Construction Task Allocation through the Collective Perception of a Dynamic Environment

Yara Khaluf[1][0000−0002−5590−9321], Michael Allwright[2][0000−0002−0932−3215], Ilja Rausch[1][0000−0002−9170−3021], Pieter Simoens[1][0000−0002−9569−9373], and Marco Dorigo[2][0000−0002−3971−0507]

[1] Ghent University–imec, Department of Information Technology, Gent, Belgium
{yara.khaluf,ilja.rausch,pieter.simoens}@ugent.be
[2] IRIDIA, Université Libre de Bruxelles, Brussels, Belgium
{michael.allwright,mdorigo}@ulb.ac.be

Abstract. Building structures is a remarkable collective process but its automation remains an open challenge. Robot swarms provide a promising solution to this challenge. However, collective construction involves a number of difficulties regarding efficient robots allocation to the different activities, particularly if the goal is to reach an optimal construction rate. In this paper, we study an abstract construction scenario, where a swarm of robots is engaged in a collective perception process to estimate the density of building blocks around a construction site. The goal of this perception process is to maintain a minimum density of blocks available to the robots for construction. To maintain this density, the allocation of robots to the foraging task needs to be adjusted such that enough blocks are retrieved. Our results show a robust collective perception that enables the swarm to maintain a minimum block density under different rates of construction and foraging. Our approach leads the system to stabilize around a state in which the robots allocation allows the swarm to maintain a tile density that is close to or above the target minimum.

1 Introduction

Building structures are among the most remarkable production processes we humans undertake. However, it is both costly and time-consuming. Consequently, integrating robots into construction processes can be of great benefit. Social insects provide important examples of collective behaviors such as creating large complex structures. These stem from simple behaviors of agents without centralized control or pre-planning. One prominent example is termites: millions of small insects successfully self-organize to build massive, complex mounds that sometimes exceed 12 m in height.

Inspired by natural swarms, researchers have started looking into swarm robotics systems that are able to construct increasingly complex structures [1,2,38]. Similar to social insects, construction by a robot swarm involves agents arranging building materials in an environment to form structures. To do so, usually robots coordinate through stigmergy. In contrast to direct communication, stigmergy

enables the coordination of the agents' activities through changing their shared
environment [32,5,8]. Upon sensing such environmental changes, the agents con-
clude their next activity. For example, the availability of building material pro-
vides a cue for robots, triggering their decision-making process. Nevertheless, in
some situations, stigmergy may not be a sufficient or appropriate means of com-
munication; in these cases, direct communication becomes necessary to exchange
particular pieces of information for successful task completion.

We base our study on the Swarm Robotics Construction System (SRoCS) [3],
a simulation of which is shown in Fig. 1(a). In this system, robots use computer
vision to monitor other robots' actions, and based on these observations, they
perform predefined construction actions that advance to complete a partially
built structure. In this paper, we consider a scenario where a swarm is divided
into two groups of robots. The first group is responsible for exploring the envi-
ronment, finding building material in form of building blocks, and transporting
it to the construction site. This behavior is referred to as the foraging task. The
second group is responsible for assembling the foraged blocks into a structure.
This behavior is referred to as the construction task. We consider two groups of
robots since some robots must remain in the cache area over extended periods
of time to estimate the tile density. This estimation is paramount to effectively
allocating the robots between the foraging and construction tasks.
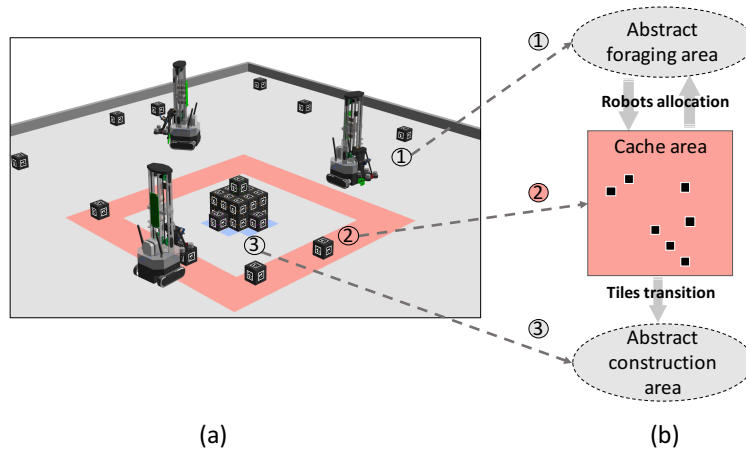
Fig. 1. A structure being built using the swarm robotics construction system: (a) in
the ARGoS simulator, and (b) an abstraction of the environment with the cache area
(pink), and building blocks (black).

