Sorting in Swarm Robots Using Communication-Based Cluster Size Estimation

Hongli Ding and Heiko Hamann
Department of Computer Science,
University of Paderborn, Paderborn, Germany
hongli.ding@uni-paderborn.de, heiko.hamann@uni-paderborn.de

April 18, 2015

Abstract

Inspired by sorting behaviors of social insects, we are interested in sorting by robot swarms using only local information and hence achieving high degrees of robustness and scalability. In this work, we propose a gossip-based sorting method which allows two swarms of simple homogeneous autonomous robots to sort themselves in two not pre-assigned areas. Key feature of this method is the estimation of cluster sizes based on communication that allows to determine the local majority. In a series of simulation experiments, we show the effectiveness of the approach and investigate the influence of different swarm sizes.

1 Introduction

Recent research shows that social insects sort their brood in sophisticated patterns. These well-organized brood sorting patterns emerge spontaneously from dynamic interactions during the process of depositing and removing brood. During this sorting process, no specified spatial plans or any global representation is required, nor any hierarchical decisions are made [1]. By interacting among other individuals and with the environment, individuals act following their own goals and knowledge about the environment. The collective behavior on the group level emerges from the sum over all individual decisions, actions, and the interactions among individuals and the environment. The sorting system of social insects has many attractive features such as scalability, flexibility, and robustness. Abstract models based on sorting behaviors of social insects have been applied in many areas such as search, collective sorting, data mining, numeric data analysis, and graph partitioning [7]. Inspired by how ants and honey bees sort their broods, we are interested in how to implement these natural sorting behaviors and strategies in a swarm of robots. We simplify the sorting task to sorting robots instead of objects. This preliminary work aims at sorting robots of different classes which can be considered as robots carrying objects of different types. Hence, our algorithm still aims to sort objects.

Object sorting by swarm robots is a complex task which involves mechanism such as self-organization, collective decision making, and pattern recognition. Abstract models of sorting objects by a group of minimalist homogeneous robots were proposed by Deneubourg et al. [1]. These models are based on simple rules which are used to determine the probabilities of picking up and dropping down objects. The major drawback of these models is the complexity of the procedure to obtain the local object density. Being limited to the minimal sensors of real robots imposes a challenge in achieving sorting similar to that observed in ants. This method was extended to sort more than three types of objects [11]. Inspired by [1], an approach using an overhead camera to identify robots, object positions and orientations, and global data about the entire arena was proposed [8]. It is unclear how this approach could be transferred to use local sensors only. This approach was extended by adding a forward-facing camera on each robot, which not only allows robots to share image data with their neighbors, but also allows robots to estimate cluster sizes [12]. Both approaches [12, 8] achieve good sorting results while it is hard to apply these approaches to real robots, in particular swarm robots, due to the applied complex and sophisticated sensors.

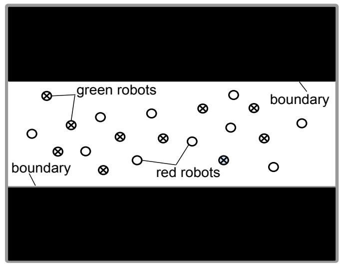
In this paper, we describe a gossip-based sorting method which aims to sort different robots by using only local information and simple onboard sensors. Similar to previous studies, our algorithm is based on simple behavioral rules. In addition, we examine how the system performance changes with different swarm densities (different swarm sizes on a constant area).

This paper is organized as follows. Sec. 2 describes the scenario and objective of this work. A global approach is proposed in this section. Sec. 3 focuses on the proposed sorting algorithm. In Sec. 4, we report the experimental setups and the simulation results. We conclude the paper and outline the future work in Sec. 5.

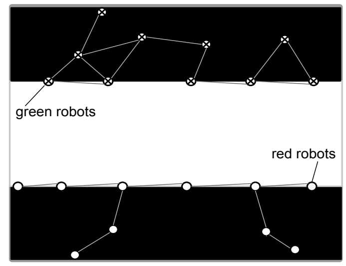
2 Scenario Description

In a given space as shown in Fig. 1a, the ground is divided into three areas: two black areas and one white area. Green and red robots are initially randomly distributed in the white part of the arena. These two groups of robots have to sort themselves in the two black areas using only local information (shown in Fig. 1b).

