Evolving Collective Cognition of Robotic Swarms in the Foraging Task with Poison

Motoaki Hiraga
Graduate School of Engineering
Hiroshima University
Higashi-Hiroshima 739-8527, Japan
hiraga@ohk.hiroshima-u.ac.jp

Yufei Wei
Graduate School of Engineering
Hiroshima University
Higashi-Hiroshima 739-8527, Japan
wei@ohk.hiroshima-u.ac.jp

Kazuhiro Ohkura
Graduate School of Engineering
Hiroshima University
Higashi-Hiroshima 739-8527, Japan
kohkura@hiroshima-u.ac.jp

***Abstract*—This paper focuses on the collective cognition of robotic swarms. The robotic swarms perform tasks that are beyond the capability of a single robot by collective behavior that emerge from local interactions. However, due to the lack of the capability of a single robot, it is necessary for robotic swarms to collectively perceive the environment. In this paper, we develop controllers for a robotic swarm to accomplish a complex cognitive task, namely the collective foraging task with poison. In this task, robots have to both collectively distinguish two objects, namely foods and poisons, and cooperatively transport foods to the nest. We applied an evolutionary robotics approach with covariance matrix adaptation evolution strategy to develop controllers for robotic swarms. The results of computer simulations show that collective cognition behavior was successfully generated, which allows the robots to transport only foods. In addition, we also perform experiments to examine the scalability of the developed controllers.**

I. Introduction

Swarm robotics is the study of a large group of autonomous robots that emerges collective behavior without relying on any global information [1], [2]. It takes inspiration from biological swarms, such as colonies of ants and termites, flocks of birds, and schools of fish, which perform a collective behavior that is beyond the capability of a single individual. Similar to biological swarms, robots in a robotic swarm are relatively simple compared to the task they are dealing with, that is, their communication is usually performed by local interactions and only the limited sensory capabilities are available. However, they can achieve complex tasks which are beyond the ability of a single robot by collective behaviors that emerged from local interactions among the robots and between the robots and the environment. Moreover, a robotic swarm shows three advantage properties; (i) robustness for operating despite the loss of robots and other disturbances in the environment, (ii) flexibility for generating modularized solutions for different tasks, and (iii) scalability for operating with a wide range of group sizes.

This paper focuses on the collective cognition of robotic swarms. The biological swarms are able to perceive the environment in a collective manner which leads to sophisticated collective behavior [3], [4]. However, in robotic swarms, due to the lack of the capability of a single robot, the robotic swarms have to rely on the collective perception more than biological swarms. Despite the importance of the collective cognition, only a few works have addressed this problem [5], [6]. Moreover, these works focus on the relatively simple collective cognition problem, i.e., a robotic swarm is to determine which feature cover the most in the environment.

In this paper, we develop controllers for a robotic swarm to accomplish a complex cognitive task, namely *the collective foraging task with poison*. In this task, the robots have to distinguish between foods and poisons scattered in the field and collectively transport the foods back the nest. It is worth noting that a single robot cannot distinguish between foods and poisons. We applied an evolutionary robotics approach [7], which is a technique to design controllers via an evolutionary algorithm. As an evolutionary algorithm, *covariance matrix adaptation evolution strategy* [8], [9] is adopted to develop controllers for a robotic swarm. We also perform computer simulations to examine the scalability of the developed controllers.

The remainder of this paper is organized as follows. The evolutionary robotics approach and the covariance matrix adaptation evolution strategy algorithm is introduced in Section II. Section III describes the collective foraging task with poison and the experimental setup. Section IV discusses the computer simulation results. Finally, we conclude this paper in Section V.

II. Related Work

A. Evolutionary Robotics Approach

Designing controllers for a robotic swarm is a challenging task. The difficulty resides in the fact that the relationship between simple local rules and complex swarm behaviors is indirect. The design methods can be classified into two types [10], namely behavior-based design methods and automatic design methods. In behavior-based design methods, individual-level behavior is designed manually by trial and error until expected collective behaviors are acquired. However, the design process is guided only by experience and intuition, which requires expertise in the undertaken task. In automatic design methods, by contrast, rather than designing each robot manually, the behavior of robots is generated automatically by transforming the design problem into an optimization problem to reduce human intervention [11].

