quad (26)$$

$$p_c^{dec}(k - T_p) = \sum_{n=1}^{M_0} p_{c_n}^{dec} N_n(k - T_p) \quad (27)$$

M_0 and N_n represent the total number of seeds in the arena and the number of clusters of size n, respectively. The dynamics of N_n are described later in this subsection.

Equation (19) is no longer valid since the probability of encountering a wall evolves with seed aggregation. The probability of encountering a wall as a free or loaded robot is also slightly different since the free and loaded workers cannot approach a cluster for manipulating it with the same angle (the incrementing and decrementing angles are different, see Table 2) and, as a consequence, they do not perceive the same obstacle length upon encounter with the side of a cluster. Therefore, p_{fw} and p_{lw} must be considered as two different, time-varying parameters of the model. p_{fw} can be expressed as follows (note that to obtain p_{lw} one must replace α_{dec} by α_{inc} in the expression below):

$$p_{fw}(k) = p_w + p_{fcw}(k) = \frac{vT}{S - 2R_w} + T\sum_{n=1}^{M_0} \frac{2R_s v N_n(k)}{S^2} \frac{2\pi R_s + 4(n-1)r_s - 2\alpha_{dec}(n)R_s}{2\pi R_s + 4(n-1)r_s} \quad (28)$$

In the right-hand side of Equation (28), the first term of the main sum (i.e., $p_w = vT/(S-2R_w)$) represents the usual probability with which a robot encounters the surrounding border separating the working and the parking zones as for the experiments of Section 4. The second term of the main sum (i.e., p_{fcw}) corresponds to the probability with which a free robot encounters the body side of a cluster of size n. In this second term, the first expression (whose denominator is S^2) refers to the probability of encountering a cluster of size n and the other expression corresponds to the probability of an unsuccessful approach.

The last variables of the aggregation system are the numbers of clusters of different sizes that arise and/or disappear over time during the aggregation process. We describe the quantitative dynamics of the clusters as follows:

$$N_n(k+1) = N_n(k) + [p^{dec}_{c_{n+1}} N_{n+1}(k-T_p) - p^{dec}_{c_n} N_n(k-T_p)] W_f(k-T_p) + [p^{inc}_{c_{n-1}} N_{n-1}(k-T_d) - p^{inc}_{c_n} N_n(k-T_d)] W_l(k-T_d) \quad (29)$$

Equation (29) shows a system of M_0 DEs, M_0 being the total number of different cluster sizes (in this experiment it corresponds to the total number of seeds used in the experiment). The DE system represents the environmental PFSM whose individual states are defined by the cluster size. Equation (29) tells us that the number of clusters of size *n* can be increased by either removing a seed from a cluster of size *n+1* or by dropping a seed in a cluster of size *n-1*. The number of clusters of size *n* can instead be decreased by either picking up a seed or dropping one in the cluster.

5.2 Steady-state analysis

As shown by Martinoli et al. in [27], if the robots do not withdraw (i.e., the team size remains constant), do not drop a seed unless it is next to another seed, and do not pick up an internal seed from a cluster, the number of clusters monotonically decreases and eventually a single cluster always arises. A practical way of defining the steady-state value is that it represents the average size of the single cluster remaining in the environment as the robots continue picking up and dropping seeds at its extremities.

Based on the assumption above, intuitively, the size of the remaining cluster corresponds solely to the total number of seeds in the arena minus the number of those being held by the loaded workers, i.e.,

$$L^* = M_0 - W_l^* - W_{lw}^* - W_{lr}^* - W_d^* \quad (30)$$

where L^* is the average size of the single cluster remaining in the arena in the steady-state. L^* is obviously a function of several steady-state variables whose values cannot be derived in closed form based solely on the equations above. Therefore, we proceed with an approximation of L^* by neglecting the number of robots in obstacle avoidance, interference, seed-picking-up, and seed-dropping states at steady-state[e], i.e., $W_f^* + W_l^* \approx W_0$. Thus, by introducing Equation (24)-(27) in Equation (10)-(17), setting all delay state variables to zero, and calculating the steady-state situation ($W_x(k+1) = W_x(k)$) we obtain:

$$\frac{W_f^*}{W_l^*} = \rho \quad (31)$$

$$L^* \approx M_0 - \frac{W_0}{1+\rho} \quad (32)$$

[e] This assumption holds as long as the density of robots in the arena is low and the different maneuvers, e.g., obstacle avoidance, interference, seed-picking-up/dropping durations are short. Its validity has been demonstrated in Subsection 4.2.2.

where $\rho = \frac{p_{c_n}^{inc}}{p_{c_n}^{dec}} = \frac{\alpha_{inc}}{\alpha_{dec}}$ is a constant for all n such that $1 < n < M_0$.

This approximation of the average size of the unique cluster remaining in the arena agrees with the embodied simulation results presented in the next section. Note that L^* is a decreasing function of the team size W_0.

5.3 Results

Figure 8 and Figure 9 present the model predictions and the simulation results of the aggregation experiment without the use of any worker allocation algorithm with groups of 1, 5, and 10 robots in an 80 x 80 cm^2 arena. Figure 8 and Figure 9, left, present the increasing average size of the clusters over time while Figure 8 and Figure 9, right, show the decreasing average number of clusters over time. Each experiment characterized by a different team size has been repeated for 30 runs and the depicted error bars represent the standard deviation over runs. Good agreement between the results collected at both implementation levels shows how reliable the macroscopic model's prediction of the outcome of the aggregation experiment is.

With a team of 10 robots (Figure 9, left), the aggregation process clearly has two phases. In the first phase, the mean cluster size increases steadily from 1 seed to about 15 seeds and in a second phase, the mean cluster size remains, on average, constant around 15 seeds. Similarly, during the first phase, Figure 9, right, shows that the average number of clusters decreases asymptotically from 20 to about 1 and then remains close to 1 during the second phase of the aggregation process. This is clearly a side effect of the lack of a mechanism to allow workers to stop performing the aggregation task once a single cluster arises, which is confirmed by the fact that 10 hours into the aggregation process, the average cluster size built by the team of 5 robots (see Figure 8, left) is larger than that obtained with the team of 10 robots.

The results in Figure 8 and Figure 9 can be explained by the fact that, once a single cluster arises only two manipulation sites remain in the environment (i.e., the two tips of that cluster); the seed pick-up and drop-off probabilities (i.e., $p_{c_n}^{dec}$ and $p_{c_n}^{inc}$) are empirically very close, therefore at any given time during the last phase of the aggregation process, on average, half of the active workers will be carrying a seed and the other half will not. This explains why the average cluster size at steady-state (i.e., L^*) as expressed in Equation (32), is a decreasing function of the team size. For instance, $L^* \approx 15.2$ *seeds* for the team of 10 robots and Figure 9 shows that after 10 hours of aggregation the average cluster size built by that team is close to this value for both the macroscopic model and the embodied simulator.

