Efficiency and Robustness of Threshold-Based Distributed Allocation Algorithms in Multi-Agent Systems

William Agassounon
Collective Robotics Group
California Institute of Technology
Pasadena, CA 91125, USA
+1 (626) 395 2243
agassw@micro.caltech.edu

Alcherio Martinoli
Collective Robotics Group
California Institute of Technology
Pasadena, CA 91125, USA
+1 (626) 395 2208
alcherio@micro.caltech.edu

ABSTRACT

In this paper we present three scalable, fully distributed, threshold-based algorithms for allocating autonomous embodied workers to a given task whose demand evolves dynamically over time. Individuals estimate the availability of work based solely on local perceptions. The differences among the algorithms lie in the threshold distribution among teammates (homogeneous or heterogeneous team), in the mechanism used for establishing threshold values (fixed, parameter-based or variable, rule-based), and in the sharing (public) or not sharing (private) of demand estimations through local peer-to-peer communication. We tested the algorithms' efficiency and robustness in a collective manipulation case study concerned with the clustering of initially scattered small objects. The aggregation experiment has been studied at two different experimental levels using a microscopic model and embodied simulations. Results show that teams using a number of active workers dynamically controlled by one of the allocation algorithms achieve similar or better performances 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. While differences in efficiency among the algorithms are small, differences in robustness are much more apparent. Threshold variability and peer-to-peer communication appear to be two key mechanisms for improving worker allocation robustness against environmental perturbations.

Categories and Subject Descriptors

J.2 [**Computer Applications**]: Physical Science and Engineering – *engineering, electronics, mathematics and statistics.*

General Terms

Algorithms, Performance, Reliability, Experimentation.

Keywords

Swarm intelligence, division of labor, response threshold, probabilistic modeling, embodied multi-agent systems.

AAMAS '02, July 15-19, 2002, Bologna, Italy.

1. INTRODUCTION

Swarm Intelligence (SI, first introduced in [2]) is a new computational metaphor for solving distributed problems using the principles guiding the collectively complex and intelligent behavior of natural systems consisting of many agents, such as ant colonies and bird flocks. The abilities of such systems appear to transcend the abilities of each constituent agent. In all the biological cases studied so far, the emergence of high-level control has been found to be mediated by nothing more than a small set of simple low-level interactions among individuals, and between individuals and the environment [5,7].

Generally speaking, our research focuses on the application of the SI approach to control embedded systems that consist of many autonomous decision making entities endowed with local perception and maybe communication capabilities. In particular, 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. One of the most appealing principles of the SI approach to a roboticist is the minimalism [3] at the individual agent's level. This characteristic prompts the roboticist to carefully evaluate each additional capability the single agent should be endowed with, which in turn should lead to an overall increased system robustness and cost effectiveness in mass production.

Recently, several macroscopic models, some of them based on threshold responses [4,16], others focusing only on task-switching probabilities [14], have been proposed to explain these mechanisms in natural colonies. However, none of these theoretical approaches has focused on 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).

In the collective robotic literature, we find threshold-based [9,15], market-based, and publish/subscribe messaging approaches [6] that take into account the embodiment of the agents but which are not scalable because of extensive communication requirements in a finite bandwidth or the necessity of an external supervisor. For instance, in the pioneering approach proposed by Parker [15] each robot at every instant of time and in every position is aware of the progress in task accomplishment of its teammates based on a global radio networking and an absolute positioning system. In Krieger and Billeter's experiment [9] the demand related to the nest energy is assessed by an external supervisor and globally transmitted to all the robots. Using this method, the team of robots has to be heterogeneous and each agent has to be characterized by a different threshold in order to regulate the activity of the team. This in turn results in a different exploitation of the teammates, the one endowed with the lowest threshold systematically being more active than that with the highest one.

In [1] we proposed a threshold-based, distributed, scalable worker allocation algorithm that is based exclusively on the local estimation of the demand by the individuals. The individuals were all characterized by the same threshold but since the agents did not perceive the demand globally but rather estimate it locally, they did not work or rest all at the same time, a behavior that would have arisen if the demand was broadcasted from an external supervisor. In this paper we propose two new algorithms of the same family and we compare their performances in efficiency and robustness with the one previously presented. Consistently with the SI approach, the two newly proposed algorithms slightly extend the individual capabilities in order to overcome some of the limitations presented by the first algorithm, in particular in case of environmental perturbations. The first new algorithm endows each agent with the ability to calibrate its own response threshold before starting to adapt its activity based on this threshold. We therefore replace here an *a priori* fixed parameter by a rule that can adapt the value of this parameter according to some locally sensed environmental constraints. The second new algorithm allows the team of embodied agents to exchange individual estimations of the demand through local peer-to-peer communication while still relying on an a priori fixed threshold. Since the local estimation of the demand is noisy, sharing this information among the teammates is a way to increase the update rate of this estimation and reduce the corresponding error without recurring to an external supervisor.

