Swarm Intelligence (2021) 15:287–310
https://doi.org/10.1007/s11721-021-00198-2

Achieving task allocation in swarm intelligence with bi-objective embodied evolution

Qihao Shan[1] · Sanaz Mostaghim[1]

Received: 4 March 2021 / Accepted: 20 June 2021 / Published online: 4 July 2021

Abstract
In this paper, we seek to achieve task allocation in swarm intelligence using an embodied evolutionary framework, which aims to generate divergent and specialized behaviors among a swarm of agents in an online and self-organized manner. In our considered scenario, specialization is encouraged through a bi-objective composite fitness function for the genomes, which is the weighted sum of a local and a global fitness function. The former depends only on the behavior of an agent itself, while the latter depends on the effectiveness of cooperation among all nearby agents. We have tested two existing variants of embodied evolution on this scenario and compared their performances against those of an individual random walk baseline algorithm. We have found out that those two embodied evolutionary algorithms have good performances at the extreme cases of weight configurations, but are not adequate when the two objective functions interact. We thus propose a novel bi-objective embodied evolutionary algorithm, which handles the aforementioned scenario by controlling the proportion of specialized behaviors via a dynamic reproductive isolation mechanism. Its performances are compared against those of other considered algorithms, as well as the theoretical Pareto frontier produced by NSGA-II.

Keywords Embodied evolution · Task allocation · Swarm intelligence

1 Introduction

Swarm intelligence has long been investigated in the field of Biology, Sociology and Computer Science. It is observed that natural swarms of intelligent agents such as insect swarms and bird flocks have no global control mechanism and only rely on local interactions to produce various collective behaviors (Camazine et al., 2003). Researchers aim to understand the inner workings of natural intelligent swarms, as well as construct artificial swarm intelligence systems. In this field, much research is devoted to the study of embodied swarm intelligence, i.e. swarm robotics. By employing a large group of simple robots

✉ Qihao Shan
qihao.shan@ovgu.de

[1] Faculty of Computer Science, Otto-von-Guericke University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany

which coordinate themselves via local interactions, a robotic system can be made robust, flexible and scalable (Şahin, 2004).

However, the large number of sensors, actuators and the distributed nature of processing make controlling a swarm robotics system difficult. Also, to achieve a particular goal for the whole swarm, individual robots often need to be divided into various groups to perform different tasks. One particular challenge in the application of swarm robotics is designing algorithms to achieve such task allocation. Task allocation as a subject has been well studied in a multi-agent setting. Approaches such as coalition formation have been studied both in software agents (Shehory & Kraus, 1998) and in robot agents (Vig & Adams, 2007). It focuses on enumerating all possible allocation results by the agents and agreeing on the best one. Later, coalition formation was extended to multi-objective problems (Agarwal et al., 2015) using evolutionary multi-objective optimization techniques. Various Market-based approaches (Dias et al., 2006) have also been proposed to perform task allocation in a multi-robot setting. These multi-agent approaches focus on optimizing the task allocation through local or global trials of possible allocations and distributing the computational load onto all the robots available. It is assumed that robots have good planning capabilities, knowledge of tasks and other robots, as well as constant communication, all of which are not feasible in a swarm intelligence setting.

On the other hand, task allocation in swarm intelligence has long been studied using models that focus on environmental interaction with agents, especially Response Threshold Models (Bonabeau et al., 1996). Such models express the interactions between the environment and the agents during task allocation via having the agents respond to task-specific stimuli from the environment. If the stimulus exceed the corresponding threshold, the agent has a high chance to engage in a task and vice versa.

A minority of literature studies task allocation via interaction among the agents (Gordon et al., 1992; Pacala et al., 1996). Recently, there has been a growing interest in such models (Gordon, 2016). Notably, Chen et al. (2020) have investigated individual and social learning in creating a division of labor among agents in a self-organized way. This is similar to the approaches to task allocation in embodied evolution. Embodied evolution (Ficici et al., 1999) refers to the technique of performing online and continuous evolutionary algorithms in a distributed manner in a multi-robot system. This practice enables the robots to modify their behaviors continuously and adapt to an unknown environment.

