Learning Collaborative Foraging in a Swarm of Robots using Embodied Evolution

Iñaki Fernández Pérez, Amine Boumaza, François Charpillet

Learning Collaborative Foraging in a Swarm of Robots using Embodied Evolution

Iñaki Fernández Pérez, Amine Boumaza
François Charpillet

Université de Lorraine, INRIA,
Campus scientifique BP 239 Vandoeuvre-lès-Nancy Cedex, F-54506,
firstname.lastname@loria.fr

June 7, 2017

Abstract

In this paper, we study how a swarm of robots adapts over time to solve a collaborative task using a distributed Embodied Evolutionary approach, where each robot runs an evolutionary algorithm and they locally exchange genomes and fitness values. Particularly, we study a collaborative foraging task, where the robots are rewarded for collecting food items that are too heavy to be collected individually and need at least two robots to be collected. Further, the robots also need to display a signal matching the color of the item with an additional effector. Our experiments show that the distributed algorithm is able to evolve swarm behavior to collect items cooperatively. The experiments also reveal that effective cooperation is evolved due mostly to the ability of robots to jointly reach food items, while learning to display the right color that matches the item is done suboptimally. However, a closer analysis shows that, without a mechanism to avoid neglecting any kind of item, robots collect all of them, which means that there is some degree of learning to choose the right value for the color effector depending on the situation.

1 Introduction

Swarm robotics aims at designing robot systems composed of a large number of simple robotic agents to perform tasks. Such systems rely on the number of agents rather than on their complexity. Swarm systems can have good properties such as redundancy and graceful degradation if some robots fail. Additionally, a swarm is able to solve tasks that are intrinsically cooperative, *i.e.* that a single agent could not solve.

How to design such swarm robot behavior is a fundamental question, and several methodologies have been proposed (Brambilla et al. (2013); Francesca and Birattari (2016)). Lacking a well-established engineering methodology to design swarm behavior, machine learning approaches to automatically build swarm robot control systems appear as a promising alternative, and many of the proposed approaches belong to the field of Evolutionary Robotics (ER, Nolfi and Floreano (2000)) and Evolutionary Swarm Robotics (ESR, Trianni (2008)). ER approaches use Evolutionary Algorithms (EA) to design robot controllers, and are of particular interest since those methods do not require complex information to guide the search for behavior, only an overall evaluation of the performance is required. Among these, Embodied Evolution (EE, Watson et al. (2002); Eiben et al. (2010a)) is a family of algorithms that take place online, once the robots are already deployed for operation. Evolution of behaviors is carried in a decentralized manner: each robot runs an Evolutionary Algorithm onboard, robots exchange genetic material when meeting (*i.e.* with robots in a close vicinity), and selection and variation are performed locally by the robot (Bredèche et al. (2015)). As such there is no central authority orchestrating the evolutionary process, in contrast with traditional ER approaches, and learning in the swarm emerges from interactions among the individual agents.

To perform cooperative tasks, swarms of robots need to coordinate their individual behaviors, and the task is solved by the resulting collective behavior. Evolving behaviors for intrinsically cooperative tasks is a complex problem. Several authors have addressed this problem in different contexts (Bernard et al. (2015); Waibel et al. (2009); Hauert et al. (2014)). In this paper, we study how a fully distributed EE algorithm can evolve swarm behavior for a cooperative foraging task that requires coordination between robots to collect different kinds of food items. The contributions of this paper are two-fold:

- First, we show that our EE algorithm run by every robot effectively learns a cooperative foraging task where different kinds of food items need to be collected.

- Second, we show that the task is indeed solved by learning a good strategy to coordinate behaviors, instead of learning simple opportunistic strategies. We show that all kind of food items are collected, although in different proportions, without resorting to any mechanism to avoid neglecting any of them.

The remainder of this paper is organized as follows. We first describe related approaches regarding evolution of cooperation, as well as studies on EE of swarm behavior. Then, we describe our experimental methodology, including the collective foraging task to be learned, as well as the Embodied Evolution algorithm run by the robots. Finally, we describe the results of the experiments and we discuss them, pointing out the contributions of this paper, and we conclude with some final remarks and perspectives.

2 Related work

A number of contributions have been made in evolving cooperative behaviors for robot swarms. This is a compelling problem in the field of evolutionary collective robotics. Although it has been widely studied with different aims, most of these works use a clonal approach where all the robots carry a copy of the same controller, leading to homogeneous team compositions (Francesca and Birattari (2016); Waibel et al. (2009); Tuci et al. (2008)). The teams are evaluated on the global fitness of the group that a centralized EA uses to optimize their behavior. On the other hand, there are some works that used coevolutionary approaches, where the population is decomposed in separate subpopulations, possibly one per robot (Gomes et al. (2015); Bernard et al. (2015)). Each genome in a subpopulation is evaluated against genomes of the other subpopulations, and evolution proceeds based on such fitness values. In this case, the composition of the team is by definition heterogeneous, *i.e.* the agents carry different genomes. In all these works, both clonal and coevolutionary approaches, evolution is performed in a centralized, offline manner, where an EA uses the fitness evaluations of all the agents to select the offspring for the next generations. Here, we study the evolution of cooperative behaviors using a distributed Embodied Evolution approach.