Collective embedded systems can be studied at several implementation levels, from macroscopic analytical models [1] to units in real world [10,11] through microscopic, numerical models and embodied simulations. Models allow for a better understanding of the experiment dynamics and for a generalization to other tasks, environmental constraints, and embedded platforms. When using quantitatively accurate models, optimal parameters of the control algorithms can be investigated much more quickly at more abstract levels and the effectiveness of the devised solution can then be verified using embodied simulations and/or real embedded systems. In this paper we present results gathered at two implementations levels using a microscopic model and embodied simulations. Both levels are well suited for studying noisy demand estimations and heterogeneous teams of agents without having to deal in this exploratory phase with all sort of problems arising in real robots experiments. The qualitative and quantitative reliability of both implementation levels for this type of experiments have been shown in previous work [1,8,10,11].

2. EXPERIMENTAL SETUP

2.1 The Aggregation Experiment

The case study used for assessing the efficiency of the worker allocation algorithm is concerned with the gathering and clustering of small objects scattered in an enclosed arena. In most of the work done so far [7], and more specifically [10,11], the size of the working team was kept constant during the whole aggregation process. These experiments define our baseline for an efficiency comparison with and without 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. We then integrate all three primary team performances in a combined metric that represents the cost of aggregation over a certain time period.

2.2 The Embodied Simulator

We implemented the aggregation experiment in Webots 2.01, a 3D sensor-based, kinematic simulator [12] of Khepera robots [13]. The simulator computes trajectories and sensory inputs of the embodied agents in an arena corresponding to the physical setup (see Figure 1).

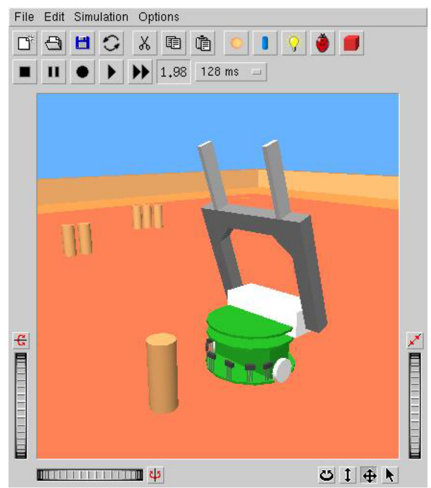

Figure 1 a. Close up of a simulated robot (5.5 cm in diameter) in Webots equipped with a gripper turret in front of a seed.

The mean comparative speed ratio for this experiment with 10 robots between Webots and real time is about 7 on a PC Pentium III 800 MHz workstation. The environment is represented by two square arenas of different sizes. Each of them contains a working zone of a corresponding size (80x80 cm and 178X178 cm) where twenty small seeds are randomly scattered at the beginning of the experiment. A resting zone surrounds the working zone, where non-active agents go to rest (or stay in an idle state to save energy). Agents are endowed with sensor capabilities to distinguish the border between resting and working zones. Without considering the mode-switching behavior (explained in section 3), we can summarize each agent's behavior with the following simple rules. In its default behavior the agent moves straight forwards within the working zone looking for seeds. When at least one of its six frontal proximity sensors is activated, the agent starts a discriminating procedure. Two cases can occur: if the agent is in front of a large object (a wall, another agent, or the body side of a cluster of seeds), the object is considered as an obstacle and the agent avoids it. In the second case, the small object is considered as 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 that 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. Note that, because agents identify only the two extreme seeds of a cluster as seeds (as opposed to obstacles), clusters are built in line.

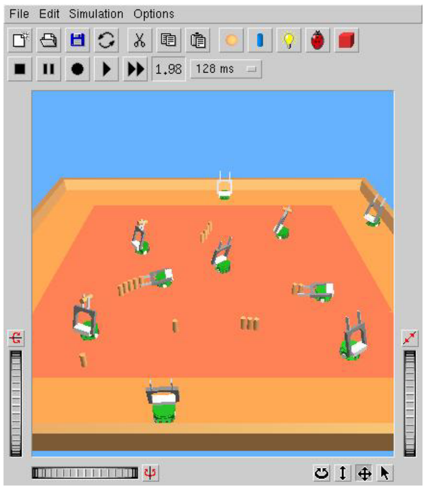

Figure 1 b. Experimental setup: inner area corresponds to the working zone, and outer area is the resting zone. Aggregation in progress with 10 agents in a 178X178 cm arena.