One widely used automatic design method is evolutionary robotics [7], [12]. In evolutionary robotics, the controller is

978-1-7281-2153-6/19/$31.00 ©2019 IEEE

conventionally represented by a single artificial neural network, in which synaptic weights are optimized with an evolutionary algorithm. An evolutionary algorithm evaluates and optimizes controllers based on a predefined fitness function that indicates the performance of the robots. Evolutionary robotics approaches have succeeded in generating a wide range of robotic swarm behaviors, such as aggregation [13], [14], flocking [15], path formation [16], [17], and cooperative transport [18]. In this paper, we adopted the covariance matrix adaptation evolution strategy [8], [9] as an evolutionary algorithm to develop controllers for a robotic swarm.

B. Covariance Matrix Adaptation Evolution Strategy

The covariance matrix adaptation evolution strategy is widely known as an efficient population-based black-box optimization algorithm, in which each candidate solution is represented by a real-valued vector. While the offspring in the next generation are sampled normally distributed in classical evolution strategies, the covariance matrix adaptation evolution strategy samples offspring the following equation:

$$\mathbf{x}_i \sim \mathbf{m} + \sigma\mathcal{N}_i(0, \mathbf{C}), \qquad \text{for } i = 1, 2, ..., \lambda, \tag{1}$$

where $\mathbf{m} \in \mathbb{R}^n$ is the mean vector (search point) in the last generation, $\sigma \in \mathbb{R}_+$ is the so-called step size, and $\mathbf{C} \in \mathbb{R}^{n\times n}$ is the covariance matrix which determines the shape of the distribution ellipsoid. Here i is the index of candidate solutions, n represent the dimension of each candidate solution, and λ is the population size.

Compared to classical evolution strategies algorithms, the covariance matrix adaptation evolution strategy provides more efficiency by using covariance matrix adaptation and cumulation on the evolution path. The covariance matrix adaptation increases the likelihood of successful steps, which, in another point of view, follows a natural gradient approximation of the expected fitness. The adaptation has five characteristics; (i) it learns all pairwise dependencies between variables, (ii) a principle component analysis is performed sequentially in time and space, (iii) learns a new and rotated problem representation, (iv) approximates the inverse Hessian on quadratic functions, and (v) approximates the inverse Hessian on quadratic functions. On the other hand, the cumulation is a widely used technique (e.g. in back-propagation algorithms for artificial neural networks) which construct the evolution path in a recursive way, which utilizes history information. Additionally, using covariance matrix adaptation and cumulation on the evolution path allows the algorithm to search more efficiently in highly correlated search space. More details about the calculation of the covariance matrix and the cumulation can be found in [8], [9].

III. Experiments

A. Collective Foraging Task with Poison

The objective of the task is that the robots should distinguish between foods and poisons collectively, and transport only foods to the nest. Additionally, foods and poisons are too heavy for a single robot to move, therefore robots have to cooperate with each other to accomplish the task.

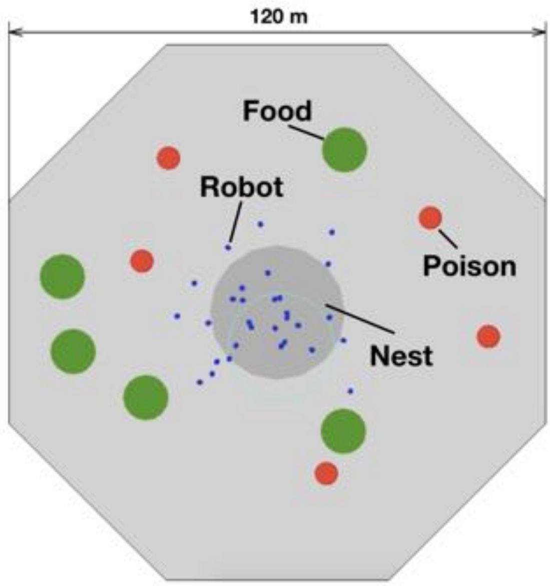


Fig. 1: The collective foraging task with poison.

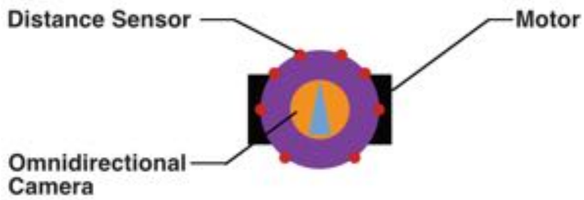


Fig. 2: The specifications of the robot.

The simulation environment has a regular octagon field, as shown in Fig. 1. A circle-shaped nest with a radius of 15 m is located in the center of the field. At the beginning of the simulations, robots are positioned inside the nest. The initial positions of the foods and poisons are randomly determined. There are always five foods and five poisons in the field, that is, when food or poison is transported to the nest, a new one will be generated with a random position. The radius of the food is set to 5.0 m and 2.5 m for the poison. The density of foods and poisons is set to have the same weight.