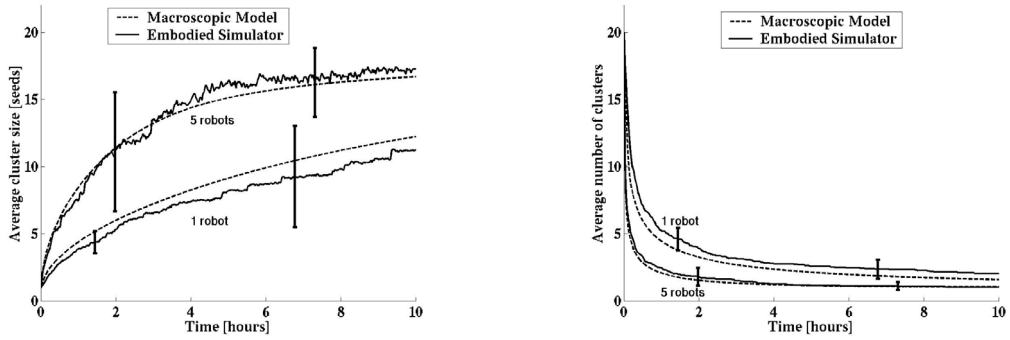

Figure 8: results of aggregation experiment with groups of 1and 5 robots and 20 seeds in an 80 x 80 cm^2 arena. Team sizes are constant. *Left*: average cluster size over time. *Right*: average number of clusters over time.

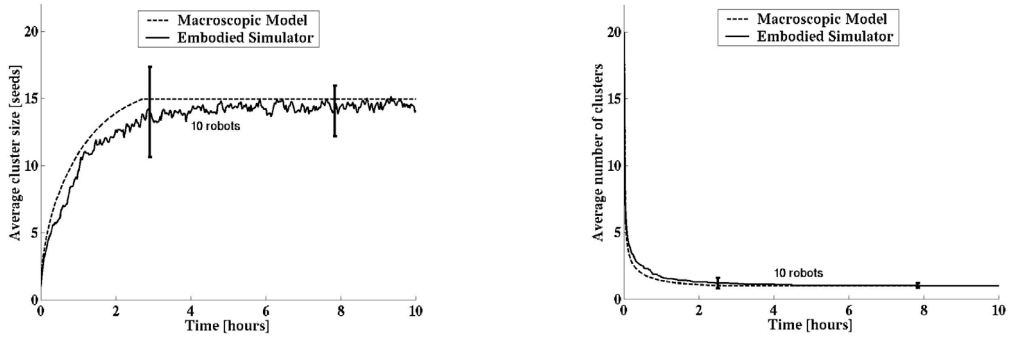

Figure 9: results of aggregation experiment with a group of 10 robots and 20 seeds in an 80 x 80 cm^2 arena. The team size is constant. *Left*: average cluster size over time. *Right*: average number of clusters over time.

6 Aggregation Experiment with Worker Allocation

In this section, we introduce a simple, distributed worker allocation algorithm that allows a team of robots to increase its efficiency as a whole by withdrawing an appropriate number of workers as a function of the demand, which is intrinsically defined by the aggregation process. We will show that our macroscopic model can be extended to capture also this new type of controller. In this scenario, even the last parameter, which has been constant so far (i.e., p_r, see Equations (10)-(17)) will become time-varying and will introduce an additional nonlinear coupling factor among the equations.

Generally speaking, we would like to implement fully distributed algorithms which are based exclusively on local perception and local communication and enable the team to allocate the right number of workers to one or several specific tasks. The demand related to each task may evolve over time either because of a driving factor intrinsic to

the environment or, as in the aggregation experiment, because of the actions of the robots on the environment.

In our case, intuitively, we can imagine that at the beginning of the aggregation there are many possible manipulation sites (i.e., several scattered seeds) that allow for a parallel work of several robots. As the aggregation process goes on, the number of these sites is reduced and having more robots competing for the same manipulation sites decreases their efficiency. Therefore, reducing the number of active robots during the last phase of aggregation should consequently increase the team efficiency (number of workers allocated per amount of work) as well as the size of the single cluster. The main challenge here is to find a simple mechanism that allows each robot to autonomously evaluate the progress in task completion so as to be able to decide whether to continue performing the task or not. SI can offer a variety of biologically inspired, distributed solutions to this type of problems. We propose here a SI-based solution in the form of a threshold-based, distributed worker allocation algorithm.

6.1 A threshold-based, distributed worker allocation algorithm

In threshold-based systems, the 'propensity' of any agent to act is governed by a *response threshold*. If the demand is above the agent's threshold then that agent continues to perform the task; conversely, if the demand is below its threshold then the agent stops performing that particular task. In the algorithm presented in this paper, the time an agent spends before finding some work to accomplish (i.e., to pick up and drop a seed) represents the agent's estimation of the demand stimulus associated with the aggregation task.

Our current worker allocation algorithm is as follows. When a robot has not been able to work (i.e., to pick up and drop a seed) for a reasonable amount of time, its propensity to accomplish that particular task is decreased. If the stimulus falls below a certain threshold (i.e., if the amount of time spent in the search of work is above a given search time-out T_s), a deterministic switching mechanism prompts the robot to leave the working zone and rest in the surrounding parking zone for the remaining duration of the experiment. In other words, our worker allocation algorithm has been particularly designed for an irreversible process such as aggregation: robots become inactive from an active mode and not vice versa as seeds are irreversibly gathered in clusters of increasing size. Furthermore, a loaded robot that decides to become inactive cannot do so until it finds an appropriate spot (i.e., one tip of a cluster) to drop the seed.

Thus, with this simple algorithm characterized by a single threshold, each robot is able to estimate the aggregation demand locally and decide whether to work or rest with no need for a central controller. This task allocation mechanism is similar to that observed in some ant colonies [7, 38] for which it has been shown that an individual performs a task as long as the demand stimulus for the task, e.g., a pheromone concentration, exceeds the individual's threshold for that particular task.

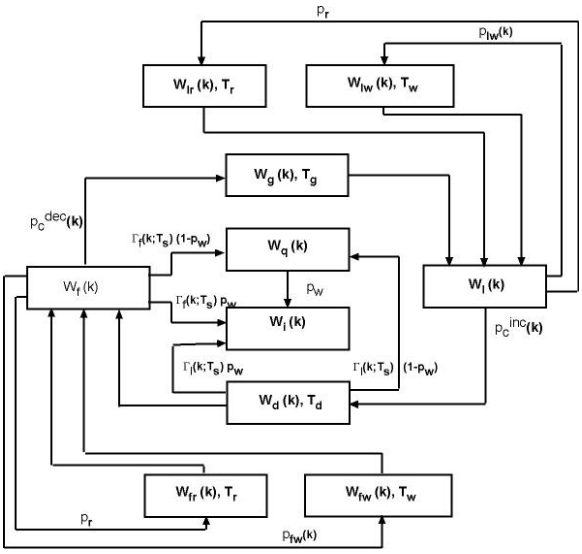

Figure 10: PFSM representation of the complete aggregation experiment with distributed worker allocation. The numerical values used in this PFSM have been derived from Table 1 and Table 2.

The threshold distribution among agents could be performed in several ways. In this paper, we use the simplest possible distribution: we assign the same response threshold to all the agents, obtaining in this way a homogeneous and fully scalable team from a control point of view. However, the resulting agents' behavior (rhythm of activity) is not identical since it is based on the local, private assessment of the aggregation demand, which is represented by the seed distribution in the environment. In other words, diversity in activity is created by exploiting the intrinsic noise of the system due to local perception and additional noise sources on simulated (miniature) sensors and actuators.