As shown in [11], if the agents do not drop a seed unless it is next to another seed or pick up an internal seed of a cluster, the number of clusters is monotonically decreasing and eventually a single cluster always arises.

2.3 The Microscopic, Probabilistic Model

The central idea of the microscopic, probabilistic model is to describe the experiment as a series of stochastic events with probabilities based on simple geometrical considerations and systematic interaction experiments with a single real or embodied agent. The probability for any agent to encounter any other object present in the arena (e.g. a seed, a teammate, the border between the working field and the resting zone, etc.) is given by the ratio of the extended area occupied by that object to the total arena area in which the agent is moving. The extended area occupied by each object is computed by considering the detection range of that object by an active agent taken from the center of that agent. In this specific collective manipulation case study, seed picking up and dropping probabilities have also to be taken into account once a cluster is found and they depend on the angle of approach of the agent to the cluster (clusters can be modified only at their tips).

In the numerical, probabilistic model a finite state machine defines the states of the agents, but instead of computing the detailed sensory information and trajectories of the agents the change of states is determined randomly by rolling a dice. The overall behavior is then computed by averaging the results of several runs of the same experiment. A more detailed description of this microscopic modeling methodology can be found in [8,10,11].

Working with models also brings an additional time saving in comparison to embodied simulations. The mean speed ratio for this experiment with 10 agents between the microscopic probabilistic model and Webots is about 700 on a PC Pentium III 800 MHz workstation.

3. THE DISTRIBUTED WORKER ALLOCATION MECHANISM

The main objective of this case study is to show that the introduction of worker allocation mechanisms allows the team of agents to increase its efficiency as a whole by allocating the right number of workers as a function of the demand intrinsically defined by the aggregation process. Intuitively, we can imagine that at the beginning of the aggregation there are several possible manipulation sites (i.e. several scattered seeds) that allow for a parallel work of several agents. As the aggregation process goes on, the number of these sites is reduced and having more agents competing for the same manipulation sites decreases their efficiency.

In threshold-based systems, the 'propensity' of any agent to act is given 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 all the algorithms presented in this paper the time an agent spends before finding some work to accomplish (i.e. to pick and drop a seed) represents the agent's estimation of the demand stimulus associated with the aggregation task.

Our current worker allocation mechanism is as follows. When an agent 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 goes below a certain threshold (i.e. if the amount of time spent in the search for work to accomplish is above a given T_{search} time-out), a deterministic switching mechanism prompts the agent to leave the working zone for resting in the adjacent parking space. An agent carrying a seed 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 agent is able to estimate the aggregation demand locally and to decide whether to work or rest.

In the following, 'public' refers to the existence of an explicit, collaborative information flow between the teammates, and 'private', to no information sharing at all. Thus, in what follows we present three different worker allocation algorithms: a private, fixed-threshold algorithm, a private, variable-threshold algorithm, and a public, fixed-threshold algorithm.

3.1 The Private, Fixed-Threshold Worker Allocation Algorithm (PrFT)

The PrFT algorithm is the same reported in [1]. Following the worker allocation mechanism described previously, we assign the same response threshold to all the agents. The team of agents is therefore homogeneous from control point of view. The resulting agents' behavior (rhythm of activity) is not identical since it is based on the local, private assessment of the current status of the shared resource, i.e. the environment. In other words, diversity in activity is created by exploiting the intrinsic noise of the system as well as local perceptions and interactions.

3.2 The Private, Variable-Threshold Worker Allocation Algorithm (PrVT)

Using fixed-threshold algorithms does not endow the team of workers with the robustness required to face external perturbations brought to the system. For instance, if some key control parameters such as the response thresholds have to assume different values as a function of the characteristics of the environment (e.g. arena size) for an optimal team performance, an algorithm is required that automatically calculates the optimal parameters for different environmental constraints.

In this paper, we propose a variable-threshold algorithm based on a threshold self-calibration rule that works in two steps: a *threshold estimation* phase followed by a *worker allocation* phase. During the estimation phase, each autonomous agent evaluates the spatial density of the demand and then sets its response threshold based on that individual estimation. During the allocation phase, the algorithm works as explained above. Two parameters govern the self-calibration mechanism: the *Estimation Steps* (ES) and the *Estimation Factor* (EF). Each autonomous agent estimates the availability of work in the environment by averaging the amount of time it spends to find some work to accomplish over its first ES successful attempts. The agent then computes its response threshold by multiplying that average amount of time by EF. Notice that even if all the agents are characterized by the same values of ES and EF and therefore the same calibration rule (homogeneous team), due to their partial perceptions in time and space the agents will end up with different thresholds in the allocation phase. In addition, since PrVT is a private algorithm, the transition time from one phase to the other is also determined by the agents individually and asynchronously. Equation 1 summarizes how each autonomous agent computes its response threshold. T_W^k represents the amount of time an agent spent to find work to accomplish at its k^{th} successful attempt.