B. Settings of the Robot

The specifications of the robot are shown in Fig. 2. Each robot is composed of eight distance sensors, an omnidirectional camera, an artificial neural network controller, and two motors to rotate the left and right wheels. The range of the distance sensor is set to 3.0 m, detecting the distance between the nearest objects, robots, or walls. The range of the omnidirectional camera is set to 15 m, gathering distance and angle (relative angle to the robot, represented by $\sin\theta$ and $\cos\theta$, henceforth) information of the followings:

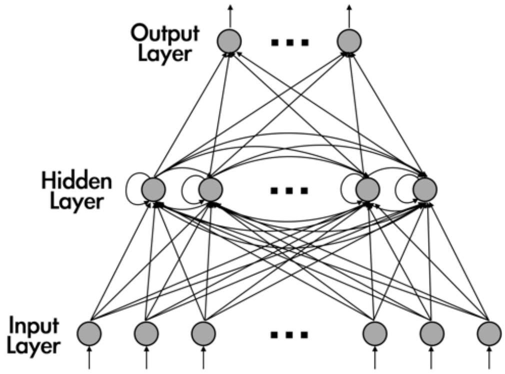


Fig. 3: Structure of the robot controller. The controller is represented by the fully-connected recurrent neural network. The nodes in the hidden layer are connected with each other and also have self-connections.

- The nearest food or poison.
- The second nearest food or poison.
- The nearest robot.
- The second nearest robot.
- Angle of the nest (without distance)

It is important to note that a single robot cannot distinguish between foods and poisons, because the omnidirectional camera treats foods and poisons as the same "objects". Each robot is governed by a three-layered neural network as shown in Fig. 3. The first layer gathers sensory inputs, which are sent to the hidden layer directly without any thresholds. The hidden layer is composed of twenty nodes with recurrent connections including self-connections. Outputs of the third layer control the two motors directly. The logistic function is adopted as the activation function for the hidden and the output layer.

C. Settings of Evolutionary Robotics Approach

The covariance matrix adaptation evolution strategy is employed to optimize the synaptic weights of the controller. At the first generation of artificial evolution, a population of λ candidates is initialized with random values. Each candidate controller is copied to N robots and evaluated for 10 times to get average performance in the collective foraging task with poison. The fitness function f is composed of two parts, a bonus for transporting foods and a penalty for poisons, which is defined as follows:

$$f = \sum_i d_{\mathrm{food},i} - \sum_j d_{\mathrm{poison},j} \tag{2}$$

where $d_{\mathrm{food},i}$ is distance shortened between the ith food and the center of the nest and $d_{\mathrm{poison},j}$ is distance shortened between the jth poison and the nest. Simulation for the evaluation of each candidate controller lasts for 9000 time-steps (0.02 s for each time-step, in total 180 s). The candidates with higher fitness are selected to produce the next population. This process is repeated until the maximum generation. In this paper, the population size of the covariance matrix adaptation evolution strategy algorithm is set to $\lambda = 300$ and the maximum generation is set to 2000.

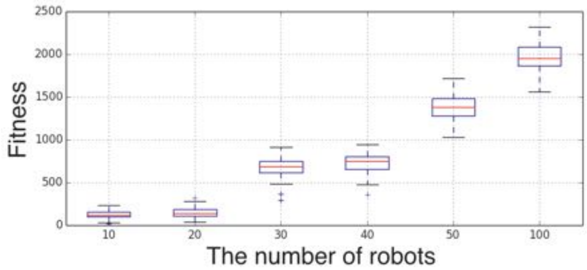


Fig. 4: The results of the re-evaluation with the best controller developed in each experiment with a different number of robots.

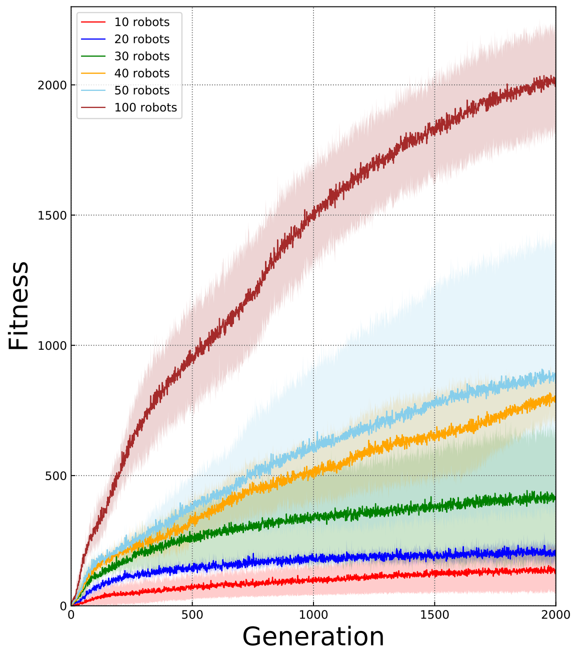


Fig. 5: The trajectories of the fitness for five trials with standard variation. The color show the experiment settings with different number of robots.