In this study, we focus on the region surrounding the construction site where
blocks can be temporarily placed for construction. We call this the cache area
(pink in Fig. 1). In order for construction robots to perform their task efficiently,
there must be enough blocks in the cache area. Such blocks are discovered and

transported to the cache by the foraging robots. The goal in this study is to maintain an amount of blocks that maximizes the rate of construction. To achieve this, the swarm must find a suitable allocation between the robots adding blocks to the cache area (i.e., robots in the foraging task) and the robots removing blocks from the cache area (i.e., robots in the construction task) such that a certain amount of blocks is always preserved in the cache. To achieve this proper allocation, the construction robots need to collectively perceive the number of blocks in the cache area. This estimate helps the individual robots switch to foraging when the estimate drops below a given threshold, or to continue estimating otherwise. This collective perception process is highly challenging as it is performed in a dynamic environment, where the number of blocks in the cache area changes continuously. To address this challenge, we consider an abstraction of the construction scenario, where we can focus on the collective perception and decision-making dynamics involved in finding a suitable robots allocation between the two tasks. We use a homogeneous swarm, in which all robots are capable of performing either the construction task or the foraging task. The robots performing the construction task can communicate with each other if they are within range. Differently, robots performing the foraging task are unable to interact with the robots in the cache and vice versa. Once a robot switches to the foraging task, it always returns to the construction task with a new block after a period of time spent on foraging. Finally, the building material is assumed to be homogeneous and the maximum rate at which the building blocks can be attached to a structure is constant throughout the experiment.

The results of our collective perception process show that a swarm can collectively estimate and track the state of a dynamic environment, and use this information to find a suitable allocation between tasks. Furthermore, we show how parameters of the task and of the environment influence the accuracy of the collective perception process. Consequently, we discuss how our controller could be modified so that it is more resilient to these parameters. The remainder of this paper is organized as follows. In the next section, we provide a brief overview of the literature on collective perception and decision-making. In Section 3, we detail the robots' behavior and the abstract environment in which we evaluate it. In Section 4, we summarize and discuss the key results. We conclude this paper in Section 5. The data generated by this research is available online as a project on the Open Science Framework.[3] This project includes all software components and documentation required to reproduce and to extend this research [15].

2   Related Work

In robot swarms, collective decision-making has been extensively researched [10,12,21,22,25,27,28,29,33,34,37], due to its essential role in a wide range of tasks including foraging, flocking, and construction. In general, two types of system outputs are associated with a collective decision-making process: (i) a consensus in which the swarm agrees on a single option [35], and (ii) a division

---

of the swarm among different tasks—i.e., task allocation—[6,19,24,16]. The goal of the swarm in the consensus-achievement is to converge as quickly as possible on one option, referred to as a symmetry-breaking decision [11,18], or to converge on the best option as quickly as possible, referred to as a value-sensitive decision [23,9]. In the case of task allocation, the goal of the swarm is to maximize the system performance by minimizing the idle time of individual robots.

In this study, we tackle a specific process of collective decision-making that is referred to as collective perception. Collective perception is a process, in which the robots are engaged in perceiving an environmental stimulus collaboratively [30,36,14,31]. In general, robots need to perceive particular signals in their environments to make decisions based on the perceived values. We assume that the environments where robot swarms are deployed are large so that the perception of a single robot is far from sufficient to sense a system-wide stimulus—i.e., a stimulus that spreads across a large space of the environment. This motivates the need for collective perception, where robots combine their perceptions in a distributed manner, and self-organize to act as a single unit.

Collective perception is observed in social insects such as honeybees, where individuals tend to evaluate, for instance, queuing delays to optimize their task allocation [26]. Also bee foragers transfer their nectar load to multiple receivers suggesting the use of this behavior to estimate the environmental nectar flow [13]. These studies inspired collective perception in artificial systems such as robot swarms. For example, authors in [30] developed a bio-inspired algorithm enabling a robot swarm to aggregate at two locations, where the size of each group corresponds to the size of the selected location. In [36], the authors use a robot swarm to collectively decide which of two colors is the most represented in a pattern drawn on arena ground. The algorithm developed in [36] was tested for benchmarking and generalization in [4] across a larger number of patterns (nine). Contrary to [36], the authors in [4] find that the difficulty of the collective perception process doesn't depend mainly on the ratio of one color to the other, but on the distribution of each color in the environment. The authors in [7] proposed a distributed Bayesian algorithm to solve the collective perception task of a similar two-color environment. They define the speed vs. accuracy trade-off of the collective perception as a multi-objective optimization problem. Additionally, the authors have shown that it is possible to guarantee the accuracy of the collective perception, at the cost of decision time.