Embodied Evolution (Watson et al. (2002)) approaches are concerned with online evolution of behaviors run in parallel by each robot in the swarm. Robots exchange genetic material and possibly fitness measures when meeting, and evolution proceeds in such a distributed and asynchronous manner, which is akin of natural evolution. Many authors have addressed Embodied Evolution of behaviors for swarms of robots in different contexts, and several algorithms have been proposed. In those works, the authors use EE to study different topics and tasks: parameter tuning (Eiben et al. (2010b)), adaptation to changing environments (Dinu et al. (2013)), environment-driven survival (Bredèche et al. (2012)) evolution of self-assembly and aggregation (Silva et al. (2015); Weel et al. (2012)), topological neuroevolution (Silva et al. (2015); Fernández Pérez et al. (2015)), phototaxis and navigation (Silva et al. (2015); Karafotias et al. (2011)).

Distributed EE of swarm robot behavior raises some questions about the emergence of behavior for cooperative tasks, of division of labor or specialization. For example, Montanier et al. (2016) study the conditions for evolving specialization using Embodied Evolution approaches. They concluded that behavioral specialization is difficult to achieve in EE approaches, unless there is some degree of reproductive isolation in the swarm. Additionally, the authors insist on the importance of the size of the swarm and the selection pressure as parameters that may influence the emergence of specialized behaviors. In (Haasdijk et al. (2014)), the authors proposed a "market" mechanism that explicitly balances between different tasks to avoid neglecting hard tasks over easier ones (*e.g.* the most frequent or most rewarding). They test their approach in a foraging context where items of different types need to be collected individually by the robots. The authors consider collecting each type of item as a different task.

Different types of items available in different amount and in different areas of the environment, thus making some of the tasks easier than others.

The problem of evolving cooperative behavior for multirobot teams and robot swarms has already been studied using centralized EAs (see above). However, a legitimate question one could ask is: how can we evolve behaviors for an intrinsically cooperative foraging task in the distributed context of Embodied Evolutionary Robotics? In this paper, we evolve swarm robot behaviors in a cooperative foraging task with items of different colors using a simple Embodied Evolutionary algorithm which is a version of mEDEA (minimal Environment-driven Distributed Evolutionary Algorithm, Bredèche and Montanier (2010)) with selection pressure at the individual level. We compare the results with a case in which there is no selection pressure toward collecting items. We also take measures regarding the evolved strategies to collect food items of different colors, and conclude that there is no color neglected without resorting to explicit balancing mechanisms as in (Haasdijk et al. (2014)), although some of them are collected more frequently than others.

3 Collaborative item collection

In our experiments, a swarm of robots is deployed at random positions in an enclosed circular environment containing food items of different colors. We defined a task in which the robots must learn to collect these items cooperatively, since each item needs at least two robots next to it to be collected. To collect an item, robots must display a matching color signal using an additional colored led effector that is controlled by the robot's neurocontroller. Furthermore, the color signal that the robots display must match the color of the item to be collected. This imposes a synchronization constraint to the task, so robots are required to have some degree of coordination, or at least to reach a consensus on a color to use when collecting.

Each robot moves using two differential wheels which are also controlled by the robot's neurocontroller. Each robot has the following sensors:

- 8 obstacle proximity sensors,
- 8 robot proximity sensors,
- 8 food item proximity sensors,
- 8 color sensors, returning a value in $[-1, 1]$, corresponding to the color of the detected item, if any.

The robot's controller is an artificial recurrent neural network with the sensors as inputs and a bias unit, which are fully connected to three outputs neurons for the two wheels and the color effector. Additionally, all the outputs have recurrent connections from the previous right and left wheel speeds. The ANN computes the weighted sum of the sensors using a vector of synaptic weights that is encoded in the genome, and the activation is squashed using a $tanh(\cdot)$ function taking values between -1 and 1.

The task we define is a cooperative version of the *concurrent foraging* problem (Jones and Mataric (2003)), a problem in which different kinds of food items are available at the same time and have to be gathered in different ways, rather than having a single resource. In our case, whenever two or more robots are next to a food item and display the same color of the item, these robots collect the item, one point of reward is split among them, and another item of the same color appears at a random position in the environment. As such, the total number of items and the number of items of each color are kept constant. The weights of the neural network that controls both the movement and the color of the led on the robots are subject to evolution using an EE algorithm run by each robot (see below).

3.1 Embodied Evolution algorithm

All the robots in the swarm run an instance of the same EE algorithm, which is a variant of mEDEA that adds task-driven selection pressure. The idea of adding selection pressure toward a user-defined task to mEDEA is not new and was already studied in several works (Haasdijk et al. (2014); Fernández Pérez et al. (2014)). The algorithm is shown in Algorithm 1.

Algorithm 1 mEDEA with task-driven selection pressure

```
$g_a := random()$
while true do
    $f_a := 0.0$
    $\mathbf{l} := \emptyset$
    for $t = 1$ to $T_e$ do
        $exec(g_a)$
        $f_a := update\_fitness(g_a)$
        $broadcast(g_a, f_a)$
        $\mathbf{l} := \mathbf{l} \bigcup listen()$
    end for
    $\mathbf{l} := \mathbf{l} \bigcup \{(g_a, f_a)\}$
    $selected := select(\mathbf{l})$
    $g_a := mutate(selected)$
end while
```

At the beginning, the robot randomly initializes its active genome, which encodes the vector of synaptic weights of its neurocontroller. For T_e timesteps (one generation), the robot executes and evaluates its active controller. At every timestep, a robot runs its controller by reading the sensor inputs and computing the motor outputs. It updates the fitness value of its controller based on the outcome of its actions, and locally broadcasts to nearby robots its genome and current fitness. Those robots store the received genomes and fitness values into a local list. If the same genome is received several times during a generation, its fitness value is updated (*i.e.* multiple copies of the same genome are not allowed in local lists).