IV. Results and Discussion

The experiments are performed with N = 10, 20, 30, 40, 50, and 100 robots. We run five trials for each experiment. We further examine the scalability of the developed controllers. In scalability experiments, the performance of the controllers is tested in different numbers of robots as they are developed. For instance, the controller developed in the environment with 10 robots is also examined in environments with 20, 30, 40, 50, and 100 robots.

The performance of the best controller developed in each experiment is re-evaluated 50 times. The results of the re-evaluation are shown in Fig. 4. We further plot the trajectories

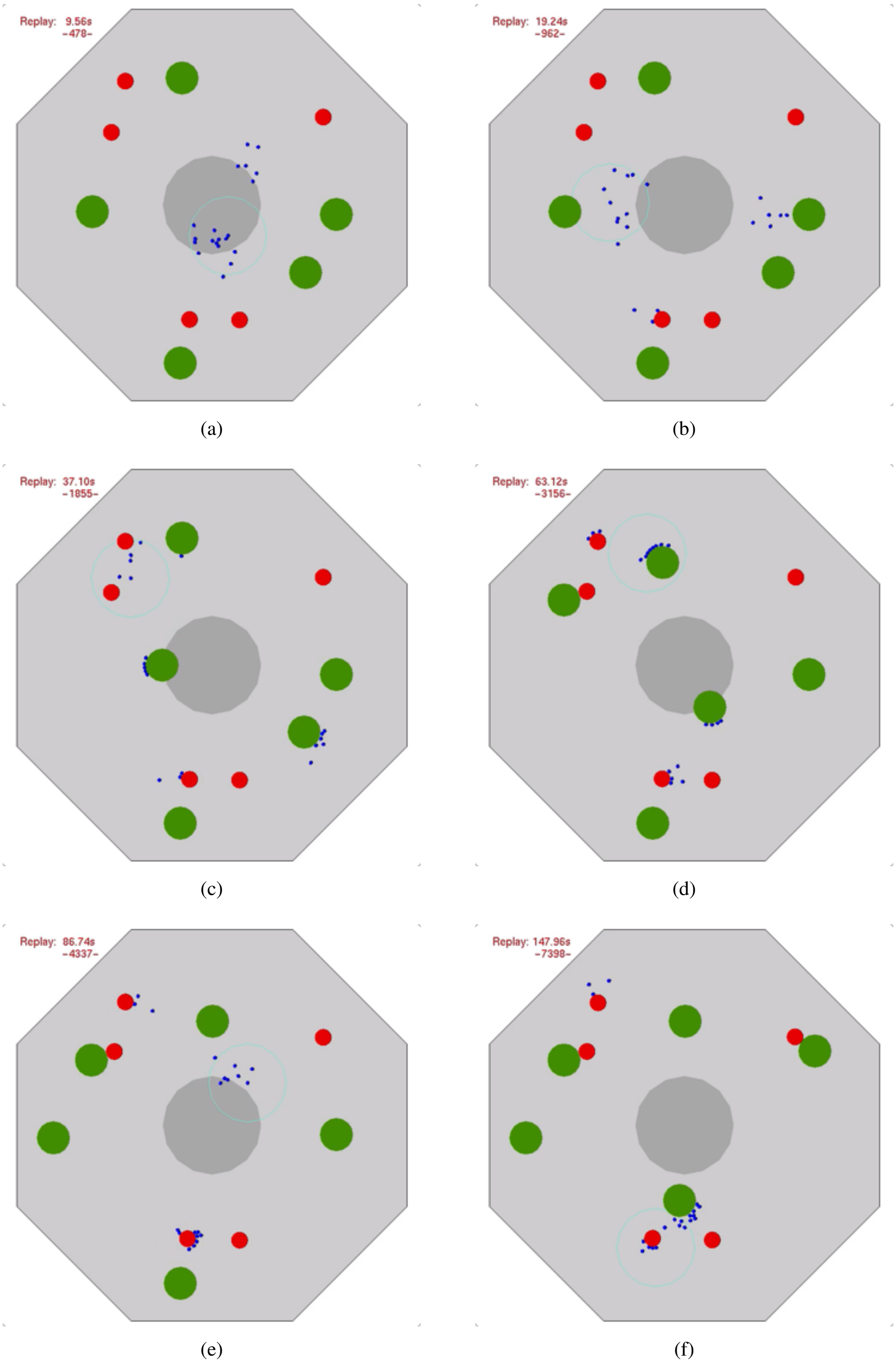


Fig. 6: Simulation snapshots of the controller developed in the experiment with 20 robots.