6.2 Macroscopic model

The PFSM of the complete aggregation experiment with distributed worker allocation depicted in Figure 10 differs from that in Figure 7 by two new states, the *quit* and the *idle* states represented by W_q and W_i, respectively. The quit state corresponds to the state of the robots that have autonomously decided to stop working and started traveling from the working zone to the surrounding parking zone (see Figure 1). The idle state corresponds to the state of the robots that have reached the parking zone after stopping performing the aggregation task. For the sake of brevity, we introduce here only the equations that describe the dynamics of the new variables and those of the old variables, i.e., those from Section 5, whose dynamics have changed due to the task allocation mechanism.

$$
\begin{aligned}
W_f(k+1) = \; & W_f(k) - [p_{fw}(k) + \tilde{p}_r(k)]W_f(k) - \sum_{n=1}^{M_0} p_{c_n}^{dec} N_n(k) W_f(k) \\
& + p_{fw}(k - T_w)W_f(k - T_w) + \tilde{p}_r(k - T_r)W_f(k - T_r) - W_f(k)\Gamma_f(k; T_s) \\
& + \sum_{n=1}^{M_0} p_{c_n}^{inc} N_n(k - T_d)W_l(k - T_d)(1 - \Gamma_l(k; T_s))
\end{aligned}
\tag{33}
$$

$$
\begin{aligned}
W_q(k+1) = \; & W_q(k) + W_f(k)\Gamma_f(k; T_s)(1 - p_w) - p_w W_q(k) \\
& + \sum_{n=1}^{M_0} p_{c_n}^{inc} N_n(k - T_d)W_l(k - T_d)\Gamma_l(k; T_s)(1 - p_w)
\end{aligned}
\tag{34}
$$

$$
\begin{aligned}
W_i(k+1) = \; & W_i(k) + \sum_{n=1}^{M_0} p_{c_n}^{inc} N_n(k - T_d)W_l(k - T_d)\Gamma_l(k; T_s)p_w + p_w W_q(k) \\
& + p_w W_f(k)\Gamma_f(k; T_s)
\end{aligned}
\tag{35}
$$

As previously stated, here $\tilde{p}_r(k) = p_r[W_0 - 1 - W_i(k)]$ is a time-varying parameter as the team size changes over time due to the distributed worker allocation. In the equations above, p_w translates a successful attempt by the robots to leave the working zone for resting in the parking zone, and 1- p_w a failure to do so. Robots that are trying to leave the working zone remain in the quit state until they are successful in doing so.

$\Gamma_f(k;T_s)$ and $\Gamma_l(k;T_s)$ correspond to the fractions of "disappointed" free and loaded workers. They represent those free (respectively, loaded) robots that have been unsuccessful in finding a single seed to pick up (respectively, a site at which to drop a seed) over the past time period [k-T_s, k] (respectively, [k-T_d-T_s, k-T_d]). Note that $\Gamma_l(k;T_s)$ is defined over the time period [k-T_d-T_s, k-T_d] because robots are not allowed to leave the arena with a seed. Therefore, a loaded robot that decides to stop working has to find an appropriate spot to drop the seed it is carrying before leaving. Hence, Γ_l only applies to the loaded robots that have finished a seed-dropping maneuver begun T_d time steps ago. The last two terms in the right-hand side of Equation (33) represent a change compared with Equation (10). The first of these two terms translates the dynamics of the free robots that decide to stop working and the other term translates the dynamics of the robots that enter the search state after releasing a seed without the intention to stop working. Equation (34) expresses the dynamics of W_q, the number of robots that have decided to stop working but have not yet succeeded in leaving the working zone. Finally, Equation (35) expresses the dynamics of the robots in idle state as the number of robots in that state depends on the robots that successfully cross the border on their way to the parking zone independently of whether their decision to stop working was made while they were free or loaded.

To calculate $\Gamma_f(k;T_s)$ (and similarly, $\Gamma_l(k;T_s)$), we use a method similar to that used in [28,29,30] for calculating the fraction of robots abandoning a stick-pulling action. The probability per iteration for a free robot of not finding a seed to pick up during the time interval [k-1, k] is $1 - p_c^{dec}(k-1)$. The overall probability of a free robot not finding a seed to pick up during the time interval [k- T_s, k] can be calculated as follows.

(36) $$\Gamma_f(k;T_s) = \prod_{j=k-T_s}^{k} \left[1 - p_c^{dec}(j)\right]$$

Similarly, $\Gamma_l(k;T_s)$ is calculated as follows

(37) $$\Gamma_l(k;T_s) = \prod_{j=k-T_{ds}}^{k-T_d} \left[1 - p_c^{inc}(j)\right]$$

$T_{ds} = T_d + T_s$ and p_c^{inc} (respectively, p_c^{dec}) is calculated with Equation (24) (respectively, Equation (26)).

6.3 Steady-state analysis

Being the system nonlinear, also in this case a thorough analysis of the steady-state is better conducted by equating $W_x(k+1)$-$W_x(k)$ to zero in the DEs. In particular, at steady-state Equation (35) becomes:

(38) $$0 = \sum_{n=1}^{M_0} p_{c_n}^{inc} N_n^* W_l^* \Gamma_l^*(T_s) p_w + p_w W_f^* \Gamma_f^*(T_s) + p_w W_q^*$$

Each term on the right-hand side of Equation (38) is non-negative, so it follows that

(39) $$W_q^* = 0$$

(40) $$W_f^* \Gamma_f^*(T_s) = 0$$

(41) $$W_l^* \Gamma_l^*(T_s) \sum_{n=1}^{M_0} p_{c_n}^{inc} N_n^* = 0$$

(42) $$W_f^* \sum_{n=1}^{M_0} p_{c_n}^{dec} N_n^* = W_l^* \sum_{n=1}^{M_0} p_{c_n}^{inc} N_n^*$$

In fact, if the arena is not overcrowded and the total number of agents is smaller than the number of seeds (this latter condition prevents the scenario in which all the seeds are picked up leaving no drop-off sites remaining), there will always be at least one cluster of one or more seeds present in the arena. Thus, there exist at least two probability values, e.g., p_{min} and p_{max},[f] such that $0 < p_{min} \leq p_c^{dec}(k) \leq p_{max} < 1$ for all time steps k. Hence

(43) $$(1 - p_{\max})^{T_s} \leq \Gamma_f(k;T_s) \leq (1 - p_{\min})^{T_s}$$

[f] Two possible values of p_{min} and p_{max} are $p_{c_{M_0-W_0}}^{dec}$ and $M_0 \cdot p_{c_1}^{dec}$, respectively.

Equation (43) implies two consequences: first, that $\lim_{T_s->\infty} \Gamma_f(k;T_s) = 0$ and second, that $\Gamma_f^*(T_s) = \lim_{k->\infty} \Gamma_f(k;T_s)$ is never null for a fixed T_s. The second consequence and Equations (39)-(42), in turn, imply that there exists a unique steady-state which is the following: $W_f^* = W_l^* = W_q^* = W_{fw}^* = W_{fr}^* = W_{lw}^* = W_{lr}^* = 0$ and $W_i^* = W_0$. This means that in a system where the proposed distributed task allocation mechanism is implemented, all the agents will eventually stop working. In other words, if the probability for a robot to become "disappointed" is different than zero and the deactivation mechanism is deterministic and irreversible, sooner or later a site to manipulate may not be found and the last active robot will switch off. Note that in reality it may take a long observation time before obtaining this stable steady-state, depending on the value of the fixed activity threshold.