Once T_e timesteps are elapsed, the generation is finished, and the robot needs to choose a new active genome for the following generation. To do so, it selects a genome from its local list, possibly based on its fitness value. Subsequently, the selected genome is mutated using a Gaussian mutation. The resulting genome is mapped to the active controller for the following generation, and the local list is emptied. This means that selection is performed on the set of genomes of robots that were encountered during the previous generation. At this moment, the evaluation of the new controller begins.

Regarding the selection method, we use an elitist selection method, dubbed *Best*, which selects the genome with the highest fitness in the local list. In (Fernández Pérez et al. (2014)), it was shown that such method leads to the highest performing behaviors in a navigation with obstacle avoidance and in an individual item collection tasks. To provide quantitative comparisons, we also run a variant where selection is done randomly, as in mEDEA, thus disregarding any task objective (*e.g.* collecting items). Such method induces no task-driven selection pressure, and, as shown by Bredèche et al. (2012), evolves survival behaviors where robots learn to spread their genomes (*e.g.* by navigating to maximize mating opportunities). This provides a baseline for our experiments.

4 Experiments

In this section, we describe the experimental settings in our work, as well as the measures and experimental methodology for the post-analysis of the corresponding results.

4.1 Settings

In our experiments, 200 robotic agents are deployed in a circular environment containing 100 food items of 8 different colors, with the same proportion of each color. Each robot runs a copy of the algorithm presented in the previous section.

At every moment, each agent carries an active genome, which defines its current controller. The controller is run and the agent updates its fitness value when it collects an item, and locally broadcasts both its genome and its current fitness value to other nearby agents. The initial active genome of each robot is initialized with random weights between -1 and 1. When the evaluation period (lasting $T_e = 800$ timesteps) is finished, the agent selects a genome from its local list and mutates it by adding a normal random variable to each gene, with mean 0 and variance σ^2, $\mathcal{N}(0, \sigma^2)$ (in our experiments, $\sigma = 0.1$). Then, the local list of genomes is emptied and a new evaluation phase starts. We consider each evaluation phase as one generation.

Since our goal is to evolve cooperative foraging behaviors, in our experiments we use a task-driven selection method that deterministically chooses the best genome from the local list on each agent, in terms of fitness. When two or more

Table 1: Experimental settings.

Experiments	
Environment size	1000 × 1000 px
Number of food items	100
Food item radius	5 px
Swarm size	200 agents
Agent radius	6 px
Sensor range	20 px
Exp. length	1.6×10^5 sim. steps
Number of runs	30
Evolution	
Evolution length	200 generations
T_e	800 sim. steps
Genome size	105
Mutation step-size	$\sigma = 0.1$

agents are next to an item and they display a color matching the color of the item, 1 point of reward is split among all these robots. Fitness is measured as the sum of rewards obtained when collecting food items. Table 1 summarizes our experimental settings. The choice of those was based on preliminary experiments, although the exact values did not change the results significantly. For example, the evaluation period of $T_e = 800$ timesteps was chosen for the robots to have enough time to collect items, the mutation step of $\sigma = 0.1$ was chosen so the mutations were not too disruptive, and the density of robots (ratio between number of robots and environment size), as well as the communication and sensor ranges, were chosen to provide enough communication between robots for the distributed algorithm, while not having a too dense environment that would relatively hinder free movement.

We compare our results when using a task-driven selection pressure (*Best* selection) with a variant with random selection over the local list of each agent, as in mEDEA, (*Random* selection), which is our control experiment.

We use the Roborobo! simulator (Bredèche et al. (2013)), a simple and fast 2D simulator developed mainly for experiments in swarm robotics. Figure 1 shows a snapshot of the simulator with a robot swarm and different colored items. For each experiment, we run 30 independent runs to get statistical results.

Our experiments aim at answering two main questions:

- Does the swarm evolve behaviors to collect food items cooperatively?
- Are evolved behaviors opportunistic regarding the color of collected items? Otherwise stated, does the swarm learn to collect items of all colors, or are some of them neglected?

4.2 Measures

Here, we describe the post-analysis measures we use to answer the previously stated questions, and how we draw conclusions from them. To ascertain if cooperative collecting behaviors are evolved, we measure the total number of collected items by the swarm per generation, that we name *Swarm Fitness*. Since EE approaches evolve behaviors during robot operation (*i.e.* online), the

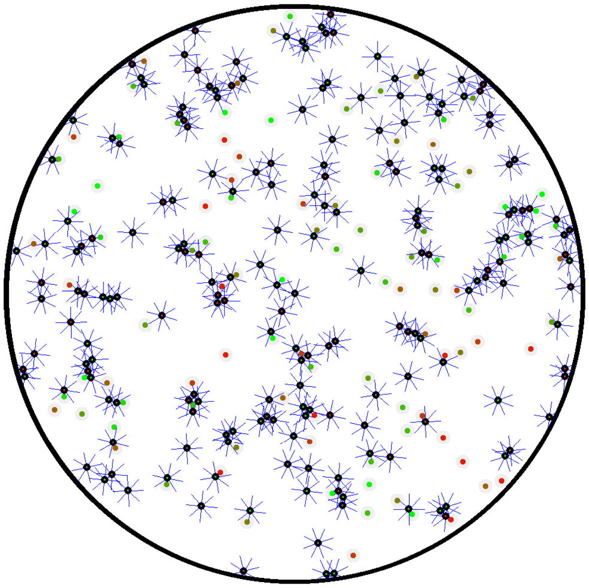

Figure 1: The simulated circular environment containing robots (black circles with thin hairlines representing sensors) and food items (colored dots).