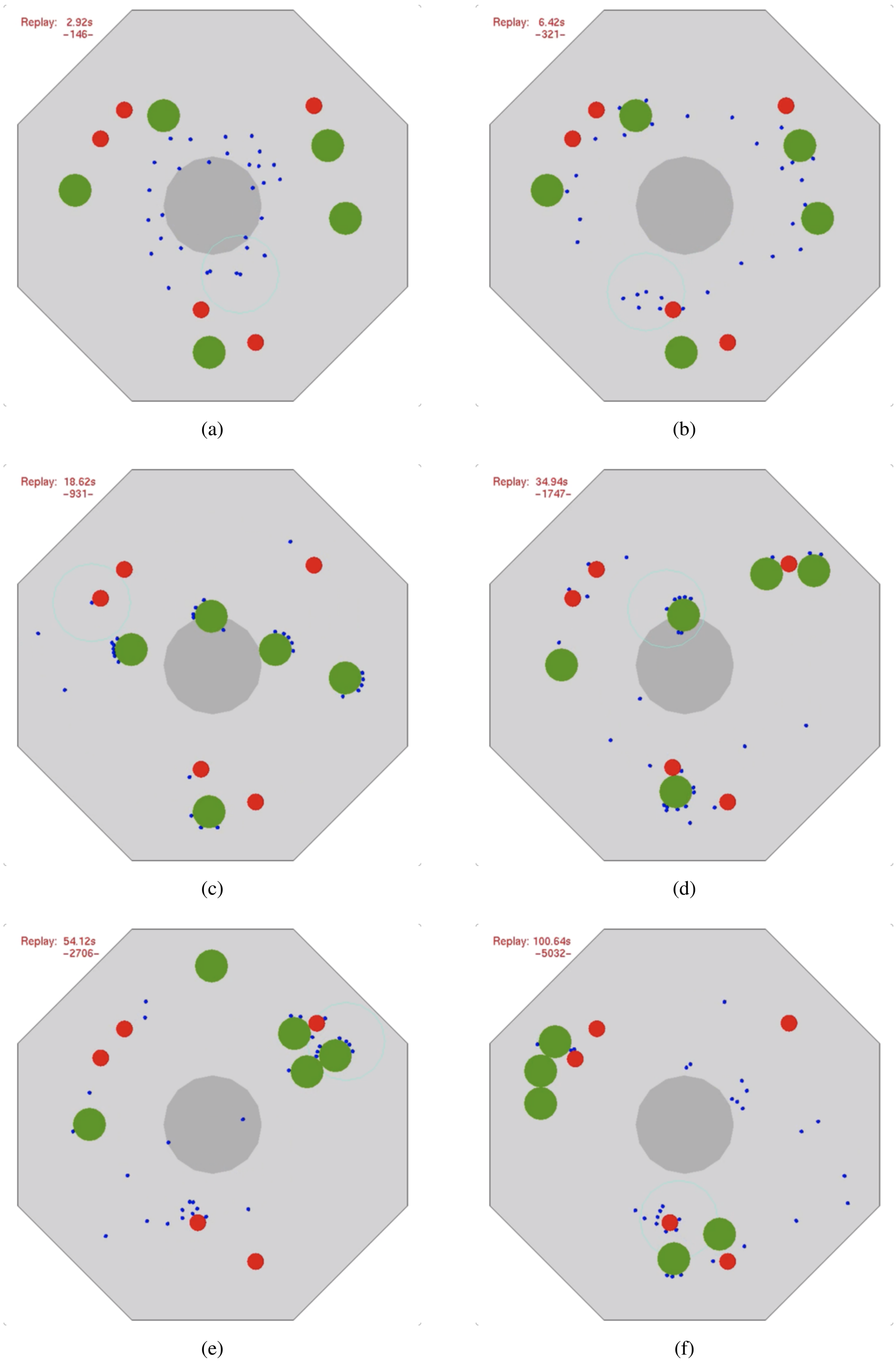