Task allocation algorithms in both multi-agent and environmental-stimuli settings attempt to divide available agents into predefined subgroups, where the behaviors are separately programmed. In contrast, embodied evolutionary approaches attempt not only to split the agents into various groups, but also to create divergent and cooperative behaviors in a self-organized way. In order to create intelligent swarm behaviors that mimic those of natural swarm intelligence, both self-organization and division of labor are crucial components (Karaboga, 2005). However, achieving both of these features is proven to be difficult when using a traditional evolutionary approach, as subgroups in a population with different genomes are unlikely to coexist, and the fitter subgroup will overtake the less fit one in selection (Montanier et al., 2016). Various approaches have been used in embodied evolution to solve this problem, notably reproductive isolation through fitness biases in selection or having sparse communications.

In evolutionary robotics, much of the literature tries to evolve specialized cooperative behaviors via cooperative co-evolution (Gomes et al., 2016; Potter & Jong, 2000; Sun et al., 2020). Such algorithms achieve specialization by evolving different cooperative components in a multi-agent system in different populations, such that there is reproductive isolation among distinct subgroups of the agents. Cooperative co-evolution however must be run offline, centralized and cannot adapt to an unknown environment. There are a few other methods in embodied evolution proposed to replicate such reproductive isolation in an online, decentralized and asynchronous way (Montanier et al., 2016; Prieto et al., 2009). However, these methods only seek to isolate various subgroups in the population, and there is no mechanism that regulates the distribution of different behaviors in the population. Furthermore, some of the methods used to enforce reproductive isolation such as geographical isolation is not always effective, as noted in (Montanier et al., 2016) that successful division of labor requires sparse communication and the desired split to be roughly equal, thus limiting these algorithms to very specific scenarios.

Therefore, in this paper, we utilize a novel framework for a task allocation problem proposed by Chen et al. (2020). This framework encourages specialized behaviors via the interactions between a global and a local objective function. It places an emphasis not only on the individual behaviors, but also the proportion of agents assigned to each behavior. The latter breaks down the constraints required by existing embodied evolutionary approaches to produce optimal divisions of labor. We thus propose a new embodied evolutionary algorithm which is able to regulate the global distribution of specialized behaviors within the population. Via this algorithm, we aim to extend embodied evolution in producing divergent and cooperative behaviors to more general situations. After that, we test our proposed algorithm and compare it to other existing embodied evolutionary methods.

The rest of this paper is arranged as follows. Section 2 will introduce in detail the related works in embodied evolution, as well as other works regarding task allocation in swarm intelligence. In Sect. 3, we define our considered task allocation scenario modeled as an optimization problem. Then in Sect. 4, we will describe the algorithms under investigation in detail. We first employ two existing embodied evolutionary algorithms. Their performances are compared with a baseline algorithm that does not utilize collective learning. We then introduce our proposed bi-objective embodied evolutionary algorithm. Section 5 contains the experiments to assess the performances of considered algorithms and their results. In Sect. 6, we provide an integrated look and discussions on the results. Finally, Sect. 7 is dedicated to the conclusion.

2 Related work

Embodied evolution is an application of evolutionary techniques in creating robot controllers for multi-robot systems. In embodied evolution, every robot is seen as an individual in the population. This concept is different from the traditional evolutionary robotics techniques, where a population of robot controllers are generated and simulated in a centralized manner. In embodied evolution, the individual robots perform reproduction, mutation and selection in an online, decentralized and localized manner (Bredeche et al., 2018). However, much of the research in this field focuses on creating a single optimal controller for individual robots in the population. In such scenarios, the design of embodied evolutionary algorithms can be very similar to traditional centralized evolutionary techniques.

A minority of literature attempts to create divergent and cooperative behaviors among individual robots. Similar to the centralized evolutionary techniques, such as cooperative co-evolution which is used to provide reproductive isolation between sub-populations with different genotypes, in embodied evolution, similar reproductive isolation is created via either geographical and communicative isolation (Montanier et al., 2016), or by considering the affinity between individuals during recombination such that an individual is recombined with others close to its own genotype (Prieto et al., 2009, 2010; Trueba et al., 2013; Trueba & Prieto, 2018). Such reproductive isolation means that the population is essentially split into multiple sub-populations. However, there has been no mechanism to regulate the number of individuals in each sub-population, causing the multi-robot system to stagnate in a local minimum with a suboptimal distribution of tasks among its individuals.