$$T_{search} = EF * \left(\frac{1}{ES} \sum_{k=1}^{ES} T_{W^k} \right) \qquad (1)$$

3.3 The Public, Fixed-Threshold Worker Allocation Algorithm (PuFT)

In order to allow this multi-agent system to react to dynamic external perturbations (e.g. sudden introduction of additional seeds) more quickly than relying on the slow implicit communication through environmental modifications, we endow all the agents with peer-to-peer communication capabilities. An additional advantage of explicit communication is that each agent is able to gather information about the work demand both from its individual experience and the experience of other teammates that it may encounter. Our current collaborative scheme is as follows. When two agents are within a certain *Communication Range* (CR), they exchange their estimation of the demand; their individual estimations are then set to the average of their original values. In the PuFT algorithm presented in this paper, the distribution of CR and T_{search} is homogeneous and fixed *a priori* among the teammates.

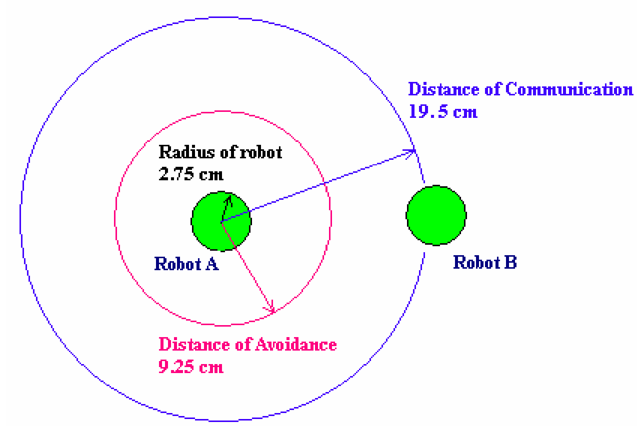

Figure 2. Example of two agents within communication range.

4. RESULTS AND DISCUSSION

In this section we present and compare results collected at two different experimental levels using microscopic modeling and embodied simulations. Unless otherwise stated, each aggregation run lasted 10 hours (simulated time), for each team size 100 and 30 runs were carried out using the microscopic model and the embodied simulator respectively, and error bars represent the standard deviations among runs. All results reported below were obtained without using any free parameters. All the parameters introduced in the model (e.g. mean obstacle avoidance duration, mean time to pick up/drop a seed, mean time to leave the working zone) were measured from a single embodied agent.

4.1 Aggregation Without Worker Allocation

Figure 3 presents the model predictions and the embodied simulation results of the aggregation experiment without the use of any worker allocation algorithm using a group of 10 agents in an 80X80 cm arena. In that plot the upper set of curves represents the (increasing) average size of the clusters over time while the other set shows the (decreasing) average number of clusters over time. Figure 3 consists of a first phase when the average cluster size increases steadily from 1 seed to about 15 seeds and a second phase when the average cluster size remains on average constant around 15 seeds. Similarly, during the first phase the average number of clusters decreases asymptotically from 20 to about 1 then remains close to 1 during the second phase of the aggregation process. This can be explained by the fact that, once a single cluster arises only two manipulation sites remain in the environment (i.e. the two end tips of that cluster). The probabilities of picking up and dropping a seed 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. A similar aggregation evolution was recorded when using an arena of size 178X178 cm. The main difference lies in the time needed to reach a single cluster, which takes on average twice as long in the larger arena. Note that the latter arena has five times the surface of the 80X80 cm arena.

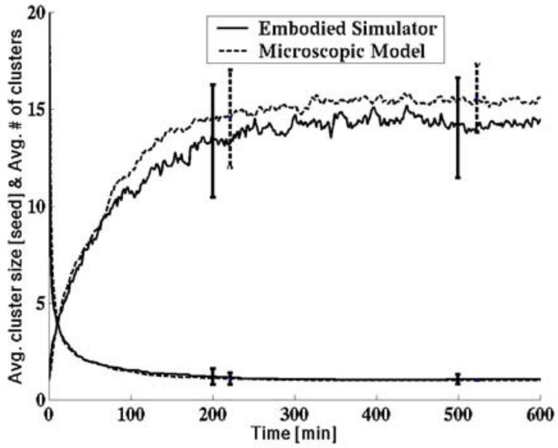

Figure 3. Results of the aggregation experiment without worker allocation using a team of 10 agents.