Nevertheless, when the aggregation experiment with worker allocation is observed over a relatively short period of time (which, in fact, depends on the number of seeds to cluster and the total number of robots performing the task), it has different outcomes depending on both the initial setup and, more strongly, on the fixed time-out T_s. On the one hand, if T_s is too small, the propensity of the robots to work will be reduced substantially and this will result in the whole team stopping working too early preventing the seeds from being gathered in a single cluster. On the other hand, if T_s is too large, the outcome of the experiment will be much like that of the scenario without worker allocation if observed over a relatively short period of time. We will discuss this further in Subsection 7.3.

6.4 Results

Figure 11 and Figure 12 present the outcome of the aggregation experiment using the proposed distributed worker allocation algorithm with teams of 1, 5, and 10 robots in an 80 x 80 cm^2 arena. Each aggregation run lasted 10 hours. All error bars represent the standard deviations among 30 runs. For all the results presented in this subsection, we hand-coded T_s = 25 minutes. This value of T_s is very close to the optimal threshold value obtained using a systematic search as showed in Subsection 7.3.

Figure 11 and Figure 12 show that, in contrast to the case without worker allocation, the average cluster size remains an increasing function of time for all three group sizes within the 10-hour experimental time window. When using a single worker, the average cluster size over time is the same as that obtained in the scenario of Section 5 and the robot usually does not leave the working zone; with a group of 5 workers, although the average cluster size over time is similar to that obtained in the scenario without worker allocation, here the average number of workers necessary to obtain the same results is substantially lower, as on average, only about half of the agents remain in the arena after 10 hours. The most important improvement comes with a large team of 10 robots. In fact, with 10 workers, during the second phase of the aggregation, the average cluster size remains an increasing function of time, eventually reaching 19 seeds, nearly the largest value possible, while the number of active workers in the environment decreases such that after 10 hours only about 2 workers remain active.

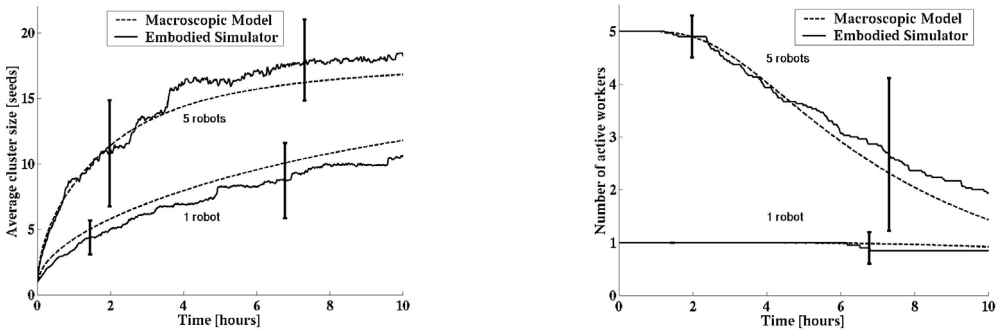

Figure 11: results of aggregation experiment with worker allocation and groups of 1and 5 robots with 20 seeds in an 80 x 80 cm^2 arena. *Left*: average cluster size over time. *Right*: average number of active workers over time.

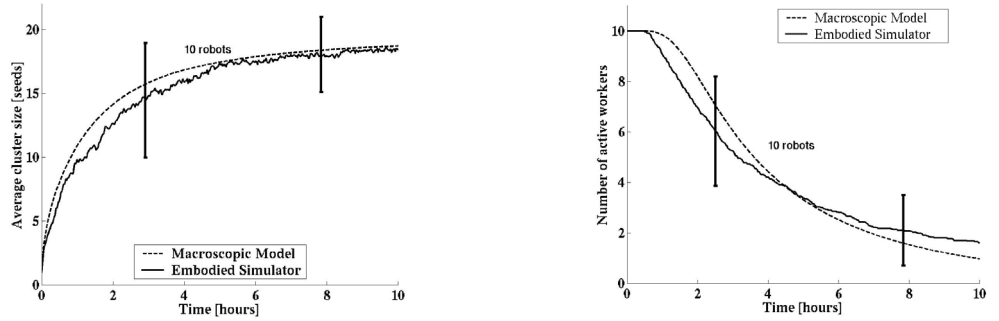

Figure 12: Results of aggregation experiment with worker allocation and a group of 10 robots with 20 seeds in an 80 x 80 cm^2 arena. *Left*: average cluster size over time. *Right*: average number of active workers over time.

7 Discussion

In this paper we have presented macroscopic models of incremental complexity from simple examples in Section 4 representing scenarios characterized by no modification of the environment to the scenario presented in Section 6, a model that captures not only environmental modification but also changes in the robotic team size. In each of these sections, we have successfully validated the macroscopic models with embodied

simulation results comparing the two levels of implementation based on either state variables in steady-state regimes (Section 4) or more high-level metrics related to the aggregation or worker allocation processes (Sections 5 and 6).

In this section, we discuss the usefulness of macroscopic models for understanding aggregation experiments, generalization to tasks other than aggregation, control optimization, the difficulties encountered in generating quantitatively accurate models, and finally, the intrinsic limitations of such models.

7.1 Capturing randomness with macroscopic deterministic DE

All versions of the aggregation experiment are permeated by randomness. Without randomness we would have neither aggregation to a single cluster nor the ability to implement such simple, distributed worker allocation algorithm regulating the team activity. Furthermore, qualitatively speaking, randomness decreases the efficiency of the robotic system but increases its robustness to deadlock situations and unfavorable initial conditions. System randomness has its origin in the local perception and in the concurrent, autonomous action of each robot as well as in the intrinsic noise of miniature sensors and actuators faithfully mimicked in the Webots simulator. Thus, a hand-coded, deterministic robot behavior is unavoidably transformed through the interactions between the robot and the environment and between the robot and its teammates into a stochastic behavior. Our macroscopic models, which at first glance may appear deterministic, capture the randomness of the system by using probabilities and probabilistic rates of occurrence of different events. In the rest of this subsection, we will present a few concrete examples of randomness-based mechanisms exploited in, and essential to, the aggregation experiments described in Sections 5 and 6.

The first example we would like to mention is the randomness in the robots' trajectories. If the trajectory of each robot were predetermined and noise-free, we would need a central supervisor with a view of the entire arena to coordinate the movements of the robots in such a way that from each possible initial scattering of the seeds, the robots could reach all of them and start the aggregation process without getting entangled. Just having deterministic trajectories without a central planner with a global view of the arena or a powerful on-board navigation system (e.g., GPS) may generate solutions that, depending on the initial conditions of the system, enable the robots to exclusively cover a partition of the arena without reaching all the seeds.

A second example is concerned with the creation of a *single* cluster at the end of the aggregation process. Due to the mechanical constraints of the robots, the approaching angles for incrementing and decrementing the size of a cluster are almost the same. Therefore, without randomness, we could have deadlock situations (e.g., 2 large clusters consisting of half of the seeds initially scattered in the arena) in which one seed is picked up from one cluster and dropped into the other, picked up again from this latter cluster and dropped, once again, into the original cluster, and so on. We could see this as a form of dynamic equilibrium different from the single-cluster situation. Such a scenario has never appeared in any of our aggregation experiments. This is explained by the fact that the independent (or uncoordinated) actions of all the robots in the arena generates relevant variations on the cluster sizes so that, each time any cluster is reduced to a single seed, that single-seed cluster is irreversibly (and deterministically) removed from the arena. Of course, this achievement was facilitated by the fact that the number of seeds

was never much larger than the number of robots (the worst case studied is a 20 to 1 ratio). With a much higher seed-to-robot ratio we would probably have significantly greater difficulty systematically achieving the single-cluster result.

