Collective Sorting with Multiple Robots

Tao Wang
Department of Computing Science
University of Alberta
Edmonton, Alberta Canada T6G 2E8
Email: trysi@cs.ualberta.ca

Hong Zhang
Department of Computing Science
University of Alberta
Edmonton, Alberta Canada T6G 2E8
Email: zhang@cs.ualberta.ca

Abstract – **Inspired by the behavior of social insects, we tackle the problem of sorting objects with a group of robots under the control of reactive behaviors. Our control algorithm is based on earlier studies of this problem, but depends on more sensing than the minimalist solution. With the additional information and our simple behavioral rules, we demonstrate through simulation studies that our control algorithm is able to create a complete separation of objects of two different classes, and that it guarantees convergence of the sorting process, which previous algorithms could not achieve. This result is independent of the number of robots participating in the task, the initial configuration of the world, and the number of objects to be sorted. We also show indirectly that sorting is not a strictly cooperative task in the sense that even a single robot is capable of performing the task, though at a reduced pace. Finally, we present a model that characterizes the growth of object clusters, which can be used to understand the dynamics of the sorting process.**

Keywords – collective sorting; local sensing; behavioral rules

I. Introduction

Collective behaviors of social insects offer inspirations on designing artificial swarm systems that are distributed, self-organized, and scalable. Inspired by the sorting behavior of ants, we investigate the problem of sorting objects of two classes by a team of robots. The robots employ only local perception of the environment and have no a priori information about the distribution or number of the objects to be sorted in the environment. Such a system can be useful when robots are built at the micro or nano-scale to perform such tasks as cleaning, recycling, and construction.

In the nest of an ant colony, larvae, cocoons, and eggs are arranged in distinct patterns such as separate piles [5] or concentric rings [7]. Biologists speculated that these patterns might be created by ants moving items probabilistically without central administration [5], and the mechanisms of self-organization, stigmergy [8], and self-sieving might be involved. In robotics research, Deneubourg's simple behavioral algorithm captures many features of the ant sorting behavior [6], [7]. Deneubourg has theorized that ants pick up and put down objects in a random manner. The probabilities of moving items are given by a short-term memory of an ant's recent observations, and the sorting behavior can be viewed as a pattern generation process through self-organization [3], [4]. In addition, stigmergy has been considered as a mechanism for behavior coordination in which modification of a shared environment serves as a cue to direct the future activities of the ants. Furthermore, the self-sieving mechanism (also called self-sorting by size, that is, small particles are able to lie between the larger particles under the influence of shaking and gravity) complements the probabilistic model of ant behavior in explaining the formation of the ring-shaped brood pattern in Leptothorax ant colonies [1]. Although biologists have closely observed - and postulated explanations of – the ant sorting behavior, how an ant colony sorts collectively is still not fully known.

Distributed clustering or, more recently, sorting is considered as a benchmark task for collective robotics [3]. As a follow-up to [5] which relies on short-term memory to record what has been perceived in the recent past, Beckers *et al.* presented experiments of robots clustering pucks into one pile [2], by picking up objects randomly and dropping them after encountering other objects. Melhuish *et al.* further studied the collective sorting problem from a minimalist perspective, and their system produced annular sorting results [8], as well as sorting of multiple classes of objects [12], with control algorithms that succeed often but fail sometimes.

In our collective sorting project, we are interested in the design of a behaviour-based multi-robot system for collective sorting. Our work is similar in philosophy to the threshold mechanism used by Beckers *et al.* [2] and its extension by Melhuish *et al.* [9]. However, we concentrate on seeking a reliable approach to the sorting problem and systematically studying the sorting process. We conjecture that the degree of convergence of the sorting task can be assured by the an increase in the sensing capability of the robots, and that increasing the number of robots in a sorting task can only speed up the process at its early stage.

The rest of the paper will be organized as follows. Section 2 describes the sorting problem in detail and the design of our robot system. In Section 3, we present simulation results to demonstrate the convergence of the our sorting algorithm and the independence of the result from the robot population size, the number of objects, and initial configuration. We also provide a model that characterizes the growth of the object clusters. Finally, conclusions are made and future work outlined in Section 4.

II. Robot Collective Sorting

We consider sorting of objects in the context of a multi-robot system, in which a number of robots rearrange scattered objects of different classes into clusters in a working arena (see Figure 1). If the robots knew the positions of all the objects and those of the final clusters of the sorted objects, then it would be trivial to arrive at a solution to the collective sorting problem. Instead, we assume robots do not have global information such as the position of the objects/clusters and the number of the objects to be sorted. To arrive at a scalable solution to the problem, we further impose the constraint of no explicit communication between robots. Instead, our robots rely on local sensing to acquire information about the state of the world, and behave independently in a distributed fashion. How could a group of robots collectively sort the scattered objects with the above constraints?