None of the aforementioned algorithms, however, support collective perception in a dynamic environment, where the perceived features change over time. Dynamic environments impose a serious challenge to collective perception algorithms, i.e. the rate at which the swarm reaches a consensus vs. the rate at which the environment changes. Also, contrary to other algorithms, our collective perception algorithm attempts to estimate absolute values of the perceived feature (e.g., the percentage of a particular color), instead of merely providing a decision on its relative properties (e.g., color x is represented more than color y).

3   The Model

We approach the collective construction problem using an abstract model, as depicted in Fig. 1(b). In the abstract model, we focus on the cache area, in which building blocks are modeled as 2D tiles. These tiles are moved from the foraging area to the cache area by robots performing the foraging task. Robots that are allocated to the construction task, explore the cache area and pick up tiles (when encountered) and use these to build a structure. We omit the details of the foraging process, and instead we replace it by a stochastic process that characterizes the retrieval of tiles. We define a lower-bound density of tiles in the cache area that we call the target density Γ. This density enables us to minimize the idle time of constructing robots—i.e., to maximize the construction rate. We assume the density Γ to be known and provided to the swarm. The goal of the swarm is to allocate the robots to the foraging and construction tasks so that Γ is satisfied and maintained in the cache area.

3.1   The Retrieval Process of Tiles

We model the output of the foraging process—i.e., the retrieval of tiles—using a renewal process. This is a sequence of random variables, which are referred to as the arrival times, at which a repeating event occurs—i.e., retrieving a tile. The inter-arrival time is the period between two consecutive events, these in our study are two consecutive retrieval of tiles. We model these inter-arrival times by sampling from an exponential distribution with the density function:

$$f_T = \frac{1}{\lambda_f} e^{-\frac{1}{\lambda_f} T}, \tag{1}$$

where the parameter λ_f is the average time a robot spent foraging before returning to the cache area with a new tile. Modeling the inter-arrival times to be exponentially distributed results in the renewal process to be a Poisson process, a common way to model arrival events [39,20,17]; therefore, the average number of tiles retrieved by foraging robots within the time period δt is given by:

$$\langle M_i(\delta t) \rangle = \left\lfloor \frac{\delta t}{\lambda_f} \right\rfloor N_{f \rightarrow c}(\delta t), \tag{2}$$

where $N_{f \rightarrow c}(\delta t)$ is the number of robots switching from foraging to construction in the time interval δt. The value of this variable changes over time as a function of the number of robots in the cache and the estimated and target tile density.

3.2   The Simulated Environment

To evaluate our collective perception process, we use the ARGoS simulator to create a 4×4 m^2 arena that is divided into 2500 tiles. We run experiments with 80 robots that can drive around the arena, avoid obstacles such as walls and each other, sense whether or not they are driving over a shaded tile, and communicate

with each other over wifi. We restrict the communication distance of the robot to be a maximum of one meter to prevent global communication in the swarm.

We simulate a robot switching to the foraging task by removing it from the simulation and setting up a countdown timer that is initialized following Eq. (1). The mean of this distribution λ_f in Eq. (1) is one of the key parameters which we vary in our experiments. Once the timer reaches zero, we simulate the robot switching back to the construction task by adding it back into the simulation. Using this strategy, the foraging robots are unable to communicate with the robots performing the construction task, which is consistent with our scenario.

When a robot returns from the foraging task, a cell in the arena is shaded to represent the tile this robot retrieved. We follow a probabilistic approach to place the retrieved tile in the cache. Our approach has the effect of creating clusters of tiles in the cache area (see Fig. 2). This is achieved by increasing the probability to select a candidate location x by a factor of $\kappa = 5$ for each tile that is adjacent (max. 8 tiles) to the location x. Hence candidate locations with more tiles in the neighborhood have a higher probability to be selected. Performing construction—i.e. removing a block from the cache and attaching it to a hypothetical structure—is simulated by a tile being unshaded. This transition occurs whenever a robot moves off a tile and onto another tile. The maximum number of tiles that can be unshaded per second is the construction limit ξ_c, which is another key parameter that we vary in our experiments. A full simulation run with default parameters is hosted on Open Science Framework.[4]
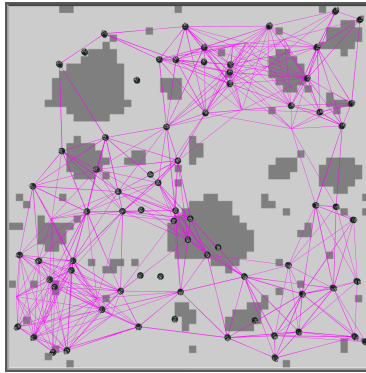
Fig. 2. Screenshot of a simulation in the cache that shows the tiles clustering effect. The magenta lines represent the communication links between the robots.