Since efficient cooperation among agents usually requires a certain distribution of task, one potential mechanism is to define a global fitness function that indicates the performance in the cooperation of the agents. A similar global fitness function was considered in (Trueba et al., 2013). However, the authors considered only the fitness with respect to genomes of the subspecies instead of those of the individual agents, thus ignoring the distribution of subspecies themselves. Therefore, we adopt a different computational model proposed by Chen et al. (2020). They have noted that the task allocation behaviors in social insects are determined by the reward functions of their individual actions and those of their cooperative actions. Thus, in their model, they consider a composite reward function with a local and a global component. The former models the performances of the agents' individual actions, while the latter models the level of cooperation among the agents. In this model, the distribution of specialized behaviors is as important as the behaviors themselves. Chen et al. have explored task allocation scenarios that favor strong specialization as well as those that favor weak specialization, in order to better understand the behaviors of social insects. They have suggested two different decision-making strategies to model the decision-making processes in social insects: individual learning and social learning. The former is individual exploration of good options, and the latter is a collective algorithm similar to evolutionary methods. In individual learning, they have assumed the ability by individual agents to simulate the environment. In social learning, they have assumed constant communication among agents in the swarm, as well as the whole swarm being divided into a number of subgroups that are in their individual independent environment. These assumptions cannot be maintained when considering an artificial swarm intelligence setting, such as swarm robotics. We have therefore extended their model to a more generalized decentralized optimization scenario, and we have carried over the standard design constraints of fully decentralized embodied evolutionary algorithms.

Since the interaction between two reward functions causes trade-offs between the two objectives, we investigate this scenario using a multi-criteria decision-making framework (Branke et al., 2008). We use Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002) to compute a Pareto frontier of the two objectives and compare it to the experimental results obtained by our considered algorithms.

3 Problem statement

We consider a continuous task allocation problem as follows. There are N agents with index $n = 1, \dots, N$ that have to be allocated to two tasks, A and B. The agents have to optimize a parameter $x_n \in [0, 1]$, which indicates the proportion of time the agent n spends on performing task A. The parameter x_n needs to be optimized with respect to 2 objective functions $f(x_n)$ and $g(\bar{x})$. Here $f(x_n)$ is the local objective function that depends only on the agent's own decision, while $g(\bar{x})$ is the global objective function that depends on the total proportion of time among all agents devoted to the two tasks, hence $\bar{x}$. The agents are assumed to have only reactive behaviors with no planning capability. They also have limited communication ranges and can only exchange information with their peers nearby. They can measure the value of the local objective function $f(x_n)$, but can only measure a local estimate of the global objective function $g(\bar{x})$, denoted as g_n^*. It is computed using only the genomes within the communication range of robot n.

$$\begin{aligned} &g_n^* = g(\bar{x}_{n^*}) \\ &\quad \{n^* | n^* \in 1, \dots, N; \text{robot } n^* \text{is within communication range of robot } n\} \end{aligned} \tag{1}$$

The total measured reward is therefore expressed as follows.

$$\Pi_n = (1 - w)f(x_n) + wg_n^* \tag{2}$$

The agents thus need to find

$$x_n^* = \operatorname{argmax}_{x_n} [(1 - w)f(x_n) + wg_n^*] \tag{3}$$

where $w \in [0, 1]$ is the weight of g relative to the total reward.

In our experiments, $g(\bar{x})$ is assumed to be unimodal and thus have a singular optimum. It models the optimality of the distribution of the whole swarm's efforts between tasks A and B. It gives the highest reward when the current effort allocation is at the desired allocation, and gives less reward the further the current effort allocation deviates from the desired allocation. On the other hand, $f(x_n)$ can be multimodal, and models the local characteristics of the agents. For example, when $f(x_n)$ is unimodal, the individual agents have an optimal distribution of efforts between the two tasks in question, and the characteristics of the agents does not favor specialization. In case that $f(x_n)$ is multimodal, the agents can operate at optimal or near-optimal efficiency at multiple distributions of effort between the two tasks, therefore the characteristics of agents favors specialization. This model simplifies the computational model used by Chen et al. (2020).