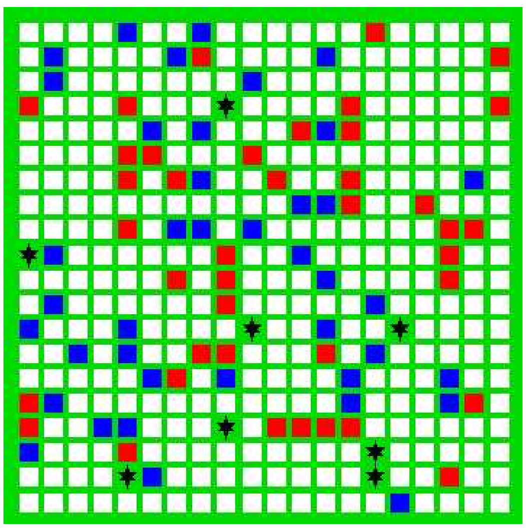

Fig. 1. The simulation model of the environment in which robots (stars) are to rearrange two types of objects (dark and grey squares) so that objects of the same type will form exactly one cluster.

A. Performance Metric

In order to measure performance, it is important to define an intuitive metric for the convergence of the sorting process. Let M be the number of types of initially randomly scattered objects, which are to be organized into clusters, with one type of objects in each cluster. The total number of objects of type i is denoted by N_i. Each robot has only local perception and affects the environment by relocating objects. At any given time, there are K_i number of clusters of object type i in the arena, each of the size $n_{ij}(t)$, where j indexes the clusters. We use the following formula to measure the degree of convergence of the sorting process:

$$T_{success}(t) = \frac{1}{M}\sum_{i=1}^{M}\left(\sum_{j=1}^{K_i}\left(\frac{n_{ij}(t)}{N_i}\right)^2\right) \tag{1}$$

where $M = 2$ in our study. Note that $T_{success}$ equals one when the robots sort the objects into two completely separate clusters, one for objects of type 1 and the other for objects of type 2.

B. Control Design

Recall that we restrict our solutions to those that require only local sensing, with no communication and memory. The basic idea is that the isolated objects are likely to be picked up and deposited at locations where more items of the same type are present [5]. This will lead to a congregation of objects of the same type. If small clusters attract passing robots to deposit their objects of the same type, then clusters will grow incrementally till the robots are terminated or the system reaches its stable state. To give chances for several clusters to merge into a bigger cluster, the robots can employ probabilistic rules regarding the deposit or removal of objects, and this is achieved indirectly by restricting a robot's field of perception.

The robots are assumed to be able to random-walk in the grid world (Figure 1), moving one step at a time, in one of eight possible discrete directions, spaced $45°$ apart. In addition, a robot is able to sense only three squares or cells directly in front of its heading. Since a cell is either empty or occupied by an object, the input state space to the robot control algorithm is of size 3^3. Robots can move on an empty cell and deposit the object it is holding at that cell; alternatively, a robot without any object can approach an occupied cell and pick up the object. For collision avoidance, we make sure that two robots do not occupy the same cell. Based on local sensing, robots are assumed to be able to recognize whether two objects are the same or different types.

All robots run an identical algorithm and are unaware of the presence of other robots. The control algorithm is based on a set of three simple rules, which map easily to a behavior-based controller.

- Rule1: With or without an object, randomly choose a direction and move in that direction one step, subject to collision constraint.
- Rule2: For a robot that does not hold an object, if it encounters an object in front, pick up the object if the object is different from at least one of the two on either side.
- Rule3: For a robot that holds an object, if it encounters an empty cell in front, put down the object if its type matches one of two objects on either side.

The first rule is a wandering behavior and moves a robot around in the environment. The second rule dictates when a robot picks up an object based on locally sensed information. The third rule defines when for the robot to lay down an object that it is carrying. In the wandering state, a robot avoids collision with obstacles (i.e., other robots in our simulated environment or a boundary of the world) by turning a random heading angle at $45°$ increments. Our system design is flexible enough to allow us to conduct experiments with different simulation parameters, such as the number of robots, the size of the environment, the number of objects, and the initial configuration.

Comparing the rules in our control algorithm with those in previous studies of the collective sorting problem, we note that our algorithm depends on local sensing, rather than short-term memory, as in the case of [5], although its idea of using a measure of the local object-type distribution to determine the robot's action is still used. In addition, our algorithm employs more local sensing than the studies by Beckers *et al.* [2] and Melhuish *et al.* [9], and this perhaps explains why we are able to achieve a significant improvement in performance with respect to the convergence behavior of the sorting algorithm, as will be shown in the next section.