3.3   The Robot Behavior

A robot moves through the cache in a straight line unless it encounters an obstacle. In case of an obstacle, the robot turns on the spot until its heading is

---

[4] Complete run of a simulation (video): https://osf.io/6mgys/

clear. When the robot is not avoiding obstacles, it samples the ground beneath it to determine whether it is on top of a tile. The robot keeps track of the total number of samples it has taken and the number of times a sample was taken on a tile. In addition to these counts, the robot's memory also contains a table that records these counts received from the robot's neighbors. An entry in this table contains three fields: (i) the neighbor's total number of samples, (ii) the neighbor's number of samples taken over a tile, and (iii) a time-to-live value that is used to drop the neighbor's entry from the table when its value reaches zero.

At each time step, a robot adjusts the table in its memory by decrementing the time-to-live field for each entry. It then sends this table with an additional entry, representing its sample counts, to all of the robot's direct neighbors. The time-to-live value in this additional entry is initialized to its maximum value. When a direct neighbor receives this information, it updates its table by replacing the contents of each neighbor's entry with the one with the highest time-to-live value found among the existing entries in its table and the entries from the received messages. In this way, each robot always has the most up-to-date entry for each robot that it has recently communicated with. This time-to-live value is also used to avoid loops and to prevent duplicate entries in a robot's table.

After a robot has updated its table, it uses both its local sample counts and the sample counts from its neighbors to estimate the tile density. To compute this estimate $\langle \gamma_i(t) \rangle$ the robot i constructs a weighted average, in which the contribution of each entry (neighbor j's information) is weighted by (i) the number of samples that neighbor has taken and (ii) the hop distance of that neighbor (calculated from the time-to-live field). This average includes i's own sample counts weighted by the number of samples i took and a hop distance of one.

$$\langle \gamma_i(t) \rangle = \frac{\omega_i(t)\gamma_i(t) + \sum_{j \in N_i} \omega_j(t)\gamma_j(t)}{\omega_i(t) + \sum_{j \in N_i} \omega_j(t)}, \tag{3}$$

where N_i is the set of robot i's direct neighbors, and $\gamma_i(t)$ is the tile density measured locally by robot i and defined as:

$$\gamma_i(t) = \frac{c_i(t)}{s_i(t)} \tag{4}$$

where $c_i(t)$ is the number of samples taken by robot i while driving over tiles, and $s_i(t)$ is the total number of samples taken by robot i. The weight ω_i assigned to the locally-measured density by robot i in Eq. (3) is defined as:

$$\omega_i(t) = s_i(t)h_{ij}(t) \tag{5}$$

where $h_{ij}(t)$ is the hop distance between robot i and robot j. The weighted average increases the influence of robots that have sampled larger areas of the arena and that are further away. The latter weighting makes the swarm more resilient against over-estimating the tile density which would otherwise occur due to the clustering of tiles.

After estimating the density of tiles, each robot i decides probabilistically to switch from the construction to the foraging task, as long as $\langle \gamma_i(t) \rangle$ is lower than

the target density Γ. The switching probability $Pr_i^{c \to f}(t)$ of robot i at time step t is proportional to the difference between robot i's estimate and Γ:

$$Pr_i^{c \to f}(t) = \begin{cases} \eta|\Gamma - \langle\gamma_i(t)\rangle| & \text{if}\langle\gamma_i(t)\rangle < \Gamma \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where η is a design parameter used to keep $Pr_i^{c \to f}(t)$ in the interval $[0,1]$.

4   Results and Discussion

We have investigated the performance of the collective perception process as well as the robots' task allocation for different experiment configurations. Specifically, our results were obtained over the following set of parameters: (i) the mean foraging time λ_f (tested values $\lambda_f \in \{5, 10, 20\}$), (ii) the lower-bound of the target density Γ (tested values $\Gamma \in \{0.1, 0.3, 0.4\}$), (iii) the construction limit ξ_c (tested values $\xi \in \{5, 10, 20\}$), and (iv) the constant η in the switching probability as defined in Eq. (6) (tested values $\eta \in \{0.1, 0.15, 0.2\}$). We have published the data from these experiments online [15]. The evaluation of our approach spans over four metrics. The first is the time trajectory of the density of tiles $\rho_{gt}(t)$, which is the ground truth; the second is the time trajectory of the swarm estimate $\rho_s(t)$; the third is the time evolution of the individual deviation from the swarm estimate of the tile density: $\Delta_i(t) = |\langle\gamma_i(t)\rangle - \rho_s(t)|$; and the fourth illustrates how the robots allocation to construction and foraging evolves over time. We run all experiments for $2\,500$ seconds ($12\,500$ time steps) with a swarm size of 80 robots and average the results of each experiment across 30 runs. In the following we discuss our findings over a subset of the parameters' tested values.