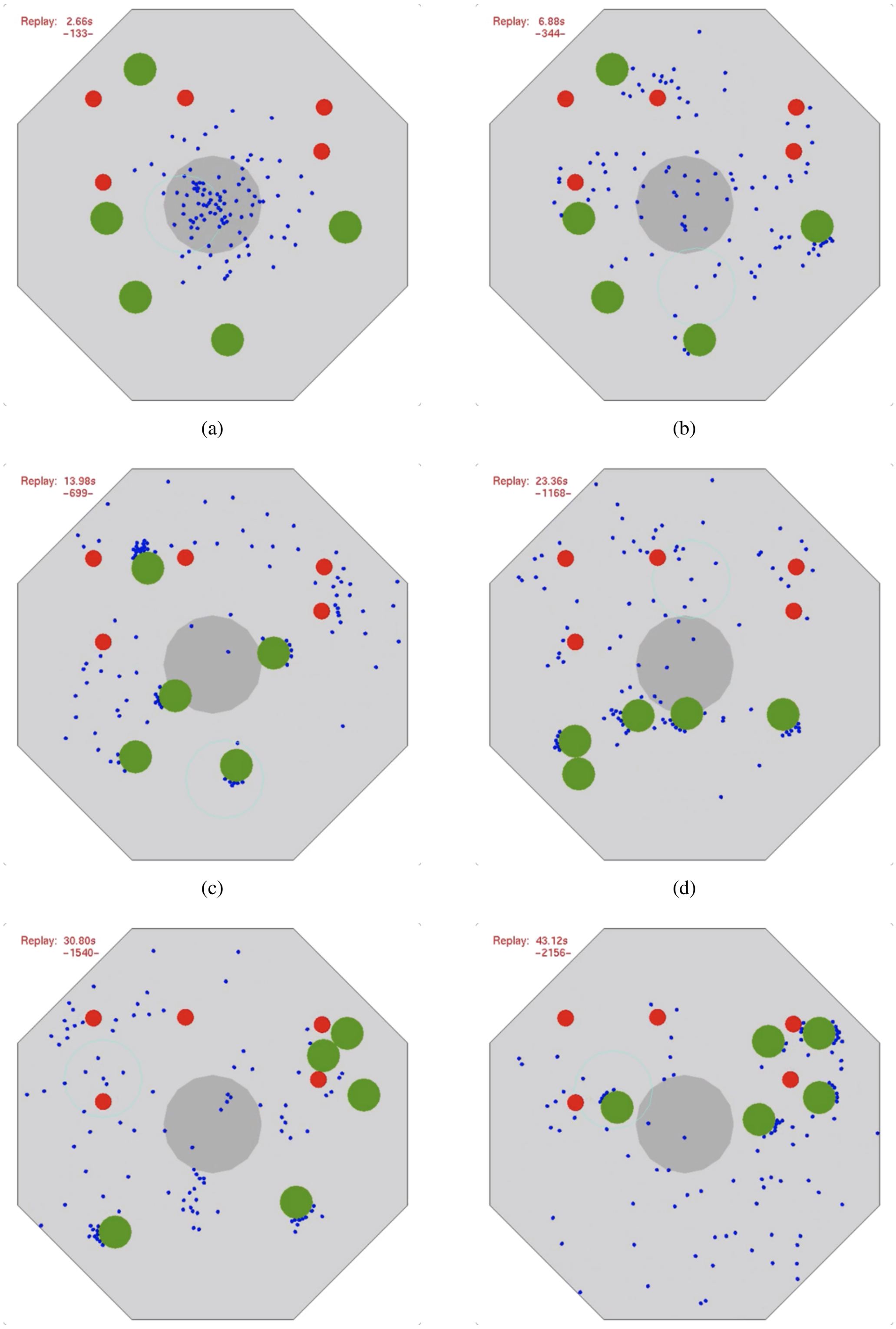