The third example is concerned with the core principle of the worker allocation algorithm presented in Section 6. Although all the teammates are endowed with the same activity threshold (i.e., a homogeneous team), the noise inherent in individual estimation of the aggregation demand due to local perception of the environment guarantees that robots deactivate at different times during the aggregation process. Thus, local perception and other noise sources create diversity in the decision to quit; as a consequence, the occurrence of deactivation increases gracefully over time as opposed to the sudden, collective withdrawal that would be seen if all agents were characterized by the same threshold and had a global perception of the environment. Finally, as shown in Subsection 6.3 and similar to the previous example, here randomness in combination with an irreversible, deterministic deactivation algorithm leads to the targeted situation in which all the robots are eventually inactive.

7.2 Understanding more sophisticated worker allocation algorithms

In this subsection we briefly describe three threshold-based, slightly more sophisticated worker allocation algorithms derived from the basic algorithm introduced in Section 6. Two of them have been already published in [2]; the third one is a combination of the previous two. The focus here is not on the algorithms themselves but rather on the insights that can be gained by using macroscopic models to explain the observed performance differences among the algorithms.

The algorithms differentiate themselves on two orthogonal axes: the threshold may be *fixed or variable*, and there may be shared (*public*) or not shared (*private*) demand estimation among teammates via local explicit communication.

According to this classification, the algorithm presented in Section 6 would be called a Private Fixed-Threshold algorithm (*PrFT*). The Private Variable-Threshold algorithm (*PrVT*) allows each worker to calculate its own activity threshold. This is done in two phases. During the first phase the robot works and evaluates the availability of seeds in the environment. At the end of that phase, the robot calculates its threshold based on the average amount of time it spent before finding work to accomplish over a fixed number of successful attempts. During the second phase, the robot works as in the PrFT algorithm.

Using the Public Fixed-Threshold algorithm (*PuFT*) algorithm, the workers are endowed with a peer-to-peer communication capability. Upon encountering another robot, robots exchange their individual estimations of the demand. This allows each robot to gather information about the work demand from both its individual experience and that of other teammates. The Public Variable-Threshold algorithm (*PuVT*) combines the individual threshold calibration mechanism of PrVT and the information exchange capability offered by PuFT.

We introduced a metric to measure the efficiency of the algorithms and assess their cost effectiveness. The metric is a cost function corresponding to the sum of the squared distances of the current values of the average cluster size, the number of clusters, and the number of active workers to the desired result (e.g., 20 seeds in a single cluster with no worker remaining active). More details are given in [2].

We tested the robustness of these algorithms when facing static, environmental changes and dynamical, external changes: we optimized the performance of each algorithm in Arena1 (80 x 80 cm^2 and 20 seeds) then measured the same performance in Arena2 (178 x 178 cm^2 and 20 seeds) and Arena3 (80 x 80 cm^2, 20 seeds at start, and 5 additional introduced 2 hours into the aggregation process). The performance of the algorithms is summarized in Table 3. The values in bold correspond to the best performance, i.e., lowest cost within the margin of one standard deviation. The last row of Table 3 represents the results of the aggregation experiment without the use of any worker allocation mechanism.

Table 3: performance summary: the metric is the mean Integrated Cost ± standard deviation.

Algorithm	Arena1	Arena2	Arena3
PrFT	**138.9 ± 7.0**	324.9 ± 10.8	154.5 ± 7.9
PrVT	155.1 ± 8.0	**231.9 ± 10.7**	152.2 ± 8.7
PuFT	**138.2 ± 6.9**	337.6± 10.7	**122.4 ± 6.4**
PuVT	**141.3 ± 5.2**	**227.2 ± 9.4**	**134.2 ± 9.1**
W/o WA	227.4 ± 4.8	310.8 ± 8.8	197.2 ± 5.9

7.2.1 Qualitative performance analysis of the worker allocation algorithms

Before using the macroscopic model as a tool for analyzing the performance of the algorithms, we would like to first qualitatively explain the performance differences shown in Table 3.

PrFT – Due to its *a priori* fixed response threshold value, the agents behave sub-optimally in an environment different from that for which it was optimized. For instance, when performing the same aggregation task in a 178 x 178 cm^2 arena, the average size of the clusters they create is smaller than the average size of those created by the team using the PrVT algorithm because the agents withdraw too soon.

PrVT – The density of manipulation sites is higher in the smaller arena and the robots are more likely to encounter them than in the larger arena. In response to this difference in density of manipulation sites, variable-threshold workers autonomously set their response thresholds higher in Arena2. Therefore, they stay active longer in the larger arena than in the smaller one and this in turn allows them to continue performing the task, as most seeds are not gathered yet into a single cluster. However, this algorithm still relies exclusively on individual local estimation of environmental modifications in order to assess the aggregation demand, a property that makes the team reaction time to external, environmental perturbations quite slow (see the performance of PrVT in Arena3 presented in Table 3). Furthermore, the absence of a continuous, adaptive mechanism that allows the agents to upgrade their activity thresholds when facing a sudden increase in the availability of work further reduces the performance of this algorithm in such dynamic conditions.

PuFT – This algorithm can quickly deal with a dynamic change in the number of objects to manipulate by exploiting its peer-to-peer communication scheme (see results of PuFT in Arena3 shown in Table 3). However, because the threshold is fixed, the PuFT algorithm does not allow the agents to respond efficiently to all modifications of the arena surface (see results of PuFT in Arena2 shown in Table 3).

PuVT – Combining the active threshold from the calibration mechanism and the ability provided by the local communication scheme to the workers to quickly access the information about dynamical changes, this algorithm performs best in all three environments.

Without Worker Allocation – This algorithm performs better than the fixed-threshold algorithms in Arena2. This is explained by the fact that, while the robots with the fixed-threshold algorithms withdrew too soon and left the task unfinished, the robots without worker allocation continued performing the task and always created a single cluster. Although the average size of this cluster is decreased by the unnecessary activity of the robots in the second phase of the aggregation, the cost for an "abandoned task" is higher (due to our choice of the parameters of the cost function) than that of just not constructing a cluster of the largest size possible.

7.2.2 Performance analysis using the macroscopic model

In this subsection, we show that the model presented in Section 6 can easily capture the features of Arena2 and Arena3 and, in turn, help to formulate quantitative predictions about why one type of controller works better than another in a certain environment. A parametric optimization of these slightly more sophisticated worker allocation algorithms based on models is left for future work.

Arena2 – This arena is five times as large as Arena1 and therefore, for a same cluster distribution, it takes, on average, more time to find a seed to manipulate than in Arena1. According to the proposed model, whose parameter calculation is based on geometrical considerations (see Subsection 3.3), this results in the probabilities ($p_{c_n}^{dec}$ and $p_{c_n}^{inc}$) with which the robots encounter a cluster of any given size *n* in Arena2 being one fifth of the probabilities with which they encounter that cluster in Arena1.