Let us start with the first metric, tile density. Fig. 3 shows that the swarm was able to increase the tile density in the cache area and keep it above the target density Γ for $\Gamma \in \{0.3, 0.4\}$. The swarm estimate is initially in full agreement with the ground truth as both start at 0 tiles. Over time, the swarm estimate $\rho_s(t)$ stabilizes around the target density Γ, with a minority of robots (see the standard deviation) estimating the tile density to be higher than Γ. This minority acts to reduce the number of robots sent to retrieve tiles, while the majority acts to increase this number, leading the ground truth to a value higher than the target density. The interplay of these two groups in the swarm causes both the swarm estimate and ground truth to stabilize with the ground truth higher than the swarm estimate. We also notice that increasing the mean foraging time ($\lambda_f \in \{10, 20\}$) leads to a slower increase in $\rho_{gt}(t)$. This is because higher values of λ_f imply longer periods between the retrieval of tiles, on average. Furthermore, we see that for a specific target density Γ, $\rho_s(t)$ seems to be maintained across different values of λ_f and ξ_c. This suggests that the collective perception process is relatively robust to both parameters. This robustness implies that our algorithm is suitable for real-world construction tasks, where complicated foraging that takes longer to find building materials, or prolonged assembly of building material does not affect the collective perception performance.
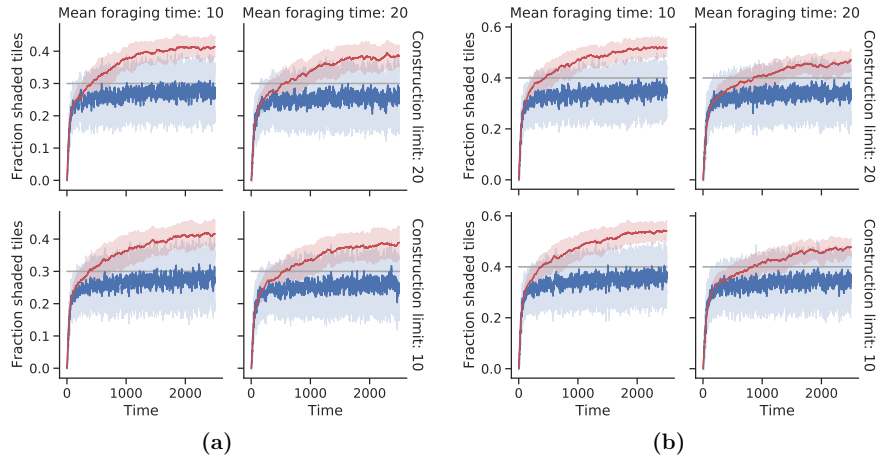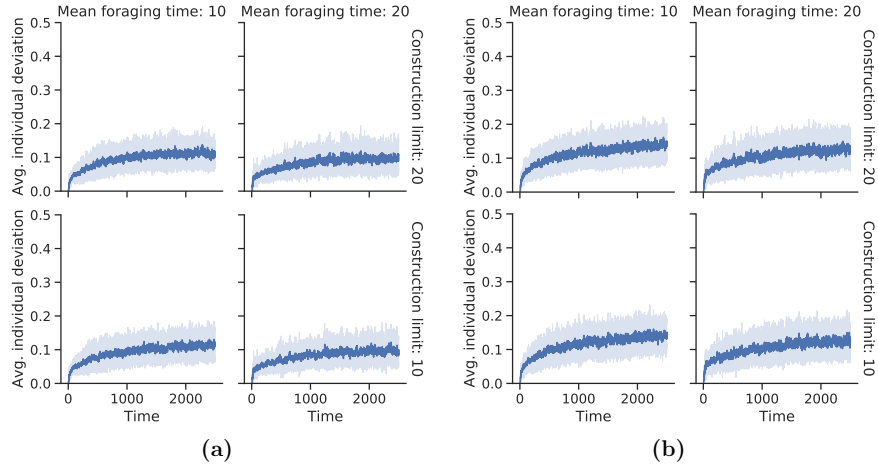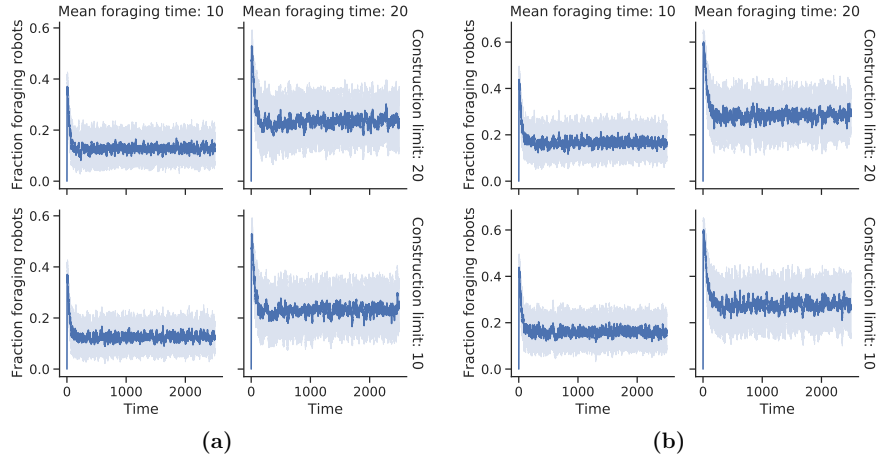
Fig. 3. Tile density for $\eta = 0.2$ and target density (a) $\Gamma = 0.3$ and (b) $\Gamma = 0.4$: ■ swarm estimate $\rho_s(t)$, ■ ground truth $\rho_{gt}(t)$ (averaged across 30 runs).