III. Experimental Results

The purpose of the experimental studies is to investigate the convergence properties of the collective sorting process, governed by the behavior rules defined in the previous section. We are also interested in the sensitivity of the convergence result with respect to the simulation parameters (e.g., the number of robots).

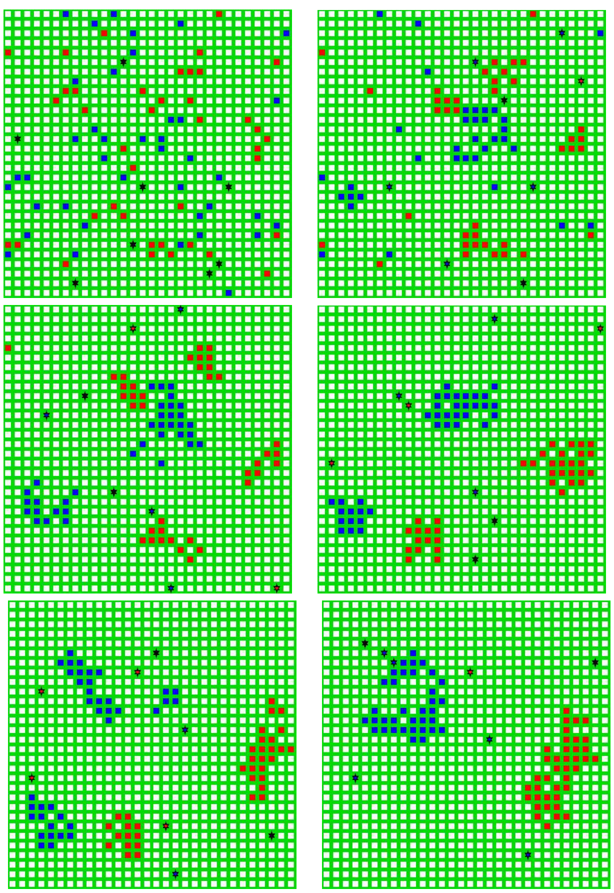

Fig. 2. Snapshots of collective sorting process by 8 robots sorting 40 objects of each type in a 30 by 30 grid world, at simulation step (from left to right, top to bottom) 0, 2048, 4096, 27,648, 37,888, and 58,8831, respectively.

Figure 2 shows some snapshots of a typical collective sorting process in an experiment with eight robots (the star) sorting 80 objects (40 black and 40 grey) in a 30×30 grid world. The execution of this experiment takes 58,831 simulation steps.

A. *Performance vs. Swarm Size*

To investigate the relationship between the swarm size and the convergence of the sorting task toward completion, we conduct a total of 400 experiments of sorting 18 objects, nine black and nine grey, in a 10×10 grid world. The experiments are divided into four subgroups, according to the number of the robots involved, which employ one, two, four, and six robots, respectively. For each simulation, we record the time history of the convergence measure (Equation (1)), and this measure is averaged over the 100 experiments for each subgroup.

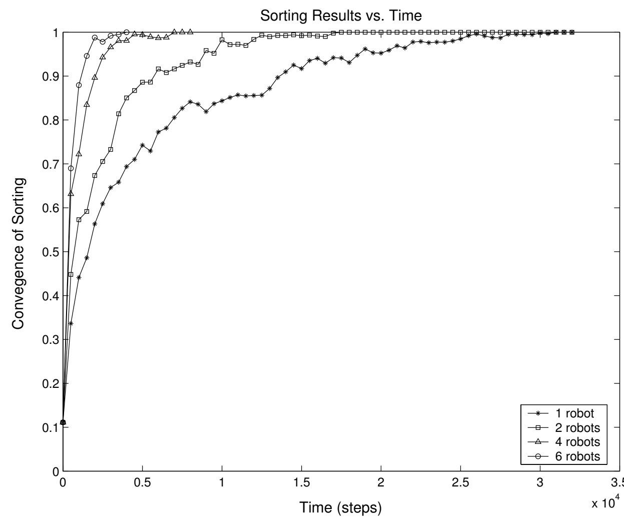

Fig. 3. Performance of our sorting algorithm - with uniform initial distribution of objects - in terms of the averaged completion metric defined by Equation (1), one curve for each subgroup of our study

Figure 3 shows the performance of the sorting algorithm of the four subgroups. At the beginning of the experiments, robots are placed at the centre of the arena and objects are uniformly distributed in the arena. The degree of completion for this initial configuration is 0.11 according to the metric of Equation (1). An experiment is terminated when robots complete the task, that is, when exactly one cluster is created for each type of objects. The most important conclusion that can be drawn from Figure 3 is that our sorting algorithm always converges, for all the simulation experiments we have conducted, although they take varying numbers of steps. Figure 3 also shows that by increasing the number of robots in the task, the sorting process can be sped up, but only at the early stage of the process; however, it is clear that how well the sorting task is completed is independent of the number of the robots in the swarm system. The sorting task is therefore not a strictly cooperative task in that it can be completed by a single robot.