The good agreement between the results collected at both implementation levels shows how reliable the microscopic model's predictions are. Additional results for the same case study have been reported in [1]. Therefore, in the following, we will be presenting results obtained using the microscopic model exclusively.

4.2 Integrated Cost and Optimization of Algorithmic Parameters

To be able to optimize the different algorithmic parameters and to access the cost effectiveness of these different algorithms, we introduced a cost function whose integrated value over the total observation time, named *Integrated Cost* (IC), corresponds to the total cost of the experiment. The IC represents an efficient combined metric for comparing the influence of each parameter of any given worker allocation algorithm, as well as the performances of different algorithms. The cost function is defined in Equation 2.

$$F_{\cos t}(x_t, y_t, z_t) = \gamma_x (X_{opt} - x_t)^2 + \gamma_y (Y_{opt} - y_t)^2 + \gamma_z (Z_{opt} - z_t)^2 \qquad (2)$$

Where:

- x_t, y_t and z_t represent the average cluster size, the average number of clusters, and the average number of active workers at time step *t* respectively.
- X_{opt}, Y_{opt} and Z_{opt} are the optimal values of the variables above (e.g. 20, 1, and 0 respectively).
- γ_x, γ_y and γ_z are coefficients selected to weight the contribution of each of the variables accordingly.

In the right-hand side of Equation 2, the first two terms can be considered as the 'penalty cost' (e.g. the cost for work not finished) and the third term can be seen as the 'worker cost' (e.g. cost for hiring workers). The *Integrated Cost* is computed by using a discrete *Riemann* integration over any observation time. Results presented in Table 1 were obtained for an observation time of 10 hours. We chose $\gamma_x = 10^{-2}$ and $\gamma_x = \gamma_y = 4\gamma_z$ in order to obtain a higher penalty cost for unsuccessful aggregation results (cluster size and number of clusters) than a worker cost. In Table 1 we optimized PuFT for a fixed communication range of 20 cm.

In this paper, we use the IC for two purposes: first, as a metric to optimize parameters of the worker allocation algorithms; second, as a metric to compare the performances of the worker allocation algorithms under different environmental constraints. Additionally, the following criterion based on central tendencies (mean and standard error) is used in order to assess the quality of a given set of parameters on the IC-axis.

Assume that (E_i, ε_i) represents the pair of mean and standard error of the IC associated with the set of simulation runs *i*. We consider the IC achieved by the set *i* as better (more efficient) than that of the set *j* if $Ei+(1+\eta)\varepsilon_i < Ej-(1+\eta)\varepsilon_j$, where η is a criterion parameter. For the results reported here we used $\eta = 0$. In the case where both sets respect the inequality above, we consider them as equivalent. Considering the acceleration factor of the probabilistic model and the limited number of design points in a given parameter space, we performed a systematic test of all possible parameter combinations for a given algorithm.

Table 1 summarizes the algorithmic parameters and the optimized values for the three worker allocation algorithms in an 80X80 cm arena. Note that in Equation 2 we do not take into account the extra power consumption cost introduced by the communication capability each agent is endowed with in PuFT.

Table 1. Algorithmic Parameters

Algorithm	Para-meter	Range (min-max)	Granularity	Optimal value
PrFT	T_{search}	10-35 min.	5 min.	25 min.
PrVT	ES EF	5 – 40 0.5 - 10	5 0.5	10 8
PuFT	T_{search}	10-35 min.	5 min.	15 min.

4.3 Aggregation with Worker Allocation

In the following, we present a set of comparative results obtained by applying the worker allocation algorithms detailed in section 3 to the same aggregation experiment. We started by optimizing their respective parameters (as explained in section 4.2) for a same experimental setup (e.g. an 80X80 cm arena, 20 seeds originally scattered in the arena, and 10 active agents at the beginning of the experiment). We then proceeded by studying the influence of external perturbations on the cluster formation (e.g. different arena size, 20% additional seeds introduced during the aggregation process) for the different algorithms. Table 2 presents the IC values and their standard errors for the three worker allocation algorithms in the different manipulation experiments introduced previously (values in bold corresponds to the best performance, i.e. minimal cost, calculated by applying the criterion mentioned above). In Table 2, Arena1 corresponds to the original setup (an 80X80 cm arena and 20 seeds). Arena2 corresponds to the case with static perturbation (20 seeds in a new 178X178 cm arena), and Arena3 refers to the case with dynamic perturbations (80X80 cm arena, 20 original seeds and 5 additional seeds introduced during the aggregation process 2 hours after the start).