Fig. 7: Simulation snapshots of the controller developed in the experiment with 30 robots.



(a) (b) (c) (d) (e) (f)

Fig. 8: Simulation snapshots of the controller developed in the experiment with 100 robots.

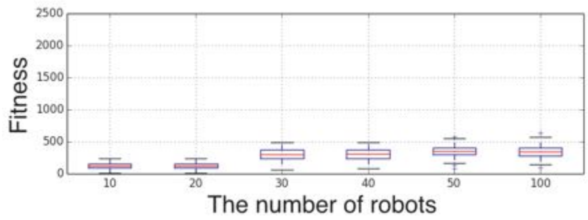


Fig. 9: The scalability of the controller developed in the experiment with 10 robots.

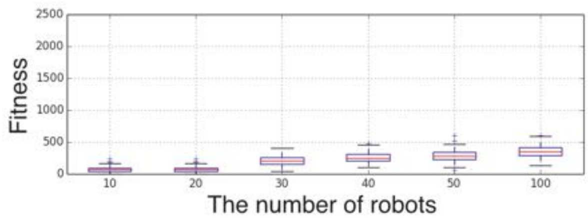


Fig. 10: The scalability of the controller developed in the experiment with 20 robots.

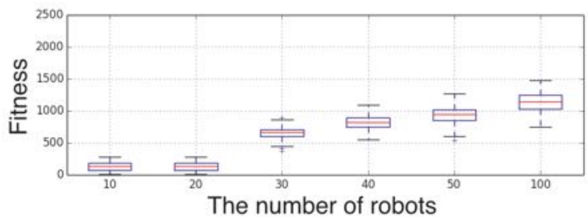


Fig. 11: The scalability of the controller developed in the experiment with 30 robots.

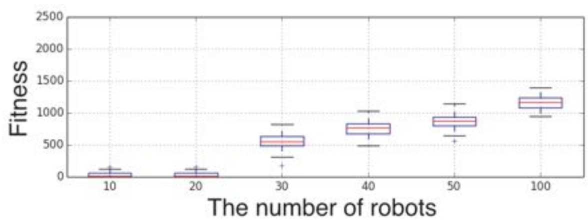


Fig. 12: The scalability of the controller developed in the experiment with 40 robots.

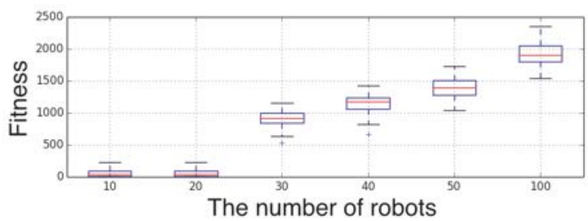


Fig. 13: The scalability of the controller developed in the experiment with 50 robots.

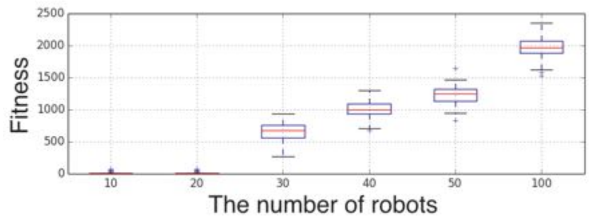


Fig. 14: The scalability of the controller developed in the experiment with 100 robots.

TABLE I: Discrimination rates in the scalability experiments. The rows of the table represent the number of robots in the development environments (Dev.) and the columns represent the execution environments (Exe.).

Exe. / Dev.	10	20	30	40	50	100
10	98.2%	96.7%	97.4%	95.1%	91.3%	87.2%
20	98.9%	99.4%	98.5%	99.2%	97.4%	94.6%
30	99.2%	99.2%	99.5%	94.3%	97.7%	91.9%
40	96.8%	96.8%	97.3%	99.2%	99.1%	94.4%
50	99.5%	99.4%	98.8%	98.3%	98.9%	92.4%
100	98.9%	99.1%	98.9%	98.8%	97.7%	95.3%

of the fitness in five trials to discuss the evolvability, i.e., how difficult it is to obtain the best controllers (see Fig. 5). The examples of the behavior obtained with 20, 30, and 100 robots are shown in Fig. 6, 7, and 8, correspondingly. In experiments with 10 or 20 robots, the artificial evolution developed strategies in which the robots are more likely to move as a group to transport only foods (see also Fig. 6). On the other hand, for experiments with 30, 40, 50, or 100 robots, the developed strategies allow the robots to act more independently, which can be explained by the high density of robots (see Fig. 7 and 8). Additionally, the performance grows exponentially as the number of robots increases. Considering the fact that the experiment with 100 robots has smaller standard variation than the others (except the experiment with 10 or 20 robots, which can be caused by the low performance), it can be said that it is easier for the artificial evolution to develop effective controllers with larger swarm sizes.