When p_c^{dec} and p_c^{dec} are small compared with unity, Equations (36) and (37) can be rewritten as follows

$$\Gamma_f(k;T_s) = e^{\ln\left[\prod_{j=k-T_s}^{k}\left[1-p_c^{dec}(j)\right]\right]} = e^{\sum_{j=k-T_s}^{k}\ln\left(1-p_c^{dec}(j)\right)} \approx e^{-\sum_{j=k-T_s}^{k}p_c^{dec}(j)} \quad (44)$$

$$\Gamma_l(k;T_s) = e^{\ln\left[\prod_{j=k-T_{ds}}^{k-T_d}\left[1-p_c^{inc}(j)\right]\right]} = e^{\sum_{j=k-T_{ds}}^{k-T_d}\ln\left(1-p_c^{inc}(j)\right)} \approx e^{-\sum_{j=k-T_{ds}}^{k-T_d}p_c^{inc}(j)} \quad (45)$$

Hence, according to Equations (44) and (45), the values of Γ_f and Γ_l in both arenas (let us name them Γ_{f1}, Γ_{l1} in Arena1 and Γ_{f2}, Γ_{l2} in Arena2, respectively) are now linked by the following relationships:

$$\Gamma_{f2}(k;T_s) \cong [\Gamma_{f1}(k;T_s)]^{\frac{1}{5}} \quad (46)$$

$$\Gamma_{l2}(k;T_s) \cong [\Gamma_{l1}(k;T_s)]^{\frac{1}{5}} \quad (47)$$

This corresponds, in particular, to a considerable increase in the fractions of free and loaded robots characterized by a fixed activity threshold (i.e., with PrFT and PuFT algorithms) abandoning the aggregation task in Arena2 as compared with Arena1. This outcome is not observed with teams characterized by variable threshold (i.e., PrVT and PuVT) because, intuitively, we can foresee that the mean of the resulting threshold distribution in those teams will be higher, thus compensating the increased number of free and loaded robots willing to abandon the working zone in Arena2.

Arena3 – Introducing 5 additional seeds at an (external) perturbation time τ_e = 2 hours suddenly increases the availability of work (i.e., seeds to manipulate) at time step T_e (the number of iterations corresponding to τ_e). However, the information about this increase is more rapidly disseminated into the network of agents using the public demand estimation mechanism (i.e., PuFT and PuVT) than in the networks of agents using the private demand estimation one (i.e., PrFT and PrVT). This can be explained by the fact that information dissemination is faster in the robot network with an explicit communication scheme than with the stigmergic, implicit communication scheme. Mathematically, at time step T_e, the probabilities of seed manipulation are suddenly increased. This results in a sudden decrease of the fractions of free and loaded robots abandoning the aggregation task. For instance, $p_c^{dec}(k = T_e)$ in Arena3 is equal to that in Arena1 plus five times $p_{c_1}^{dec}$, where $p_{c_1}^{dec}$ is the probability with which an isolated seed (i.e., a cluster of size 1) is found and picked up when present in the arena as previously defined. Equation (48) expresses that the fraction of free agents abandoning the task is reduced by a factor $e^{-5T_s p_{c_1}^{dec}}$. This effect will remain until the newly introduced seeds are all gathered with the original seeds into a single cluster.

(48) $\Gamma_{f3}(T_e;T_s) \cong \Gamma_{f1}(T_e;T_s) e^{-5T_s p_{c_1}^{dec}}$

7.3 Modeling as an optimization tool

The modeling methodology we presented in this paper can be useful also for optimization purposes. First, the model presented here delivers results in time lapses that are at least four orders of magnitude shorter than a corresponding embodied simulation. Second, abstraction in general allows researchers to understand the role of key system parameters, to generalize, and to analyze underlying principles. In particular, we think that the macroscopic model may be improved further so as to predict the optimal threshold distribution among the teammates based solely on the experimental setup. Nevertheless, as shown in [30] with the steady-state analysis of the stick pulling experiment, even if macroscopic models can provide quantitatively correct predictions, nonlinearities and the complexity of these systems can prevent us from going further in analysis (and, indirectly, in optimization) with the mathematical tools currently available.

Although to date we have not investigated whether or not an optimal T_s can be derived analytically (perhaps with the help of a simplified model as we did in [29,30] for the stick-pulling experiment), we used the full macroscopic model described in Section 6 to systematically search for the optimal activity threshold. As also shown in [29], the advantage of using the full model is that, although the complexity of the system may

render the model analytically intractable, the optimization results obtained using the model are still quantitatively correct. The results are presented in Figure 13.

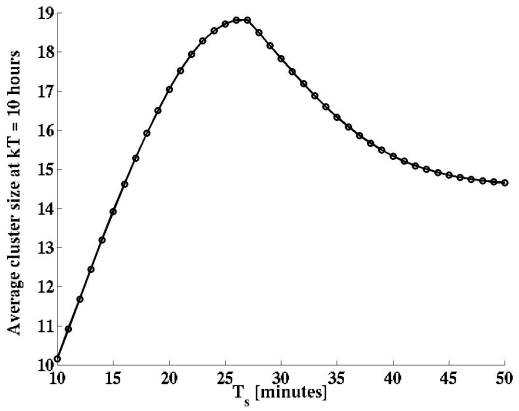

Figure 13: macroscopic model's prediction of the average cluster size at time kT = 10 hours as a function of the activity threshold T_s. Experiments conducted in an 80 x 80 cm^2 arena with a group of 10 robots and T_s varying from 10 to 50 minutes with a 60-second incremental step.

Figure 13 presents the average cluster size obtained after 10 hours of aggregation for different values of the activity threshold T_s. This figure illustrates that for a group of 10 robots, an activity threshold value of about 27 minutes provides the optimal cluster size at time kT = 10 hours. On the one hand, when the threshold value is lower than 27 minutes, the workers stop performing the task too early, before all the seeds have been gathered in a single cluster. On the other hand, when the threshold value is larger than 27 minutes, the robots keep working even after a single cluster has been created and, as a consequence, after 10 hours some seeds are still being carried around by some active agents, decreasing the efficiency of the team.

Although the objects manipulated (fixed stick vs. transportable seeds), the metric used for assessing the swarm performance (average collaboration rate vs. average cluster size), and the significance of the threshold (a time-out regulating how long a robot waits for help vs. a time-out regulating the whole robot activity), the stick-pulling experiment [16,28,29,30] and the aggregation experiment with worker allocation share a few important similarities in their dynamics which are outlined, in particular, at the macroscopic model level. Some of them have been pointed out in Section 6 but we believe that the similarities between the profile of Figure 13 and, for instance, Figure 7, left in [30] are striking. Describing the two experiments at a high level, we can say that in both of them there is a dynamical process taking place in which a specific robot's control parameter characterizing a threshold-based mechanism plays a crucial role. We can say that in both experiments the optimization goal is to find the right threshold (or the right distribution of thresholds) so that the robotic dynamics of the system can be optimally coupled with the dynamics of the task accomplishment, which in turn is constrained by the robot and environmental features. In both cases, the axis defined by all possible

threshold values presents an optimum and, therefore, the model-based optimization procedure for both experiments is also similar.

7.4 Calibration of models' parameters and prediction accuracy

As seen in Subsection 3.3, the models' parameters characterize the microscopic robot-robot and robot-environment interactions. They include probabilities of encountering specific objects (sometimes even considering a preferred angle of approach) and delays required by specific maneuvers whose details are irrelevant for the metric considered. This implies, for instance, an effort to mathematically describe groups of noisy, perhaps different and unevenly-spaced sensors as well as their relative detection and reactive control algorithms with an average detection area or capture an obstacle avoidance maneuver with a mean duration.