Table 2. Integrated cost

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**
W/o WA	227.4 ± 4.8	310.8 ± 8.8	197.2 ± 5.9

The PuFT and PrFT algorithms appear to be the most efficient in Arena1 (i.e. equivalent performances following the criterion), PrVT in arena2, and PuFT in Arena3.

4.3.1. Private, Fixed-Threshold Worker Allocation

Figure 4 shows the outcome of the aggregation experiment using the worker allocation algorithms with a team of 10 agents in an 80X80 cm arena. Figure 4 shows that here, conversely to the case without worker allocation, during the last phase of the aggregation, the average cluster size remains an increasing function of time eventually reaching 20 seeds, the optimal largest value possible, while the number of active workers in the environment decreases. Intuitively, this can be explained by the fact that with only two manipulation sites remaining in the arena, and on average half of the active agents always carrying a seed and the other half not, reducing the number of active agents, consequently increases the size of the single cluster.

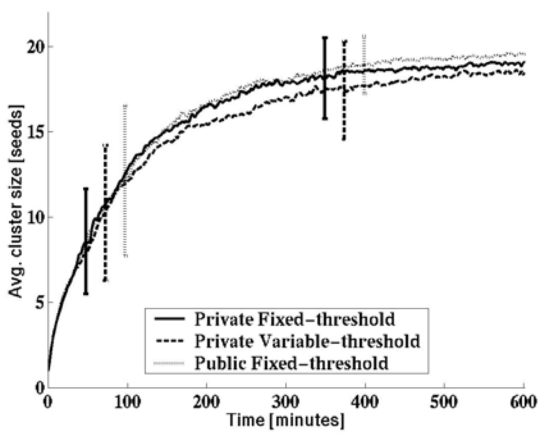

Figure 4 a. Average cluster size for aggregation experiment with worker allocation algorithms in an 80X80 cm arena

However, due to the *a priori* fixed response threshold value the agents behave sub-optimally in a different environment. For instance, when performing the same aggregation task in a 178X178 cm arena, the average size of the clusters they create are smaller on average than the average size of those created by the

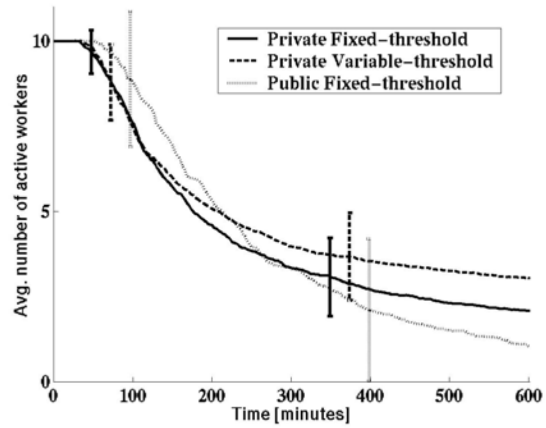

Figure 4 b. Average number of active workers for aggregation experiment with worker allocation algorithms in an 80X80 cm arena

team using the PrVT algorithm (with similar standard deviations) because the agents withdraw too soon. This is illustrated in Figure 5 where after 120 minutes, the size of the clusters created using the PrFT algorithm becomes (and remains for the rest of the experiment) distinctively smaller on average than that of the clusters created using the PrVT algorithm. As a consequence, the aggregation efficiency of the PrFT algorithm deteriorated considerably in Arena2 as shown in Table 2.

4.3.2. Private, Variable-Threshold Worker Allocation

The density of manipulation sites (seeds that can be manipulated) 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 and this in turn allows them to continue performing the task, as most seeds are not gathered yet into a single cluster. This is illustrated by Figure 5 where it clearly appears that PrFT and PuFT under-perform due to a relatively too low homogeneous threshold value and their inability to adapt to a new environment.

However, the PrVT algorithm is not appropriate for an optimal response of the agents to a dynamic change in the number of objects to manipulate. For instance, results in Table 2 show that when additional seeds are dropped in the arena after 2 h into the aggregation process, the efficiency of the PrVT algorithm deteriorates. This results from the nonexistence of a continuous adaptive activity threshold mechanism that allows the agents to upgrade their activity thresholds when facing a sudden increase in the availability of work.