In this scenario, the allocation of the two black areas to the two robot groups is not predetermined. The robot swarms decide collectively on this allocation at runtime during the sorting process by interacting with other robots and the environment. Each agent's individual decision is based on local information only. The mechanism is implemented as a self-organizing system which requires no global information like spatial plans, hierarchical decisions, or message broadcasting. We investigate how swarm robots can make correct decisions and self-organize themselves to achieve this complex sorting task under these



(a) initial setup (red and green robots positioned within the white area)



(b) expected result (green robots share one black area and red robots share the other black area)

Figure 1: Initial setup and expected result of the simulation experiment.

strict constraints.

The two groups of robots are randomly distributed in the white area initially as shown in Fig. 1a. At the beginning, all robots walk randomly in the white area by avoiding obstacles, especially other robots. When a robot detects a black ground, it stops while continuing to communicate with other robots in its neighborhood. At the same time, each stopped robot identifies other stopped robots in its neighborhood based on a unique robot identity (UID). They communicate and transfer the UIDs of stopped neighbors to other stopped robots and count the number of stopped robots of both groups at the boundary. When one group of stopped robots represents the majority at this boundary, then the minority robots leave this boundary and search for an unoccupied boundary. With time going on, the number of robots representing the majority is expected to increase at this boundary. Robots position themselves in a line formation at the boundaries to form, what we call a *robot barrier*. It allows the same robots to pass and prevents the passage of other robots. For example, if the barrier is composed of a certain number of red robots which is above a threshold, then the red robots are allowed to cross this barrier. They hence allocate this black area and stay inside of it. In contrast, green robots are not allowed to pass this barrier. The control algorithm is organized in several modules as follows:

- *Count robots*: Stopped robots count the number of different robots staying at the boundary by communicating with each other.
- *Minority robots leave*: After counting the number of different robots at a boundary, the robots know whether they represent the minority. If they are the minority, they leave this boundary for a random walk in the white area.
- *Barrier formation*: When a moving robot reaches the communication range of a stopped robot of the same color, it moves on a circular trajectory around the stopped robot in order to position itself in a neighboring

position at the boundary. It stops once a black ground is detected, and consequently becomes a part of the robot barrier. The circular trajectory enforces a certain distance between robots in the robot barrier.

- *Pass robot barrier*: When the number of majority robots on the boundary exceeds a threshold, robots of the same color are allowed to pass the barrier and position themselves at an appropriate spot within the black area.

3 Gossip-based Sorting Algorithm

In this work, the gossip-sorting method is based on gossip communication to count the number of robots in a swarm. The idea of gossip communication is that robots exchange their local knowledge in pairs when robots reach the communication range of each other. Each robot that is stopped at a boundary and that is in the communication range of other robots, communicates with each neighbor, and they mutually exchange their local knowledge. In this work, robots exchange information about the number of robots of each group at the boundary that the robot either can perceive itself or about which it has received information via past gossiping. After several gossip communication iterations between robots, every robot knows the number of different robots at the same boundary.

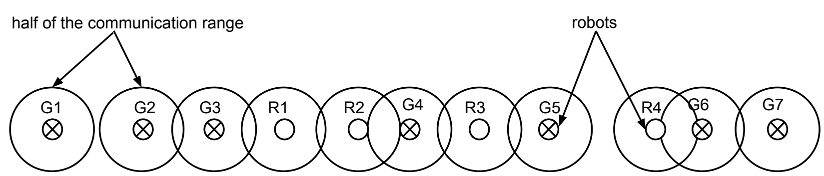


Figure 2: Robot barrier under construction.

	Direct line communication	Gossip communication	After gossip communication
G1	G1	- -	- -
G2	G2, G3	G3 says: G2,G3,R1 are in the barrier	G2 knows G2,G3,R1
G3	G2, G3, R1	R1 says: G3, R1,R2 are in the barrier	G3 knows G2,G3,R1,R2
R1	G3, R1, R2	G3 says: G2,G3,R1,R2 are in the barrier R2 says: R1,R2,G4 are in the barrier	R1 knows G2,G3,R1,R2,G4
⋮	⋮	⋮	⋮

Table 1: Example of the gossip-based communication.