In this paper, we used a method similar to that proposed by Lerman and Galstyan in a recent paper [23] as described in Subsection 3.3.2. With this method the rate at which a given object is encountered by a robot is given by the fraction of the total area swept out by the robots' sensors per time step. This method contrasts with that adopted in previous publications [16,25,26,28,29,30], which required an explicit link between time-partitioning and area-partitioning, based on the smallest object present on the arena. While all these heuristic methods are certainly good attempts in the right direction, none of them has, thus far, taken into account the fact that model parameters, measured with systematic tests at lower levels of implementation, may also be characterized by measurement errors and should, therefore, be introduced in the models with a more mathematically accurate approach. For instance, each parameter should be represented as a distribution with a mean and a standard deviation instead of just a single average value. This is an interesting hypothesis, and one we plan to investigate in the near future, although the complexity of the models and the propagation of errors in nonlinear, time-delayed systems will not be trivial.

Further difficulties may arise because of the behavioral granularity captured in the model. The robot's controller used in the aggregation case study, developed prior to the models presented here, can be approximated as a FSM, though certain routines (obstacle avoidance and interference) have been implemented with proximal controllers. Proximal controllers, in our case neural network-based controllers, tightly couple actuators with sensors without passing through a distal representation as, for instance, is the case for behavior-based implementations. Parameters used to describe the states corresponding to such routines (in our case, the duration of obstacle avoidance and interference as well as the probability of detecting an obstacle and a teammate) can still be measured in systematic tests with one or two robots, as mentioned above, even if this implies some inaccuracy. For predicting "high-level" metrics such as the average cluster size, the average number of clusters, or the average number of active workers in the arena, this approximation is quite sufficient. However, as shown in Section 4, when we attempted to predict the values of state variables in steady-state regime (a "low-level" metric) using the models, the inaccuracies became more noticeable. The description of a state as proposed in this paper is therefore inadequate for such controllers; temporal attractors in the state space rather than static state definitions may achieve better results.

7.5 The limitations of the modeling methodology

The macroscopic modeling method proposed in this paper is nonspatial as it does not make use of the trajectories of the agents, the correlation between the positions occupied over consecutive time steps, or the spatial distribution of the agents resulting from their movement pattern and the environmental configuration. Therefore, this model cannot accurately describe swarm-based systems that are sensitive to the trajectories of the agents or their positions over time. An example of such a system is a team of rovers carrying out a coverage task in a planetary exploration mission. In the case of remote operations, the lifetime of each robot is crucial and sets a maximal boundary on the time within which the mission must be accomplished. In this case, the current model would have to be extended to take into account states defined by the position (and orientation) of the robot in order to be quantitatively accurate. Depending on the metric chosen, spatial distributions rather than detailed trajectory information may suffice to achieve this goal.

The current calculation of the model's encounter rates and probabilities is based on geometrical considerations and relies on the assumption that there are no overlapping detection areas between the objects (e.g., walls, seeds, and robots). A soon as this assumption no longer holds, such as in an overcrowded scenario, the current modeling method reaches its limitations and the predictions are no longer quantitatively correct (see also [28,30].

The proposed model is based on a PFSM description of the system, thus, it assumes that the system is Markovian. While this approach works very well when describing a system using a behavior-based controller, it can hardly be used to represent less "state-based" controllers such as proximal controllers, as mentioned in the previous subsection, or (continuously) adaptive controllers. By analogy, the states of a PFSM description can be thought of as variables that take discrete values over a given interval whereas the states in the proximal systems can be thought of as variables that take real, continuous values over the same interval.

Finally, in the mathematical description of systems consisting of discrete numbers of agents, numerical effects can emerge on the predictions. For instance, in the calculation of the size and the number of clusters, using real numbers can result in obtaining cluster sizes or numbers of clusters smaller than unity but still greater than zero. Similar problems emerge in population dynamics, for instance, when using mathematical models to describe propagation of an infectious disease in animal populations (see [40]): the number of agents in a given state often falls below unity without being explicitly zero. These numerical effects may reduce the meaningfulness of the model's predictions, in particular with small swarm sizes.

8 Conclusion and Outlook

In this paper, we have presented a series of macroscopic models incrementally implemented to study aggregation experiments using embodied agents in groups of fixed or variable sizes. A heuristic but precise calibration procedure based on systematic experiments with one or two embodied agents has helped us avoid the use of free parameters in the models. We have defined a clear mapping between the parameters of the model and the geometrical characteristics of the environment. We have validated the

predictions of the macroscopic model with a realistic, sensor-based, embodied simulator. Results show that the proposed approach delivers quantitatively accurate predictions, in particular for nonspatial metrics related to the aggregation process, and constitutes a computationally efficient tool.

Moreover, we have extended the model to capture aggregation experiments performed with robotic teams of time-varying sizes. To this purpose, we used a scalable, distributed, threshold-based worker allocation algorithm that allows a team of autonomous robots to dynamically allocate an appropriate number of workers to a given task based solely on their individual estimations of the progress in the execution of the task. We have shown that teams of workers dynamically controlled by the allocation algorithm achieve similar or better performance in aggregation than those characterized by a constant team size, while using on average, a considerably reduced number of agents over the whole aggregation process. The macroscopic model has also helped us shed light on performance differences observed when using a suite of threshold-based algorithms in different scenarios.

The simplicity of the modeling methodology suggests that it is easily applicable to other aggregation/segregation or sorting experiments characterized by different agent capabilities and individual control algorithms.

Future work will involve an effort to investigate different calibration procedures, the propagation of parametric errors in the models, and the extension of the methodology to more complex controllers endowed with learning and/or more sophisticated navigation and communication capabilities. Models could also be of a great help in devising more powerful worker and task allocation algorithms, either threshold-based, as proposed in this paper and by other authors [10,21,33], or market-based [12].

Acknowledgements

The authors wish to thank Kristina Lerman and Aram Galstyan for having stimulated the discussion and fundamentally contributed to the macroscopic modeling of multi-agent systems.

References

1. Agassounon, W., Martinoli, A., and Goodman, R. M. 2001. A Scalable, Distributed Algorithm for Allocating Workers in Embedded Systems. In *Proc. IEEE Int. Conf. on System, Man and Cybernetics*, Tucson, AR, pp. 3367-3373.
2. Agassounon, W. and Martinoli, A. 2002. Efficiency and Robustness of Threshold-Based Distributed Allocation Algorithms in Multi-Agent Systems. In *Proc. First Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, pp.1090-1097.
3. Agassounon, W. and Martinoli A. 2002. A Macroscopic Model of an Aggregation Experiment using Embodied Agents in Groups of Time-Varying Sizes. In *Proc. IEEE Int. Conf. on System, Man and Cybernetics*, Hammamet, Tunisia.
4. Beckers R., Holland O. E., and Deneubourg J.-L. 1994. From Local Actions to Global Tasks: Stigmergy and Collective Robotics. In Brooks R. and Maes P., editors, *Proc. Fourth Workshop on Artificial Life*, Boston, MA, pp. 181-189.
5. Beshers S. N. and Fewell J. H. 2001. Models of Division of Labor in Social Insects. *Annual Review of Entomology* 46: 413-440.
6. Beni G. and Wang J. 1989. Swarm Intelligence. In *Proc. Seventh Annual Meeting of the Robotics Society of Japan*, Tokyo, Japan, pp. 425-428.