The results of the scalability experiments are as shown in Fig. 9, 10, 11, 12, 13, and 14. As can be seen from the figures, as the number of robots increases, the scalability to larger swarm sizes becomes better, meanwhile the scalability to smaller swarm sizes becomes worse. That is, the strategies (more likely to move as a group) developed in experiments with 10 or 20 robots show poor scalability to a larger number of robots. Furthermore, the strategies (more likely to move independently) developed in experiments with more robots also show poor scalability to a smaller number of robots. It is worthy to note that the best controllers developed in experiments with 50 or 100 robots exhibit similar scalability, which implies that with certain conditions, the developed controllers may have good scalability. For further discussion, the discrimination rate is calculated by the number of transported foods divided by the number of all transported objects. The discrimination rates for all scalability experiments are as shown in Table I. As can be observed from Table I, although the performance shows poor scalability in different environments in terms of the number of robots, the discrimination rates were kept at high values.

V. Conclusions

In this paper, we addressed a complex collective cognition task, namely the collective foraging task with poison, in which robots have to both cooperatively transport foods to the nest and collectively distinguish between foods and poisons. The covariance matrix adaptation evolution strategy algorithm was adopted to develop controllers for the robotic swarm under different conditions. We also performed experiments to examine the scalability of the developed controllers. The computer simulation results show that collective cognition behavior was successfully generated, which allows the robots to transport only foods.

We are planning to develop controllers in different situations in future work. In this paper, the sizes of the foods and the poisons were fixed with the foods having the larger size than poisons. Therefore, we are planning to test in situations where the poisons with the larger size than that of foods. We are also interested in how well a robotic swarm can distinguish between foods and poisons with similar sizes.

References

[1] E. Şahin, "Swarm robotics: from sources of inspiration to domains of application," in *Swarm Robotics*, ser. Lecture Notes in Computer Science, vol. 3342. Springer, 2005, pp. 10–20.

[2] M. Dorigo, M. Birattari, and M. Brambilla, "Swarm robotics," *Scholarpedia*, vol. 9, no. 1, p. 1463, 2014.

[3] I. D. Couzin, "Collective cognition in animal groups," *Trends in cognitive sciences*, vol. 13, no. 1, pp. 36–43, 2009.

[4] V. Trianni, E. Tuci, K. M. Passino, and J. A. Marshall, "Swarm cognition: an interdisciplinary approach to the study of self-organising biological collectives," *Swarm Intelligence*, vol. 5, no. 1, pp. 3–18, 2011.

[5] G. Valentini, D. Brambilla, H. Hamann, and M. Dorigo, "Collective perception of environmental features in a robot swarm," in *International Conference on Swarm Intelligence*. Springer, 2016, pp. 65–76.

[6] J. T. Ebert, M. Gauci, and R. Nagpal, "Multi-feature collective decision making in robot swarms," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 1711–1719.

[7] S. Nolfi and D. Floreano, *Evolutionary robotics: the biology, intelligence, and technology of self-organizing machines*. MIT press, 2000.

[8] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of IEEE International Conference on Evolutionary Computation, 1996*. IEEE, 1996, pp. 312–317.

[9] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[10] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[11] G. Francesca and M. Birattari, "Automatic design of robot swarms: achievements and challenges," *Frontiers in Robotics and AI*, vol. 3, no. 29, 2016.

[12] V. Trianni, *Evolutionary swarm robotics: evolving self-organising behaviours in groups of autonomous robots*, ser. Studies in Computational Intelligence. Springer, 2008, vol. 108.

[13] V. Trianni, R. Groß, T. H. Labella, E. Şahin, and M. Dorigo, "Evolving aggregation behaviors in a swarm of robots," in *European Conference on Artificial Life*, ser. Lecture Notes in Artificial Intelligence, vol. 2801. Springer, 2003, pp. 865–874.

[14] O. Soysal, E. Bahçeci, and E. Şahin, "Aggregation in swarm robotic systems: Evolution and probabilistic control," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 15, no. 2, pp. 199–225, 2007.

[15] G. Baldassarre, S. Nolfi, and D. Parisi, "Evolving mobile robots able to display collective behaviors," *Artificial life*, vol. 9, no. 3, pp. 255–267, 2003.

[16] V. Sperati, V. Trianni, and S. Nolfi, "Self-organised path formation in a swarm of robots." *Swarm Intelligence*, vol. 5, no. 2, pp. 97–119, 2011.

[17] M. Hiraga, Y. Wei, T. Yasuda, and K. Ohkura, "Evolving autonomous specialization in congested path formation task of robotic swarms," *Artificial Life and Robotics*, vol. 23, no. 4, pp. 547–554, 2018.

[18] R. Gross and M. Dorigo, "Towards group transport by swarms of robots," *International Journal of Bio-Inspired Computation*, vol. 1, no. 1-2, pp. 1–13, 2009.