To test whether the initial configuration of a simulation has an impact on the convergence of the sorting process, we conduct experiments identical to the above except with a

TABLE I

MAXIMUM, MINIMUM, AND AVERAGE EXECUTION TIMES IN SIMULATION STEPS OF THE FOUR SUBGROUPS OF CASES IN OUR STUDY

# of Robots	1	2	4	6
Minimum	1,334	688	345	187
Average	9,294	4,558	1569	969
Maximum	30,550	17,034	6,630	3,623

random initial distribution of the objects and robots. Figure 4 shows the results of these experiments. It can be seen that that these results are nearly identical to those in Figure 3, indicating the independence of the algorithm's convergence from the initial configuration.

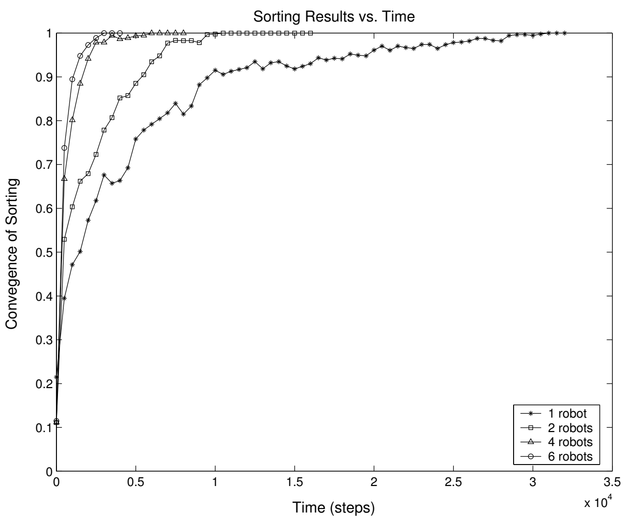

Fig. 4. Performance of our sorting algorithm - with random initial distribution of objects - in terms of the averaged completion metric defined by Equation (1), one curve for each subgroup of our study.

Table I compares the best, worst, and average cases of the four robot populations in our simulation study, where the numbers are in simulation steps. As one can see, there is a large range of execution times for the same robot collective size, and the execution time can be reduced in general with an increase in the number of robots performing the task.

B. Critical and Peak Cluster Size

We have also studied the sorting algorithm in a series of experiments with different simulation parameters from those in the previous section. These experiments include 150 trials for four robots sorting 20 objects of each type in a 10x10 grid world, 100 trials for eight robots sorting 20 objects of each type in a 20x20 grid world, 20 trials for eight robots sorting 40 objects of each type in a 30x30 grid world. Objects and robots are initially randomly distributed in these experiments because the experimental studies in Section 3.1 show that the convergence of the algorithm is not initial distribution sensitive. We observe that scattered objects of the same type are grouped into small clusters very quickly at the beginning of the experiments. Then small clusters are more likely to grow in size than isolated objects. After the cluster size reaches a certain threshold, some of the the clusters are occasionally split, and this is more likely to happen to small clusters than to large ones. However, one cluster per object type eventually emerges, running a sufficiently long period of time.

In the sorting process, objects are deposited and removed from the clusters stochastically. Whether a cluster grows is affected by the behavior rules implemented in the robots. We can measure the net change of a cluster by the growth of a cluster, which is the fraction of objects deposited to a cluster over the cluster size at that moment. Figure 5 depicts the history of the growth of a cluster based on observations made on the data collected from experiments. There are two important parameters that define how the cluster size changes. We name them the critical size and the peak size.

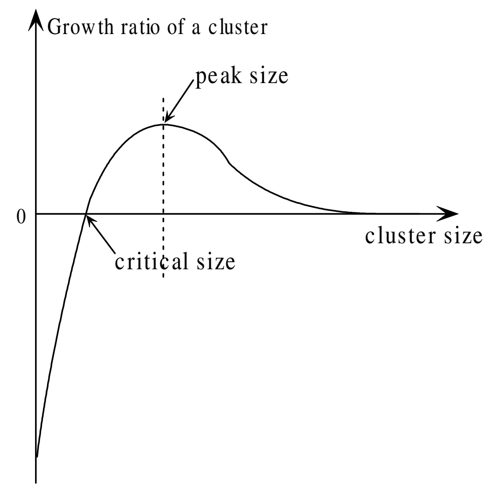

Fig. 5. Cluster size is determined by the grow ratio of a cluster.