7. Bonabeau E., Theraulaz G., and Deneubourg J.-L. 1998. Fixed Response Thresholds and Regulation of Division of Labour in Insect Societies. *Bulletin of Mathematical Biology* 60: 753-807.
8. Bonabeau E., Dorigo M., and Theraulaz G. 1999. *Swarm Intelligence: From Natural to Artificial Systems.* SFI Studies in the Science of Complexity, Oxford University Press: New York, NY.
9. Camazine S., Deneubourg J.-L., Franks J., Sneyd E., Bonabeau E., and Theraulaz G. 2001. *Self-Organization in Biological Systems*, Princeton University Press: Princeton, NJ.
10. Cicirello V. A. and Smith S. F. 2004. Wasp-like Agents for Distributed Factory Coordination. *J. of Autonomous Agents and Multi-Agent Systems*, 8(3): 237-266.
11. Chrétien, L. 1996. Organisation spatiale du matériel provenant de l'excavation du nid chez Messor Marbarus et des cadavres chez Lasius Niger (Hymenopterea: Formicidea). Ph.D. Dissertation, Université Libre de Bruxelles, Belgium.
12. Gerkey B. P. and Matarić M. J. 2002. Sold!: Auction Methods for Multirobot Coordination. Special Issue on Advances in Multi-Robot Systems, Arai T., Pagello E., and Parker L. E., editors, *IEEE Trans. on Robotics and Automation*, 18(5): 758-768.
13. Gordon D. M. 1999. *Ants at Work: How an Insect Society is Organized*. The Free Press, Simon & Schuster Inc.: New York, NY.
14. Hayes A. T., Martinoli A., and Goodman R. M. 2002. Distributed Odor Source Localization. Special Issue on Artificial Olfaction, Nagle H. T., Gardner J. W., and Persaud K., editors, *IEEE Sensors*, 2(3): 260-271.
15. Holland O.E. and Melhuish C. 1999. Stigmergy, Self-Organisation, and Sorting in Collective Robotics. *Artificial Life*, 5: 173-202.
16. Ijspeert A. J., Martinoli A., Billard A., and Gambardella L.M. 2001 Collaboration through the Exploitation of Local Interactions in Autonomous Collective Robotics: The Stick Pulling Experiment. *Autonomous Robots*, 11(2): 149-171.
17. Kazadi S., Abdul-Khaliq A., and Goodman R. M. 2002. On the Convergence of Puck Clustering Systems. *Robotics and Autonomous Systems*, 38(2): 93-117.
18. Krause J. and Ruxton G. D. 2002. Living in Group. Oxford University Press: New York, NY.
19. 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.
20. Kube C. R. and Bonabeau E. 2000. Cooperative Transport by Ants and Robots. *Robotics and Autonomous Systems*, 30(1-2): 85-101.
21. Labella T. H., Dorigo M., and Deneubourg J.-L. 2004. Efficiency and Task Allocation in Prey Retrieval. In Ijspeert A. J. and Murata M., editors, *Proc. First Int. Workshop on Biologically Inspired Approaches to Information Technology*, Lausanne, Switzerland, Lecture Notes in Computer Sciences, pp. 32-47.
22. Lerman K., Galstyan A., Martinoli A., and Ijspeert A. J. 2001. A Macroscopic Analytical Model of Collaboration in Distributed Robotic Systems. *Artificial Life*, 7(4): 375-393.
23. Lerman K. and Galstyan A. 2002. Mathematical Model of Foraging in a Group of Robots: Effect of Interference. *Autonomous Robots*, 13: 127-141.
24. Martinoli A. and Mondada F. 1995. Collective and Cooperative Group Behaviours: Biologically Inspired Experiments in Robotics. In Khatib O. and Salisbury J. K., editors, *Proc. Fourth Int. Symp. on Experimental Robotics*, Stanford, CA, Lecture Notes in Control and Information Sciences, pp. 3-10.
25. Martinoli A., Ijspeert A. J., and Mondada F. 1999. Understanding Collective Aggregation Mechanisms: From Probabilistic Modelling to Experiments with Real Robots. *Robotics and Autonomous Systems*, 29: 51-63.
26. Martinoli A., Ijspeert A. J., and Gambardella L. M. 1999. A Probabilistic Model for Understanding and Comparing Collective Aggregation Mechanisms. In Floreano D., Mondada F., and Nicoud J.-D., editors, *Proc. Fifth European Conf. on Artificial Life*, Lausanne, Switzerland, Lectures Notes in Computer Science, pp. 575-584.
27. Martinoli A. 1999. Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Control Strategies. Unpublished doctoral manuscript, EPFL Ph.D. Thesis Nr. 2069, Lausanne, Switzerland. Downloadable at: http://www.coro/caltech/edu/people/alcherio/am_pub.
28. Martinoli A. and Easton K. 2003. Modeling Swarm Robotic Systems. In Siciliano B. and Dario P., editors, *Proc. Eighth Int. Symp. on Experimental Robotics*, Sant'Angelo d'Ischia, Italy. Springer Tracts in Advanced Robotics (2003), pp. 285-294.

29. Martinoli A. and Easton K. 2003. Optimization of Swarm Robotic Systems via Macroscopic Models. In Schultz A., Parker L., and Schneider F., editors, *Proc. of the Multi-Robot Systems Workshop*, Naval Research Laboratory, Washington, DC, pp. 181-192.
30. Martinoli A., Easton K, and Agassounon W. 2004. Modeling Swarm Robotic Systems: A case Study in Collaborative Distributed Manipulation, *Int. Journal. of Robotic Research,* 23(4): 415-436.
31. Michel O. 1998. Webots: Symbiosis Between Virtual and Real Mobile Robots. In Heuding J.-C., editor, *Proc. First Int. Conf. on Virtual Worlds*, Paris, France, pp. 254-263. See also http://www.cyberbotics.com/webots/.
32. Mondada F., Franzi E., and Ienne P. 1993. Mobile Robot Miniaturization: A Tool for Investigation in Control Algorithms. In Yoshikawa T. and Miyazaki F., editors, *Proc. Third Int. Symp. on Experimental Robotics*, Kyoto, Japan, Lecture Notes in Control and Information Sciences, pp. 501-513.
33. Nouyan S. 2002. Agent-Based Approach to Dynamic Task Allocation. In Dorigo M., Di Caro G., and Samples M., *Proc. Third Int. Workshop on Ant Algorithms*, Brussels, Belgium, Lecture Notes in Computer Sciences, pp. 28-39.
34. Pacala S. W., Gordon D. M., and Godfray H. C. J. 1996. Effects of Social Group Size on Information Transfer and Task Allocation, *Evolutionary Ecology*, 10: 127-165.
35. Parrish J. K. and Hamner W. M. 1997, editors. *Animal Groups in Three Dimensions*. Cambridge University Press: Cambridge, UK.
36. Sugawara K. and Sano M. 1997. Cooperative Accelleration of Task Performance: Foraging Behavior of Interacting Multi-Robots System. *Physica D*, 100: 343-354.
37. Sugawara K., Sano M., Yoshihara I., and Abe K. 1998. Cooperative Behavior of Interacting Robots. *Artificial Life and Robotics*, 2: 62-67.
38. Theraulaz G., Bonabeau E., and Deneubourg J.-L. 1998. Response Threshold Reinforcement and Division of Labour in Insect Societies. *Proc. Royal Society of London*, Series B, 265: 327-332.
39. Wilson E. O. 1984. The Relation Between Caste Ratios and Division of Labour in Ant Genus Pheidole (Hymenoptera: Formidacea). *Behav. Ecol. Sociobiol.*, 16: 89-98.
40. Mollison D. 1977. Spatial Contact Models for Ecological and Epidemic Spread. *Royal Statist. Soc*., B 39:283-326.