best Swarm Fitness value or the Swarm Fitness at the last generation of our experiments are not reliable measures. Performance measures taking into account the quality of behaviors over time are thus needed. Following our previous work (Fernández Pérez et al. (2014)), we measure the average accumulated Swarm Fitness at the end of evolution as the average Swarm Fitness during the final 20% generations of each run.

Additionally, we measure the average individual reward obtained per collected item, over generations. Since each time an item is gathered one fitness point is split among the robots that participated in collecting it, averaging the individual rewards tells us if items are mainly collected by pairs of robots or by larger groups.

To evaluate how robots collect items cooperatively, we measure two of the components of the Swarm Fitness. First, we measure the average ratio of food items that could be collected at any moment over the total number of items, per generation (*i.e.* the number of items at every timestep that have at least two robots next to them, divided by the total number of items, averaged for each generation). The better the robots are in reaching items in groups of at least two robots, the higher will be the ratio of items that could be collected. Second, we measure the average ratio of items at every timestep that are actually collected among the possible items, over generations. This gives us an idea of how good behaviors are in terms of synchronizing the color effectors by jointly displaying the right color when collecting.

In order to evaluate if items of all the 8 colors are gathered, we measure the total number of items collected of each color for each run, and we compute the ratio over the total number of collected items (*i.e.* the proportion of collected items of each color in each run). Additionally, we compute the entropy of the proportion of items of each color, $H(\mathbf{p}) = -\sum_{i\in\mathbf{Colors}} p_i{\cdot}log_2 p_i$, which indicates how close is a proportion of items of each color to a uniform distribution where all colors are collected in the same quantity. When all the colors are collected in equal proportion (*i.e.* $\forall i,\ p_i = \frac{1}{8}$), the entropy is maximal with a value of 3. However, when only items of one color are collected, the entropy is minimal with a value of 0.

Finally, we perform pairwise comparisons of the aforementioned color proportions among all 30 runs of each experiments, to test if all the runs of each experiment yield a similar distribution of color proportions. First, the vectors of the 8 color proportions of a run are linearly normalized. Since different runs could evolve a preference toward different colors, we sort the coordinates of the normalized vectors from the most frequent color to the least frequent one. We compute a pairwise similarity measure between each pair of sorted vectors by using the dot product. The dot product yields 1 if the two vectors are collinear, 0 if they are orthogonal, and -1 if they are antiparallel. Summarizing, this measure gives us an idea of how similar are runs in terms of proportion of collected items per color. Note that this similarity measure is computed on the sorted vector, because we are interested in how the runs compare in terms of proportion between colors, not in the actual color value.

In the next section, we present the results of the experiments in terms of the aforementioned measures. To provide statistical results, we show the measures over 30 independent runs. In the case of measures over generations, the plots show the median and the interquartile range of the measure for the 30 runs over time. In the case of single measures, we provide violin plots showing the kernel density function of the dispersion on the data, as well as the datapoints as reference. The violin plots also show the median value, and the whiskers correspond to the maximum and minimum value over the 30 independent runs.

5 Results and discussion

In this section, we present and discuss the results of our experiments. A video of one simulation run using *Best* can be found at `rebrand.ly/FernandezECAL2017`. Figure 2 shows the Swarm Fitness (*i.e.* the number of collected items) per generation (left) for the experiment with task-driven selection pressure (*Best*, in blue) and for the experiment without task-driven selection pressure (*Random*, in orange). There is a clear increasing trend showing that the swarm learns how to collect items cooperatively in the case of *Best*. It reaches values of around 150 items per generation. There is also a slight trend of improvement in the case of *Random*, although much lower (around 12 items per generation). This is due to the fact that the robots learn to spread their respective genomes, and, as a byproduct, sometimes two of them meet on an item while displaying the right color, thus collecting the item. On the right, we show the average accumulated Swarm Fitness for both experiments. A Mann-Whitney U test shows that the difference between *Best* and *Random* is highly significant.