Fig. 2 is an example of the gossiping method which shows a robot barrier under construction. This robot barrier consists of two groups of robots: green labeled with UIDs G1, G2, etc., and red labeled with UIDs R1, R2, etc. We assume that the communication is robust (subject to future work, see Sec. 5). Robots are assumed to communicate only with neighbors in the line of sight, that is, messages are neither addressed to a particular robot (i.e., no multi-hop communication) nor sent through obstacles including other robots. In Fig. 2, robot G1 cannot communicate with robot G2, because they are not in each others communication range. Robot G2 cannot communicate with robot R1, because multi-hop messaging is not available (robot G3 would need to serve as relay).

Table 1 shows how robots exchange messages based on gossip communication. As shown in Fig. 2, robot G1 is isolated. Similarly, Robot G5 and R4 cannot communicate with each other. In consequence, the robot barrier shown in Fig. 2 consists of three sub-groups: $S_1 = \{G1\}$, $S_2 = \{G2, G3, R1, R2, G4, R3, G5\}$, and $S_3 = \{R4, G6, G7\}$. Obviously, sub-group S_1 will consider itself as majority robot until the approach of other red robots. Each robot in sub-group S_2 knows about all other robots from the same sub-group after the gossip-based communication. Each robot has a counter for red robots n_r and a counter for green robots n_g. In this case, green robots are the majority ($n_g = 4$, $n_r = 3$). Consequently, red robots leave the barrier and start a random walk into the white area until they find another place to stay. Similarly for sub-group S_3, green robots are the majority and hence red robot leaves the barrier. In this case, all red robots leave the boundary and there are only green robots left. In the sequel, other green robots have a higher probability to stop at this boundary compared to red robots, even before the green robots have formed a complete barrier. If a red robot tries to stop at this boundary, it detects the green majority and leaves. Similarly, the probability of forming a barrier at another area is increased for the red robots. We set a threshold to determine the length of robot barriers (The threshold is set to 5 for all experiments in this work). Once the majority group size at a barrier reaches this threshold, other robots of that kind are allowed to pass the barrier through two robots of the same color. Hence, this robot group claims the respective black area.

4 Simulation Environment and Results

In this section, we give a brief description of our simulation environment. Results of our sorting method for different swarm densities are shown. Videos of our simulations are available online[1].

4.1 Simulation environment

Experiments were conducted using the foot-bot robot [14] in the ARGoS simulator [13]. ARGoS is designed to simulate complex experiments involving large swarms of robots of different types. It allows to transfer robot controllers from the simulation directly to real robots without any modification [13].

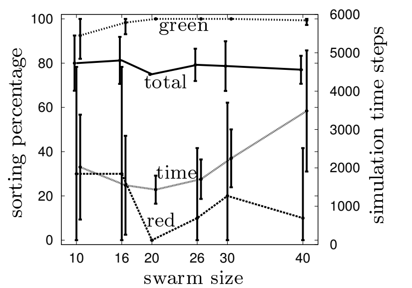
In the following simulation experiments, we use an arena of 3 meters by 3 meters, divided to three areas (two blacks, one white). Two groups of robots (green and red) are randomly distributed in the white area. The communication range of each robot is set to 60 cm, which is the minimal communication range required to allow a robot to pass between two robots in communication distance

[1]see https://www.youtube.com/user/SortingRobots

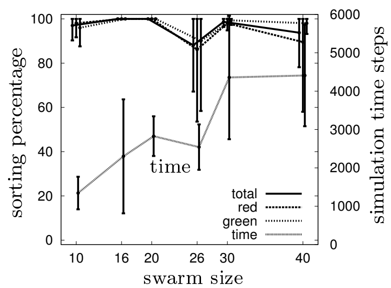
(requirement for passing barriers). With this communication range the robot covers 12.6% of the total arena. We use only basic onboard sensors: IR sensors, range and bearing sensors.