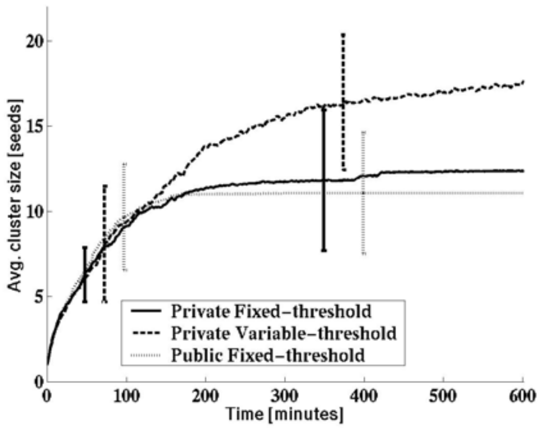

Figure 5. Average cluster size for aggregation experiment with worker allocation algorithms in a 178X178 cm arena

4.3.3. Public, Fixed-Threshold Worker Allocation

PuFT can efficiently deal with a dynamic change in the number of objects to manipulate in a distributed way. However, because the threshold is fixed, the PuFT algorithm does not allow the agents to respond efficiently to all modifications of the arena surface as shown in Figure 5 and Table 2. Nevertheless, this algorithm has the advantage of providing the team of autonomous agents with the ability to quickly access the information about dynamic changes brought to the working environment through its peer-to-peer communication scheme. As shown in Figure 6, the sudden increase in the availability of work (i.e. 20% additional seeds dropped in the environment) was quickly sensed by the teammates. This results in an appropriate number of agents staying active to accomplish the task throughout the aggregation process. This in turn (based on the explanation in section 4.1) contributes to the collaborative team achieving larger clusters (see Figure 6) as well as to a faster decrease in the activity of the robots as soon as a single cluster arises. Consequently, the PuFT algorithm offers the best efficiency in Arena3 (see Table 2 and Figure 7). Figure 7 presents the IC for the different allocation algorithms as a function of the observation time i.e., the integration time interval in Arena3. Error bars represent the standard errors among sets of simulation runs. From Figure 7, it appears that despite the dynamic change in the workload the algorithms with worker allocation are still more efficient than that without any worker allocation. The PuFT algorithm offers the best efficiency after the introduction of the additional seeds by exploiting teammates' collaboration via peer-to-peer communication.

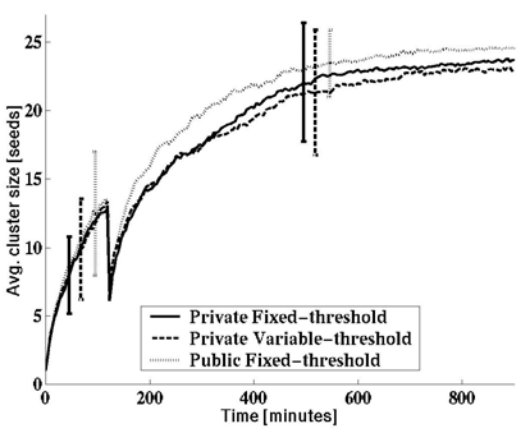

Figure 6. Average cluster size for aggregation experiment with worker allocation algorithms in an 80X80 cm arena. 5 more seeds dropped in the arena after 2h into the experiment.

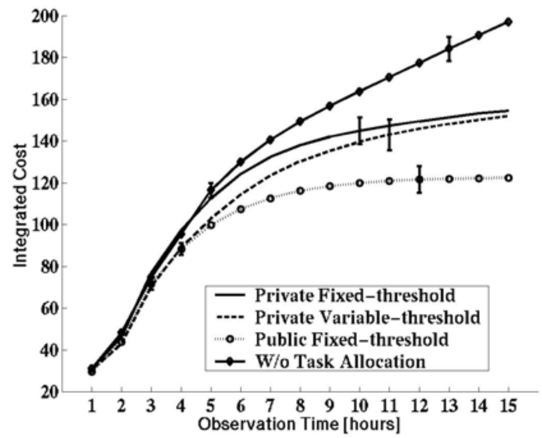

Figure 7. Integrated cost as a function of observation time for aggregation experiment in Arena3

5. CONCLUSION

In this paper, we have presented a comparative study of three scalable, distributed, threshold-based worker allocation algorithms that allow a team of autonomous, embodied agents 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 compared their efficiency and robustness in a collective manipulation case study concerned with gathering and clustering of seeds.

Teams consisting of active workers dynamically controlled by one of the allocation algorithms achieve similar or better performances 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. Moreover, after a systematic optimization process in a given environment common to all the algorithms, it appeared that quantitative differences in efficiency among these allocation algorithms are less apparent. Although PuFT appears to be one of the two most efficient algorithms in the smaller arena, its energy cost due to communication was not included in these experiments. In addition, even if the demand can be estimated more accurately by sharing information and the collective reaction to external perturbations is faster, the PuFT algorithm still suffers from the same drawback as the PrFT algorithm in facing environmental changes for which its threshold was not a priori optimized for.