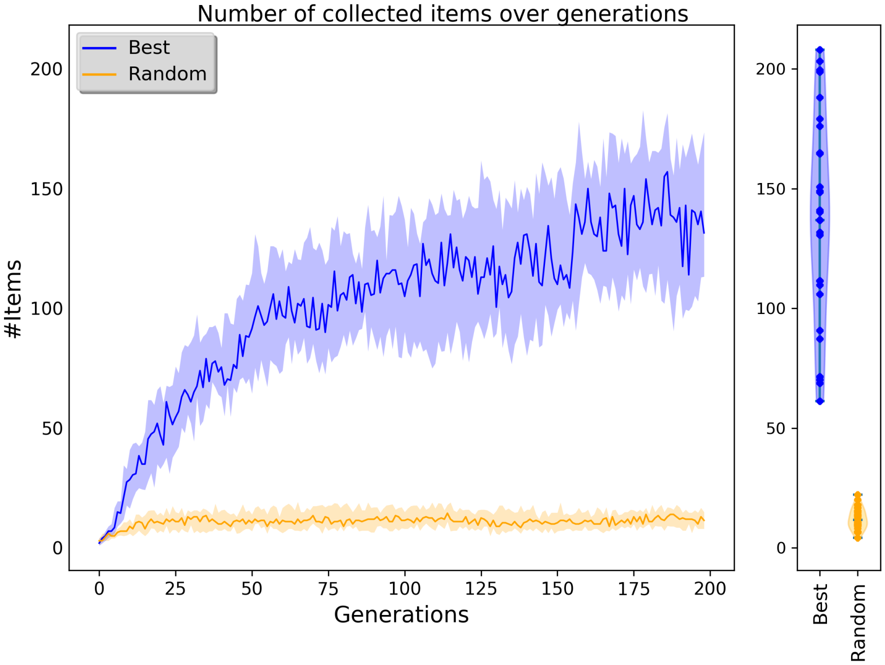

Figure 2: (*Left*) Total number of cooperatively collected items over time. (*Right*) Average fitness cumulated during the last 20% of the experiments in 30 independent runs. Mann-Whitney U tests show highly significant statistical difference between *Best* and *Random* (p-value $= 1.5 \cdot 10^{-11}$).

The average individual reward obtained per collected item has an almost constant value of 0.5 units in both *Best* and *Random* in all the runs (not shown here). This means that items are almost always collected by pairs of robots, and only very occasionally an item is collected by more than two robots. This is expected, for two reasons. First, reaching items and simultaneously displaying the right color in larger groups (more than two robots) is more complicated than reaching it in pairs. Second, when two robots collect an item, they get each a reward of 0.5 units. When an item is collected by more than two robots, the individual reward is lower. Since the number of items is always kept constant, there is no shortage of resources in the environment, and robots do not need to compete for them, *i.e.* they can search for other items to collect rather than collect items in large groups. Concretely, there is a reproductive disadvantage in terms of fitness in collecting items in larger groups.

Figure 3 (respectively, Figure 4) shows the average ratio of items that could be collected over the total number of items over generations (resp., the average ratio of collected items over the number of items that could be collected). These are two components of the Swarm Fitness, as discussed in the previous section. The trends in the results provide an interesting insight on the evolved behaviors regarding the cooperative item collection task.

The ratio of items that could be collected (Figure 3) improves considerably over time in the case of *Best*, reaching values of arond 20% of the total items, which is a high value, considering the number of items and robots, and the fact that, if an item gets collected, robots must search for another one, and thus will spend some time before they are next to it, even with an optimal controller. Consequently, this means that robots get very proficient at finding and getting next to food items. In the case of *Random* selection, the values are much lower, there is not such improvement, and the ratio even decreases slightly, stabilizing around 2% of the items in the environment. This shows that, at least in our experiments, getting next to items does not increase the chances for spreading robots' genomes, and the robots ignore the items. We visually observed that, indeed, the robots in this case do not move toward the items. Additionally, we show on the right the average accumulated ratio of items that could be collected over the last 20% of each run. Mann-Whitney U tests show a highly significant difference between *Best* and *Random*.

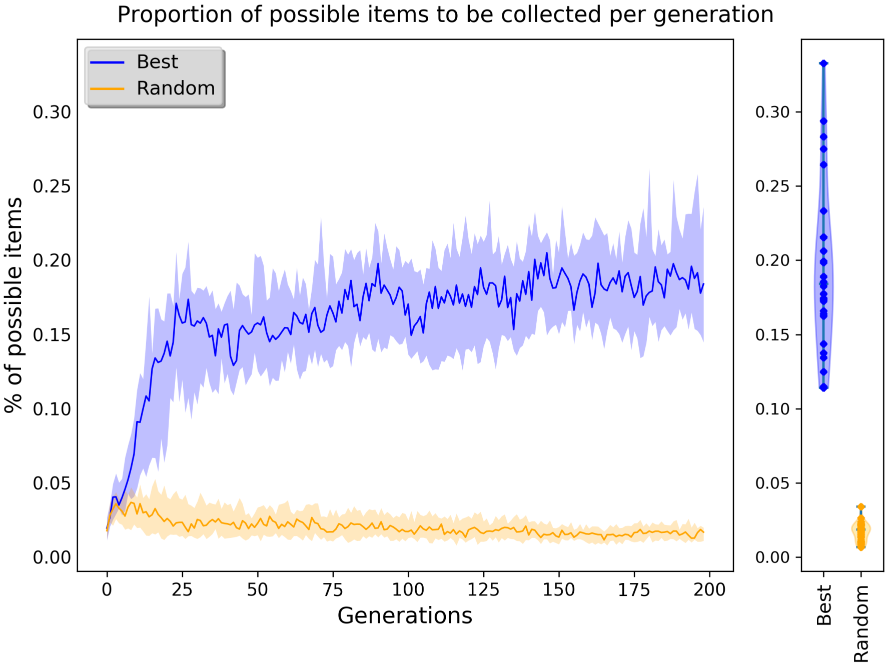

Figure 3: (*Left*) Proportion of items that have at least two robots next to them over time. (*Right*) Average accumulated proportion of items that could be collected during the last 20% of the experiments over 30 runs. Mann-Whitney U tests show highly significant statistical difference between *Best* and *Random* (p-value = $1.5 \cdot 10^{-11}$).