4.2 Simulation results

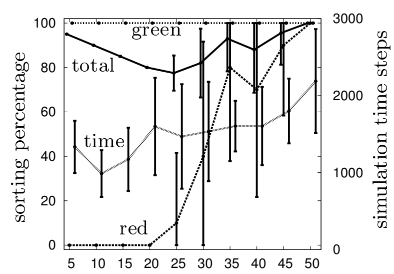
Second swarm as disturbance. In this experiment, our sorting method is tested for different swarm sizes $N \in \{10, 16, 20, 26, 30, 40\}$ (i.e., different swarm densities because the area is constant) forming groups of red and green robots in the following way: $(n_r, n_g) \in \{(2, 8), (4, 12), (5, 15), (6, 20), (8, 22), (10, 30)\}$. The swarms are composed of approximately 25% of red robots and 75% of green robots. Initially, both red and green robots are uniformly positioned in the white area. For each parameter setting, 10 runs were done. The simulation ends when either all robots have stopped in black areas or one group of robots has stopped in black areas while the other group of robots moves in the white area. We define the sorting rate as the percentage of robots sorted in black areas (i.e., robots positioned correctly in the neighborhood of their own kind). When both kinds of robots are found within the same black area, we consider the group of robots which represents the majority as sorted. Fig. 3a shows the sorting rates and the required time for the different swarm sizes.



(a) 25% red robots, group compositions $(n_r, n_g) \in \{(2, 8), (4, 12), (5, 15), (6, 20), (8, 22), (10, 30)\}$



(b) 50% red robots



(c) varied swarm ratios

Figure 3: Results of the simulations, sorting rate and simulation time (mean values of 10 runs, error bars give standard deviation).

From Fig. 3a, we can see that the disturbance from approximately 25% of red robots has no significant influence on the sorting results of green robots, while the low sorting rate of red robots decreases the total sorting rate. The relatively small numbers of red robots induce a small probability to occupy any boundaries. However, the total sorting rate is still bigger than 75%.

As seen in Fig. 3a, the swarm size has little influence in these experiments. Only the required time indicates a trend to increased times for bigger swarms. Furthermore, there is a trend indicating that a swarm size of $N = 20$ might be optimal in terms of the system's convergence time.

The total sorting rate is influenced by several conditions. The total swarm density is too low to guarantee a good cooperation among robots. The sizes of the two robot groups are of importance. The green robots are enough to occupy both boundaries, hence, the red robots have a limited chance to occupy any boundary. Many of the experiments end with the red robots being trapped in the white area. Therefore, the red robots have a small chance to sort themselves.

The curve of required time decreases at first, probably because the low swarm density provides insufficient opportunities for cooperation among individuals. The robots spend much time searching or waiting for other robots. We therefore observe a tradeoff between positive effects of interference and obstructive interference which has commonly seen in swarm intelligence [**?**]. This indicates the existence of an optimal density for this scenario.

Two swarms of equal size. In this experiment, we keep the same simulation setups except the composition of two swarms. 10 runs were done for swarms composed of $N \in \{10, 16, 20, 26, 30, 40\}$ robots. Each robot swarm is composed of 50% of red robots and 50% of green robots.

Fig. 3b shows the results for these settings. Both robot swarms sort themselves efficiently for different swarm sizes. In the worst case, the total sorting rate is still higher than 85%. The required time clearly increases with the swarm size. Comparing to the experiments with only 25% red robots, this experiment achieves a better sorting rate, while the required sorting time relative to the same swarm size is longer. This seems mainly because both swarms have the same probability to occupy any boundary. The self-organized collective decision-making process about which group is occupying which black area takes time.

Optimal ratio for two swarms. Given the previous results, we are interested in how the swarm behaves with different proportion for two swarms. What is the optimal proportion? The total swarm size is fixed to $N = 20$ robots. For each series of runs, the number of red robots is increased by one robot and the number of green robots is reduced by one robot (5% of the total swam size).