Fig. 4 illustrates the mean individual deviation $\langle \Delta_i(t) \rangle$ from the swarm estimate for different target densities $\Gamma \in \{0.3, 0.4\}$, mean foraging times $\lambda_f \in \{10, 20\}$, and construction limits $\xi_c \in \{10, 20\}$. Our results show robustness of the individual deviation with regard to changes in these parameters, with a maximum average deviation of 0.15 ($\langle \Delta_i(t) \rangle \leq 0.15$). Such a small variance $\langle \Delta_i(t) \rangle$ indicates a strong agreement between the individual and the group estimate.

Finally, Fig. 5 shows the fraction of robots allocated to foraging over time. At the beginning, all 80 robots are in the cache area for all experiment configurations. Thus, initially, there is a jump in the number of robots leaving from the construction task to the foraging task. This jump is due to the low swarm estimate $\rho_s(t)$ of the tile density during the first 100 seconds of the experiments (see Fig. 3). During this initial period, the estimate $\rho_s(t)$ of the swarm remains consistently below the target density. Hence the condition $\langle \gamma_i(t) \rangle < \Gamma$ is true for a large majority of the robots, that then switch to the foraging task following Eq. (6) with a relatively high probability. The magnitude of the spike increases with the mean foraging time λ_f. This is due to the longer time it takes for the robots to arrive back from the foraging area, and thus for the tile density and the swarm estimate to rise. Nevertheless, as soon as the swarm estimate stabilizes and the ground truth of tiles in the cache reaches or exceeds its target $\rho_{gt}(t) \geq \Gamma$, the fraction of the foraging robots starts to drop until it stabilizes, leading the system into an equilibrium state with respect to the robots allocation. The drop takes longer time for larger λ_f. Furthermore, the fraction of robots that continue foraging is higher for higher Γ values. This is because tiles need to be retrieved at a faster rate. Additionally, the fraction of foraging robots is higher for larger λ_f given the same Γ. This is due to the slower rate of tile retrieval when λ_f is

Fig. 4. Average individual deviation $\langle \Delta_i(t) \rangle$ from swarm estimate $\rho_s(t)$ for $\eta = 0.2$ and target density (a) $\Gamma = 0.3$ and (b) $\Gamma = 0.4$ (averaged across 30 runs).

Fig. 5. Fraction of foraging robots for $\eta = 0.2$ and target density (a) $\Gamma = 0.3$ and (b) $\Gamma = 0.4$ (averaged across 30 runs).

larger. Thus, this slower rate pushes more robots to foraging while the robots that remain in the cache are performing construction and estimating the tile density $\langle \gamma_i(t) \rangle$ for a longer time.