If we look at the ratio of collected items over the possibly collected items (*i.e.*

that had at least two robots), Figure 4 shows a different picture. The measure over generations in the case of *Best* is only slightly higher than for *Random*. We compute the average accumulated ratio over the last 20% of each run (shown on the right). Mann-Whitney U tests show that there is a significant statistical difference between the distribution of values for *Best* and *Random* (p-value = 0.0049). However, this difference in terms of actual value is slim. For *Best*, the value is around 1%, which means that robots only collect around 1% of the items having at least two robots next to them. This directly relates to their overall ability to jointly display the right color when they reach a food item, which is not very high (although slightly better than in the case of *Random*).

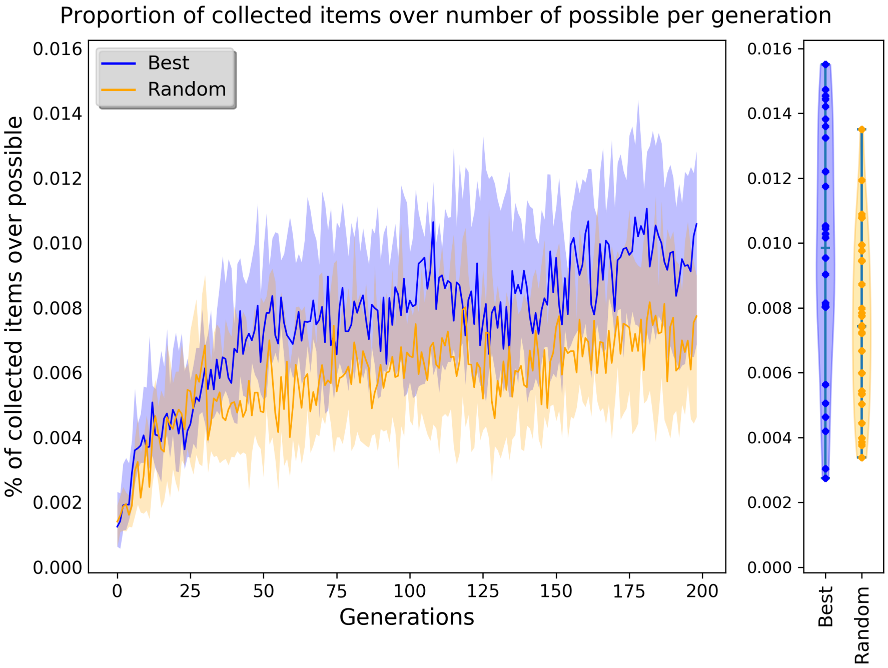

Figure 4: (*Left*) Proportion of items that are collected over the number of items that could be collected over time. (*Right*) Average accumulated proportion of collected items over possible ones. Mann-Whitney U tests show significant statistical difference between *Best* and *Random* (p-value= 0.0049).

To summarize, good cooperative foraging behaviors are mainly due to the ability of robots to jointly reach food items in groups of two robots. However, robots learn suboptimally to jointly display the color matching the item. The cause of this could lie in the encoding of the color output neuron in the neurocontroller, which has a sigmoid ($tanh(\cdot)$) activation function. Such function squashes the weighted sum of the inputs in a non-linear manner, and values saturating the neuron toward -1 or 1 are easier to display. This means that the items of the two colors corresponding to the maximum and minimum values are easier to collect.

To further investigate this behavior, we look at how items of different colors are collected. Figure 5 shows the proportion of the total number of items of each color collected per run. The violin plots are grouped in pairs corresponding to *Best* (left) and *Random* (right). The figure shows that in the two experiments the items of colors in both ends of the range are collected more frequently than the other colors. As said before, this is probably due to the saturation, either toward high or low values, of the output neuron controlling the color effector of each robot. However, there is a significant differerence between each pair of proportions (*i.e.* between *Best* and *Random*), as pairwise Mann-Whitney U tests reveal (all p-values < 0.03), except for the proportion of one type of items, orange in Figure 5, with a p-value $= 0.1975$. Furthermore, not only the difference is overall significant between *Best* and *Random*, but also *Best* systematically yields more balanced proportions, approaching a uniform distribution with $\frac{1}{8}$ for each color. To get quantitative measures of the balance among colors, we show on the right of Figure 5 the entropy of the proportion of items per color of *Best* and *Random* (see previous section). The results clearly show that *Best* leads to swarm behaviors that are more balanced in terms of the color of the collected items than in the case of *Random*. Mann-Whitney U tests show highly significant difference between both experiments in terms of entropy. Additionally, the entropy for *Best* gets close to the maximum value of 3.0 that would correspond to a completely uniform distribution.

As previously mentioned, Haasdijk et al. (2014) proposed a “market” mechanism to avoid neglecting tasks and balance the effort in a concurrent foraging task using a similar algorithm to ours. In their work, different types of items are found in different proportions, and are detected using different sets of sensors to emphasize the fact that the tasks are distinct. In constrast with that work, in our experiments all the types of food items are in the same amount. As such, it may seem that collecting items of each type is equally difficult. However, counterintuitively, due to the encoding of the color effector activation to collect the items (a single value between -1 and 1, discretized into 8 intervals for 8 different colors) and the neural activation function of the controller (a hyperbolic tangent, $tanh(\cdot)$), the colors corresponding to saturated activations (-1 or +1) are easier to display.