The results are given in Fig. 3c. When the red swarm size is $n_r \leq 20\% \cdot N$, red robots are not able to sort themselves at all. For ratios of $n_r > 20\% \cdot N$, the sorting rate for red robots increases until reaching 100%.

In contrast to the red robots, the number of green robots reduces from $95\% \cdot N$ to $50\% \cdot N$. For all the cases $n_g > 50\% \cdot N$, they represent the majority, hence, they have a bigger probability to occupy one of the boundaries to sort themselves well. The case for both swarms having the same quantity ($n_r = n_g = 10$) has the same result as shown in Fig. 3b for $N = 20$: a 100% sorting rate.

5 Conclusion

Sorting by swarm robots is a complex task. Recent research in sorting by swarm robots relies often on sophisticated sensors and complex image processing methods. We have proposed a gossip-based sorting method based exclusively on local information without using any sophisticated sensors. Our method allows different robot swarms to sort themselves efficiently by using the mechanism of self-organization. Comparing to previous research, we use a smaller communication range and simpler sensors to achieve sorting. The relationship between system performance, swarm size, and different compositions of competing swarms are studied in this work.

We plan to extend this work to the actual task of sorting objects by considering the two kinds of robots as robots carrying different types of objects. Instead of having the robots stop at the black areas they drop the carried object, pick a new one, and try to find the appropriate black area for it. Our findings about the influence of ratios of robot types will be applied, for example, by trying to guarantee that both kinds of objects are carried by two robot groups of approximately equal size at all time. The influence of our assumption of robust communication will be investigated by simulating unreliable communication. This might introduce difficulties for the gossiping method which might require additional methods. We also plan the obvious follow-up work to implement this method on real robots.

References

[1] Deneubourg, J.L., Goss, S., Franks, N., Franks, A.S., Detrain, C., Chrétien, L.: The dynamics of collective sorting: robot-like ants and ant-like robots. In: Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats, Cambridge, MA, USA, MIT Press (1990) 356–363

[2] Franks, N., Sendova-Franks, A.: Brood sorting by ants: distributing the workload over the work-surface. Behavioral Ecology and Sociobiology **30**(2) (1992) 109–123

[3] Camazine, S.: Self-organizing pattern formation on the combs of honey bee colonies. Behavioral Ecology and Sociobiology **28**(1) (January 1991) 61–76

[4] Deneubourg, J., Fresneau, D., Gross, S., Lachaud, J., Pasteels, J.: Self-organization mechanisms in ant societies (ii): learning during foraging and division of labor. From Individual to Collective Behavior on Social Insects Basel, Boston (1987) 177–196

[5] Holland, O., Holl, O., Melhuish, C., Lane, C.: Stigmergy, self-organisation, and sorting in collective robotics. Artificial Life 5 173–202

[6] G. Barker, M.G.: The physics of muesli. New Scientist **126 (1718)** (1990) 37–40

[7] Handl, J., Knowles, J., Dorigo, M.: Design and Application of Hybrid Intelligent Systems. IOS Press, Amsterdam,The Netherlands (2003)

[8] Verret, S., Zhang, H., Meng, M.Q.H.: Collective sorting with local communication. In: IROS, IEEE (2004) 2687–2692

[9] Melhuish, C., Holland, O., Hoddell, S.: Collective sorting and segregation in robots with minimal sensing. 5th Int. Conf. on the Simulation of Adaptive Behaviour (1998)

[10] Maris, M., te Boekhorst, R.: Exploiting physical constraints: Heap formation through behavioral error in a group of robots. In: International Conference on Intelligent Robots and Systems. (1996) 1655–1660

[11] Melhuish, C., Wilson, M., Sendova-Franks, A.B.: Patch sorting: Multi-object clustering using minimalist robots. In: ECAL. (2001) 543–552

[12] Vardy, A.: Accelerated patch sorting by a robotic swarm. 9th Conference on Computer and Robot Vision (CRV) (2012)

[13] Pinciroli, C., Trianni, V.e.a.: Argos: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intelligence **6** (2012) 271–295

[14] Bonani, M., Longchamp, V., et al., M.: The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In: International Conference on Intelligent Robots and Systems. (2010) 4187–4193