We can categorize the behaviours of clusters during the sorting process into three classes, determined by a cluster's size relative to the critical size and peak size. If the size of a cluster is less than the critical size, the growth of a cluster is likely to be negative. That is, more objects are removed from the cluster than deposited. Therefore, the cluster becomes smaller and smaller. When the cluster size is equal to the critical size, the cluster is almost unchanged as its net change approaches zero. As the sorting is carried out by random-walking robots, the cluster may become bigger even when it is the same as the critical size. The temporary stability in the cluster size is changed probabilistically. Finally, if the size of the cluster is greater than the critical size but less than the peak size, the cluster grows very quickly until the size of the cluster reaches the peak size. When its size is greater than the peak size, it still grows but at a much slower speed. This explains why exactly one cluster will form, provided that an experiment is given a long enough time.

IV. Conclusions

In this paper we have presented our simulation study of robot collective sorting. We have identified a set of simple rules which mimic decentralized, reactive behaviors in social insects, and lead to the perfect sorting of two classes of objects, when the rule set is implemented on a group of robots. This represents a significant improvement over the previous studies where no local-based control rules have been able to produce the same kind of result. We attribute this success to the slight increase in the sensory information made available to the robot and to the proper design of the control algorithm, which performs the best among many that we have examined. More than 1000 experiments with different parameters have been run, and results suggest that the convergence of the sorting process is independent of initial system distribution, robot population size, as well as the number of objects, although an increased number of robots can reduce the amount of time for the system to converge. One interesting observation we have made regarding the dynamics of a typical cluster size is that two sizes determine whether the cluster will grow and the rate at which it will grow.

Our future work will look at how to quantitatively measure the capabilities of the robots, and derive the critical/peak size in the model of Figure 5 theoretically from the behavioral rules of the robots. This theoretical model will shed light on how to prove the convergence property we have observed experimentally, in a way similar to [10]. This will allow us to better explain the behavior of the robot collective as a whole. Finally, we will compare our simulation results with those in the experiments involving physical robots [11].

Acknowledgment

Research supported by University of Alberta.

Thanks to Dale Schuurmans and Rich Sutton for providing financial and moral support and to Michael Bowling for helpful discussions of collective sorting behaviors.

References

[1] G. Barker and M. Grimson. The physics of muesli. *New Scientist*, 126 (1718),1990, pp. 37–40.

[2] R. Beckers, O. E. Holland, and J. L. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In *Proceedings of the Fourth International Workshop on the Syntheses and Simulation of Living Systems: Artificial Life IV*, pages 181–189, 1994.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural To Artificial System*. Oxford University Press, 1999.

[4] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, 2001.

[5] J.L.Deneubourg, S.Goss, N.Franks, A.Sendova-Franks, C.Detrain, and L.Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats SAB90*, pages 356–363, 1990.

[6] J.L. Deneubourg, J.M. Pasteels, and J.C. Verhaeghe. Probabilistic behavior in ants: a strategy of errors? In *J. theor. Biol.*, 105, pages 259-271,1983

[7] N.R. Franks and A.B. Sendova-Franks. Brood sorting by ants: distributing the workload over the work-surface. *Behav. Ecol. Sociobiol*, 30: pp.109-123, 1992.

[8] Owen Holland and Chris Melhuish. Stigmergy, self-organization, and sorting in collective robotics. In *Artificial Life V*, pages 173–202, 1999.

[9] Chris Melhuish, Owen Holland, and Steve Hoddell. Collective sorting and segregation in robots with minimal sensing. In *5th Int. Conf. on the Simulation of Adaptive Behaviour*, 1998.

[10] S. Kazadi, A. Abdul-Khaliq, and R. Goodman. On the convergence of puck clustering systems. *Robotics and Autonomous Systems*, 38(2):93–117, 2002.

[11] S. Verret, *Communication and Sensing in Robot Collective Sorting*, M. Sc. Thesis, Department of Electrical and Computing Engineering, University of Alberta, 2004.

[12] Wilson, M., Melhuish, C., and Sendova-Franks, A., Multi-Object Segregation: Ant-Like Brood Sorting Using Minimalist Robots, In *International Conference on Simulation of Adaptive Behaviors*, August 4-9, 2002.