Finally, we measured the collinearity of the color proportions with the dot product for all the pairwise combinations of the 30 runs of *Best*, as described in the previous section. The results yielded values close to 1.0 (median $= 0.946$, upper quartile $= 0.987$, lower quartile $= 0.866$). This means that the independent runs of *Best* follow a similar trend regarding the balance between colors. Figure 6 shows the number of collected items over time of a typical run of *Best*, and the colored areas correspond to the number of items of each color collected per generation. The figure shows that items of all the colors are collected, so no one is neglected, and the number of collected items of each color increases over time, which means that robots progressively learn to cooperatively collect items of all the colors.

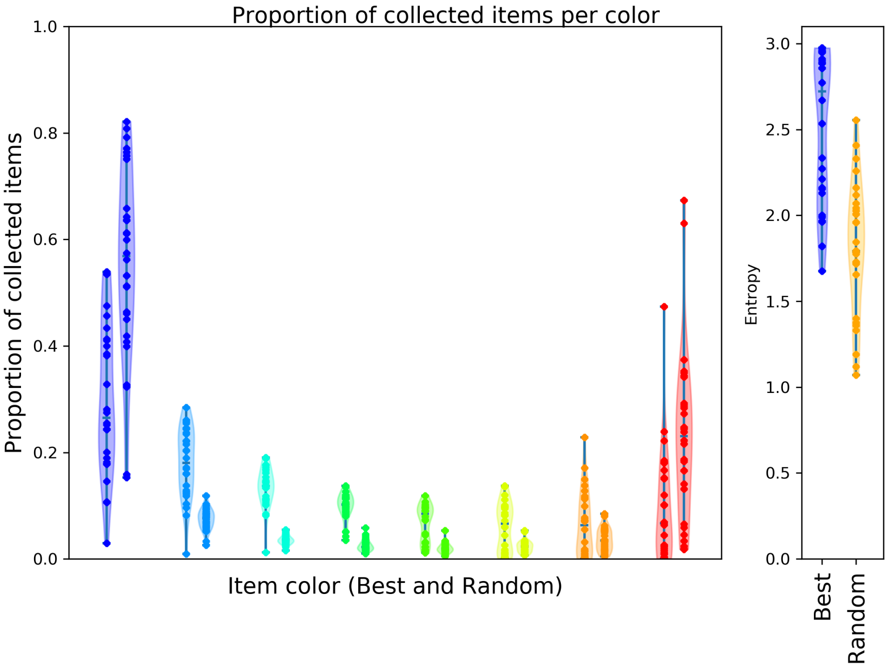

Figure 5: (*Left*) Proportion of collected items of each color over the total number of collected items. Each pair of violin plots shows such proportion of the corresponding color, for *Best* (left) and *Random* (right) selection methods. (*Right*) Entropy of the proportions of collected items of each color over the total number of collected items, with both selection methods. Mann-Whitney U tests reveal a highly statistical difference (p-value $= 5.1 \cdot 10^{-7}$).

6 Conclusion and perspectives

In this paper, we studied the capability of a distributed EE algorithm, mEDEA with task-driven selection pressure, to evolve swarm robot behaviors to solve a cooperative foraging task with different kinds of food items. The evolution of cooperative multirobot behaviors in ER is a challenge that has been widely studied from different points of views and with different approaches. Here, we used a fully distributed EE algorithm, that evolves heterogeneous controllers to learn collaborative behaviors. We showed that such an algorithm evolves behaviors for a cooperative foraging task with several types of food items that requires two or more robots to synchronize their actions to collect the items. We also showed that items are collected by pairs of robots rather than larger groups. Furthermore, we showed that the robot swarm evolves behaviors that do not neglect any kind of items, even without an explicit mechanism to enforce it.

Our experiments also showed that cooperative item collection is achieved mostly by jointly reaching food items. However, choosing the right color is achieved suboptimally. Additionally, even if our algorithm evolves behaviors

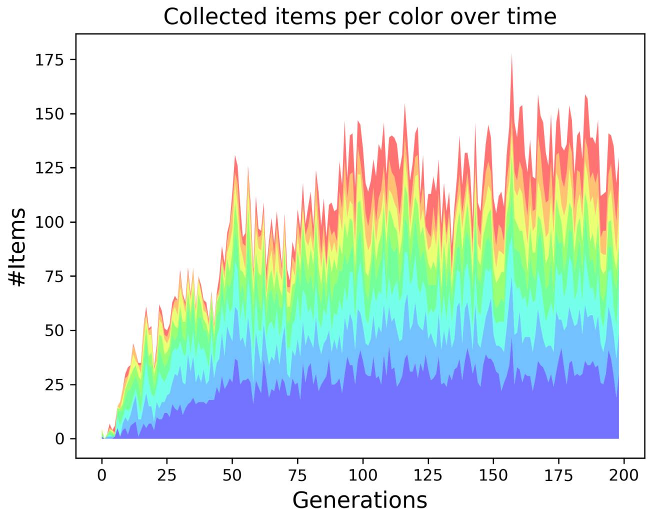

Figure 6: Number of collected items of each color in a typical run of the experiments using *Best* task-driven selection.

that do not neglect any type of items, the proportions of collected items of each color are not equal. These two facts could be due to the encoding of the color effector in the neurocontroller, which should be further studied. It would also be interesting to investigate if the learned strategies are different if we increase the frequency at which the items reappear for the colors in the middle (would the agents collect more of them?), if there is a shortage of items, or, in the same vein, if the reward is increased when items are collected by more than two robots (would agents collect items in larger groups to share the reward?). The analysis of the results of this paper sheds some light on the distributed evolution of cooperative foraging behavior, and it also raises several new questions about how to further improve the cooperative strategies, and how can resources of different types be shared in the robot population, depending on item density or proportion per type of item. Further research is needed to answer these questions.