We believe that combining the characteristics of PuFT and PrVT, threshold variability and information sharing will allow us in the near future to further improve the robustness (and maybe the efficiency) of the resulting worker allocation algorithm. In particular, transforming the calibration rule in a continuously adapting algorithm as suggested in [16] seems to be a promising solution to static and dynamic external perturbations.

Finally, it is worth noting that an agent that withdraws from a task can allocate itself to a different task following a similar response-to-stimulus mechanism, thus making this set of algorithms easily applicable to multi-task problems and more complex labor division schemes.

6. ACKNOWLEDGEMENTS

Special thanks to Xiaofeng Li for helping to collect systematic simulation data. This work is supported in part by the TRW Foundation and the TRW Space and Technology Division. Further funding was received from the Caltech Center for Neuromorphic Systems Engineering as part of the NSF Engineering Research Center program under grant EEC-9402726.

7. REFERENCES

[1] Agassounon, W., Martinoli, A., and Goodman, R. A Scalable, Distributed Algorithm for Allocating Workers in Embedded Systems. In Proceedings of IEEE SMC '01 Tucson AZ, October 2001), 3367-3373.

[2] Beni, G., and Wang, J. Swarm Intelligence. In Proceedings of the 7th Annual Meeting, Robotics Society of Japan, 1989, 425-428.

[3] Böhringer, K., Brown, R., Donald, B., Jennings, J., and Rus, D. Distributed Robotic Manipulation: Experiments in Minimalism. Proceedings of ISER '95 (Stanford, CA, June 1995), 11-25.

[4] Bonabeau, E., Theraulaz, G., and Deneubourg J.-L. Fixed Response Thresholds and Regulation of Division of Labour in Insect Societies. Bulletin of Mathematical Biology, 1998, Vol. 60, 753-807.

[5] Bonabeau, E., Dorigo, M., and Theraulaz, G. Swarm Intelligence: From Animal to Artificial Systems. SFI Studies in the Science of Complexity. Oxford University Press, New York NY, 1999.

[6] Gerkey, B. P., and Mataric, M. J. MURDOCH: Publish/Subscribe Task Allocation for Heterogeneous Agents. In Proceedings of Autonomous Agents'00, Barcelona, Spain, June 2000, 203-204.

[7] Holland, O., and Melhuish, C. R. Stigmergy, Self-organization, and Sorting in Collective Robotics. Artificial Life, 1999, Vol. 5, 173-202.

[8] Ijspeert, A. J., Martinoli, A., Billard, A., Gambardella L. M. Collaboration through the Exploitation of Local Interactions in Autonomous Collective Robotics: The stick Pulling Experiment. Autonomous Robots, 2001, Vol. 11, No. 2, 149-171.

[9] Krieger, M. B., and Billeter, J.-B. The Call for Duty: Self-Organized Task Allocation in Population of up to Twelve Mobile Robots. Robotics and Autonomous Systems, 2000, Vol. 30, No. 1-2, 65-84.

[10] Martinoli, A., Ijspeert, A. J., and Mondada, F. Understanding Collective Aggregation Mechanisms: From Probabilistic Modeling to Experiments with Real Robots. Robotics and Autonomous Systems, 1999, Vol. 29, 51-63.

[11] Martinoli, A., Ijspeert, A. J., and Gambardella, L. G. A Probabilistic Model for Understanding and Comparing Collective Aggregation Mechanisms. In Proceedings of ECAL '99 (Lausanne, Switzerland, September 1999), 575-584.

[12] Michel, O. Webots: Symbiosis Between Virtual and Real Mobile Robots. Proceedings of ICVW '98 (Paris, France, 1998), 254-263.

[13] Mondada, F., Franzi, E., and Ienne, P. Mobile Robot Miniaturization: A Tool for Investigation in Control Algorithms. Proceeding of ISER '93 (Kyoto, Japan, October 1993), 501-513.

[14] Pacala, S. W., Gordon, D. M., and Godfray, H. C. J. Effects of Social Group Size on Information Transfer and Task Allocation. Evolutionary Ecology, 1996, Vol. 10, 127-165.

[15] Parker, E. L. ALLIANCE: An Architecture for Fault Tolerant Multi-robot Cooperation. IEEE Transactions on Robotics and Automation, 1998, Vol. 14, No. 2, 220-240.

[16] Theraulaz, G., Bonabeau, E. and Deneubourg, J.-L. Response Threshold Reinforcement and Division of Labour in Insect Societies. In Proceedings of Royal Society of London, Series B, 1998, Vol. 265, 327-332.