5   Conclusions

We studied an abstract scenario of a collective construction process by a robot swarm in a dynamic environment. In this scenario, we focused on the cache area—i.e., an area that surrounds the construction area—, where blocks are modeled as tiles that appear through foraging and disappear through construction. Robots can switch between two tasks: foraging and construction. Foraging robots explore the foraging area and retrieve the building material, while construction robots conduct a collective perception process to maintain a minimum tile density in the cache. The details of the foraging task are abstracted away and it is modeled using a Poisson process which delivers tiles with a specific rate $(1/\lambda_f)$ to the cache. This enabled us to focus on research questions concerning the design of a task allocation mechanism that exploits collective perception in a dynamic environment. In future work, we plan to simulate a detailed foraging process. The collective perception process aims to assign robots to the foraging task to increase the tiles retrieval rate whenever the density in the cache drops below the target density Γ—i.e., the required lower-bound on the tile density. Robots in the cache rely on both their samples and the samples of their neighbors to compute an estimate of the tile density. This estimate is computed as a weighted average that assigns higher importance to (i) robots that are further away, making the swarm more resilient to over-estimation or under-estimation due to the clustering of tiles; (ii) robots with larger samples, as these contributions are more representative. Robots use their estimate from Eq. (3) to probabilistically decide whether to switch to the foraging task or to continue estimating/performing the construction task.

Our results show that the proposed collective perception process leads to a proper robots allocation, which in turn guarantees a minimum tile density in the cache. This allocation changes as a function of the average time it takes a robot to find and retrieve a tile λ_f, and the target density Γ. Furthermore, our results show a strong agreement between the individual estimate $\langle \gamma_i(t) \rangle$ and the swarm estimate $\rho_s(t)$, with a maximum variance of 0.15. This study is a first step towards designing collective perception processes in dynamic environments, in which the perceived feature (e.g., tile density) changes over time and the goal is to estimate its absolute value. In future work, we plan to study the influence of the size and update rate of the robots' memorized samples on the performance of the perception process. We also intend to extend the proposed algorithm to enable the swarm to maintain a tile density close to its target.

References

1. Allwright, M., Bhalla, N., Dorigo, M.: Structure and markings as stimuli for autonomous construction. In: Eighteenth International Conference on Advanced Robotics – ICAR 2017. pp. 296–302. IEEE Press, Piscataway, NJ (2017)
2. Allwright, M., Bhalla, N., Pinciroli, C., Dorigo, M.: Simulating multi-robot construction in ARGoS. In: Swarm Intelligence – Proceedings of ANTS 2018 – Eleventh International Conference. Lecture Notes in Computer Science, vol. 11172, pp. 188–200. Springer, Berlin, Germany (2018)
3. Allwright, M., Zhu, W., Dorigo, M.: An open-source multi-robot construction system. HardwareX **5**, e00049 (2019)
4. Bartashevich, P., Mostaghim, S.: Benchmarking collective perception: New task difficulty metrics for collective decision-making. In: EPIA Conference on Artificial Intelligence. pp. 699–711. Springer (2019)
5. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
6. Brutschy, A., Pini, G., Pinciroli, C., Birattari, M., Dorigo, M.: Self-organized task allocation to sequentially interdependent tasks in swarm robotics. Autonomous Agents and Multi-Agent Systems **28**(1), 101–125 (2014)
7. Ebert, J., Gauci, M., Mallmann-Trenn, F., Nagpal, R.: Bayes Bots: Collective bayesian decision-making in decentralized robot swarms. Intl. Conference on Robotics and Automation (ICRA) (2020)
8. Garnier, S., Gautrais, J., Theraulaz, G.: The biological principles of swarm intelligence. Swarm Intelligence **1**(1), 3–31 (2007)
9. Gray, R., Franci, A., Srivastava, V., Leonard, N.E.: Multiagent decision-making dynamics inspired by honeybees. IEEE Transactions on Control of Network Systems **5**(2), 793–806 (2018)
10. Gutiérrez, A., Campo, A., Monasterio-Huelin, F., Magdalena, L., Dorigo, M.: Collective decision-making based on social odometry. Neural Computing and Applications **19**(6), 807–823 (2010)
11. Hamann, H., Schmickl, T., Wörn, H., Crailsheim, K.: Analysis of emergent symmetry breaking in collective decision making. Neural Computing and Applications **21**(2), 207–218 (2012)
12. Hamann, H., Valentini, G., Khaluf, Y., Dorigo, M.: Derivation of a micro-macro link for collective decision-making systems. In: International Conference on Parallel Problem Solving from Nature. pp. 181–190. Springer (2014)
13. Huang, M., Seeley, T.: Multiple unloadings by nectar foragers in honey bees: a matter of information improvement or crop fullness? Insectes Sociaux **50**(4), 330–339 (2003)
14. Khaluf, Y.: Edge detection in static and dynamic environments using robot swarms. In: 2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO). pp. 81–90. IEEE (2017)
15. Khaluf, Y., Allwright, M., Rausch, I., Simoens, P., Dorigo, M.: Construction task allocation through the collective perception of a dynamic environment (2020), https://osf.io/n7kr3/
16. Khaluf, Y., Birattari, M., Hamann, H.: A swarm robotics approach to task allocation under soft deadlines and negligible switching costs. In: International Conference on Simulation of Adaptive Behavior. pp. 270–279. Springer (2014)
17. Khaluf, Y., Dorigo, M.: Modeling robot swarms using integrals of birth-death processes. ACM Transactions on Autonomous and Adaptive Systems (TAAS) **11**(2), 1–16 (2016)