References

Bernard, A., André, J.-B., and Bredèche, N. (2015). Evolution of cooperation in evolutionary robotics : the tradeoff between evolvability and efficiency. In *Proceedings of the European Conference on Artificial Life 2015*, pages 495–502.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41.

Bredèche, N., Haasdijk, E., and Prieto, A. (2015). Elements of embodied evolutionary robotics. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic* *and Evolutionary Computation*, GECCO Companion '15, pages 1247–1247, New York, NY, USA. ACM.

Bredèche, N. and Montanier, J.-M. (2010). Environment-driven Embodied Evolution in a Population of Autonomous Agents. In *PPSN 2010*, pages 290–299, Krakow, Poland.

Bredèche, N., Montanier, J.-M., Liu, W., and Winfield, A. (2012). Environment-driven Distributed Evolutionary Adaptation in a Population of Autonomous Robotic Agents. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):101–129.

Bredèche, N., Montanier, J.-M., Weel, B., and Haasdijk, E. (2013). Roborobo! a fast robot simulator for swarm and collective robotics. *CoRR*, abs/1304.2888.

Dinu, C. M., Dimitrov, P., Weel, B., and Eiben, A. (2013). Self-adapting fitness evaluation times for on-line evolution of simulated robots. In *Proc. of GECCO'13*, pages 191–198. ACM.

Eiben, A., Haasdijk, E., and Bredèche, N. (2010a). Embodied, On-line, On-board Evolution for Autonomous Robotics. In P. Levi, S. K. E., editor, *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution.*, volume 7 of *Series: Cognitive Systems Monographs*, pages 361–382. Springer.

Eiben, A., Karafotias, G., and Haasdijk, E. (2010b). Self-adaptive mutation in on-line, on-board evolutionary robotics. In *Fourth IEEE Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshop (SASOW), 2010*, pages 147–152. IEEE.

Fernández Pérez, I., Boumaza, A., and Charpillet, F. (2014). Comparison of selection methods in on-line distributed evolutionary robotics. In *Proceedings of the Int. Conf. on the Synthesis and Simulation of Living Systems (Alife'14)*, pages 282–289, New York. MIT Press.

Fernández Pérez, I., Boumaza, A., and Charpillet, F. (2015). Decentralized innovation marking for neural controllers in embodied evolution. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, GECCO '15, pages 161–168, Madrid, Spain. ACM.

Francesca, G. and Birattari, M. (2016). Automatic design of robot swarms: Achievements and challenges. *Frontiers in Robotics and AI*, 3:29.

Gomes, J., Mariano, P., and Christensen, A. L. (2015). Novelty-driven cooperative coevolution. *Evolutionary computation.*

Haasdijk, E., Bredèche, N., and Eiben, A. E. (2014). Combining environment-driven adaptation and task-driven optimisation in evolutionary robotics. *PLOS ONE*, 9(6):1–14.

Hauert, S., Mitri, S., Keller, L., and Floreano, D. (2014). Evolving cooperation: From biology to engineering. In *The horizons of evolutionary robotics.* MIT Press.

Jones, C. and Mataric, M. J. (2003). Adaptive division of labor in large-scale minimalist multi-robot systems. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1969–1974. IEEE.

Karafotias, G., Haasdijk, E., and Eiben, A. E. (2011). An algorithm for distributed on-line, on-board evolutionary robotics. In *Proc. of GECCO '11*, pages 171–178, New York. ACM.

Montanier, J.-M., Carrignon, S., and Bredèche, N. (2016). Behavioral specialization in embodied evolutionary robotics: Why so difficult? *Frontiers in Robotics and AI*, 3:38.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics.* MIT Press.

Silva, F., Urbano, P., Correia, L., and Christensen, A. L. (2015). odNEAT: An algorithm for decentralised online evolution of robotic controllers. *Evol. Comput.*, 23(3):421–449.

Trianni, V. (2008). *Evolutionary Swarm Robotics: Evolving Self-Organising Behaviours in Groups of Autonomous Robots (Studies in Computational Intelligence) (Studies in Computational Intelligence)*. Springer, 1 edition.

Tuci, E., Ampatzis, C., Vicentini, F., and Dorigo, M. (2008). Evolving homogeneous neurocontrollers for a group of heterogeneous robots: Coordinated motion, cooperation, and acoustic communication. *Artificial Life*, 14(2):157–178.

Waibel, M., Keller, L., and Floreano, D. (2009). Genetic team composition and level of selection in the evolution of cooperation. *Trans. Evol. Comp*, 13(3):648–660.

Watson, R. A., Ficici, S. G., and Pollack, J. B. (2002). Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Syst.* Elsevier.

Weel, B., Haasdijk, E., and Eiben, A. (2012). The emergence of multi-cellular robot organisms through on-line on-board evolution. In *Applications of Evolutionary Computation*, pages 124–134. Springer.