18. Khaluf, Y., Pinciroli, C., Valentini, G., Hamann, H.: The impact of agent density on scalability in collective systems: noise-induced versus majority-based bistability. Swarm Intelligence **11**(2), 155–179 (2017)
19. Khaluf, Y., Rammig, F.: Task allocation strategy for time-constrained tasks in robots swarms. In: Artificial Life Conference Proceedings 13. pp. 737–744. MIT Press (2013)
20. Kim, S.H., Whitt, W.: Choosing arrival process models for service systems: Tests of a nonhomogeneous poisson process. Naval Research Logistics (NRL) **61**(1), 66–90 (2014)
21. Meyer, B.: Optimal information transfer and stochastic resonance in collective decision making. Swarm Intelligence **11**(2), 131–154 (2017)
22. Montes de Oca, M.A., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., Dorigo, M.: Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. Swarm Intelligence **5**(3–4), 305–327 (2011)
23. Pais, D., Hogan, P.M., Schlegel, T., Franks, N.R., Leonard, N.E., Marshall, J.A.: A mechanism for value-sensitive decision-making. PloS one **8**(9) (2013)
24. Pini, G., Brutschy, A., Frison, M., Roli, A., Dorigo, M., Birattari, M.: Task partitioning in swarms of robots: An adaptive method for strategy selection. Swarm Intelligence **5**(3-4), 283–304 (2011)
25. Prasetyo, J., Masi, G.D., Ferrante, E.: Collective decision making in dynamic environments. Swarm Intelligence **13**(3–4), 217–243 (2019)
26. Ratnieks, F.L., Anderson, C.: Task partitioning in insect societies. ii. use of queueing delay information in recruitment. The American Naturalist **154**(5), 536–548 (1999)
27. Rausch, I., Khaluf, Y., Simoens, P.: Collective decision-making on triadic graphs. In: Complex Networks XI, pp. 119–130. Springer (2020)
28. Reina, A., Miletitch, R., Dorigo, M., Trianni, V.: A quantitative micro–macro link for collective decisions: the shortest path discovery/selection example. Swarm Intelligence **9**(2–3), 75–102 (2015)
29. Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., Trianni, V.: A design pattern for decentralised decision making. PloS one **10**(10) (2015)
30. Schmickl, T., Möslinger, C., Crailsheim, K.: Collective perception in a robot swarm. In: International Workshop on Swarm Robotics. pp. 144–157. Springer (2006)
31. Strobel, V., Castelló Ferrer, E., Dorigo, M.: Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario. In: Dastani, M., Sukthankar, G., André, E., Koenig, S. (eds.) Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems. pp. 541–549. International Foundation for Autonomous Agents and Multiagent Systems (2018)
32. Theraulaz, G., Bonabeau, E.: A brief history of stigmergy. Artificial life **5**(2), 97–116 (1999)
33. Trianni, V., De Simone, D., Reina, A., Baronchelli, A.: Emergence of consensus in a multi-robot network: from abstract models to empirical validation. IEEE Robotics and Automation Letters **1**(1), 348–353 (2016)
34. Valentini, G., Hamann, H.: Time-variant feedback processes in collective decision-making systems: influence and effect of dynamic neighborhood sizes. Swarm Intelligence **9**(2–3), 153–176 (2015)
35. Valentini, G.: Achieving Consensus in Robot Swarms. Springer (2017)

36. Valentini, G., Brambilla, D., Hamann, H., Dorigo, M.: Collective perception of environmental features in a robot swarm. In: International Conference on Swarm Intelligence. pp. 65–76. Springer (2016)
37. Valentini, G., Ferrante, E., Hamann, H., Dorigo, M.: Collective decision with 100 kilobots: Speed versus accuracy in binary discrimination problems. Autonomous Agents and Multi-Agent Systems **30**(3), 553–580 (2016)
38. Werfel, J., Petersen, K., Nagpal, R.: Designing collective behavior in a termite-inspired robot construction team. Science **343**(6172), 754–758 (2014)
39. Wolff, R.W.: Poisson arrivals see time averages. Operations Research **30**(2), 223–231 (1982)