4 Methodology

In this section, we introduce four distributed optimization algorithms investigated in this paper. We use individual random walk as a baseline algorithm to gauge whether the adoption of a particular collective approach is beneficial. We also use a basic embodied evolution algorithm as another baseline to assess the effectiveness of having more complexity in the design of embodied evolution algorithms. In addition, we adopt an existing technique for embodied evolution algorithms to evolve divergent specialized behaviors by adding an affinity bias to the selection process. Finally, we introduce our proposed algorithm for the considered bi-objective scenario.

4.1 Individual random walk optimization

We first introduce the baseline algorithm as shown in Algorithm 1, which is a random walk optimization on an individual level. The agents start from randomly selected x_n values within the decision space [0, 1]. At every control loop, each agent n measures the fitness Π_n , which denotes the current total obtained reward indicating the fitness of its current genome. The agent then compares Π_n with the fitness from its last control loop Π_n' and adopts the genome with the higher fitness. It then mutates its chosen genome using a Gaussian exploration noise with standard deviation of $\sigma_{\exp}$ (line 9 and 11). This baseline

```
Algorithm 1: Individual Random Walk Baseline
Input  : Initial $x_n \forall n$
Output: Optimized $x_n^* \forall n$
$x'_n = 0$
$\Pi'_n = -\infty$
Define $\sigma_{\exp}$
while True do
    Current Weighted Total Fitness: $\Pi_n = (1 - w)f(x_n) + wg_n^*$
    if $\Pi_n >= \Pi'_n$ then
        $x'_n = x_n$
        $\Pi'_n = \Pi_n$
        $x_n = N(x_n, {\sigma_{\exp}}^2)$, restricted between [0, 1]
    else
        $x_n = N(x'_n, {\sigma_{\exp}}^2)$, restricted between [0, 1]
```

algorithm achieves optimization via only the information available to the individual agents themselves, with no communication. It is expected that a viable collective approach needs to perform better than this baseline algorithm to justify the added communication complexity.

4.2 Baseline embodied evolution

Algorithm 2 shows the outline for a baseline Embodied Evolution (EE) algorithm similar to the one presented in (Bredeche et al. 2018). At every control loop, the agents exchange genomes and their corresponding fitness with their neighbors (line 4, 5). An agent selects the genome that corresponds to the highest fitness value among its neighbors (including itself), denoted by the index $m = 1, \ldots, M$ (line 7). After that, a Gaussian exploration noise with standard deviation $\sigma_{\exp}$ is added to mutate the genome (line 8). This is a very standard implementation of embodied evolution and is widely used in other studies in the field.

4.3 Embodied evolution with affinity bias

In Algorithm 3, we use a similar affinity bias in the selection process as proposed in (Prieto et al., 2009, 2010). Most of the mechanisms are the same as in Algorithm 2. However, in this algorithm, the fitness of neighboring robots which are compared during selection is multiplied with an affinity factor. This factor decreases as the difference between the two

```
Algorithm 2: Baseline Embodied Evolution
  Input  : Initial $x_n \forall n$
  Output: Optimized $x_n^* \forall n$
1 Define $\sigma_{\exp}$
2 while True do
3     Current Weighted Total Fitness: $\Pi_n = (1 - w)f(x_n) + wg_n^*$
4     Broadcast $x_n$, $\Pi_n$ to neighbors
5     Collect $x_m$, $\Pi_m$ from $M$ agents in communication radius
6     if $M > 0$ & $max \Pi_m > \Pi_n$ then
7         $x_n = x_{m*}, m* = argmax_m \Pi_m$
8         $x_n = N(x_n, {\sigma_{\exp}}^2)$, restricted between [0, 1]
9     Clear recorded $x_m$ and $\Pi_m$
```

Algorithm 3: EE with Affinity Bias

```
Input   : Initial $x_n \forall n$
Output: Optimized $x_n^* \forall n$
1  Define $\sigma_{\mathrm{exp}}$
2  while True do
3      Current Weighted Total Fitness: $\Pi_n = (1-w)f(x_n) + wg_n^*$
4      Broadcast $x_n$, $\Pi_n$ to neighbors
5      Collect $x_m$, $\Pi_m$ from $M$ agents in communication radius
6      if $M > 0$ & $max\Pi_m > \Pi_n$ then
7          $\Pi'_m = \Pi_m * exp(-|x_n - x_m|)$
8          $x_n = x_{\mathrm{m}*}, m* = argmax_m \Pi'_m$
9          $x_n = N(x_n, {\sigma_{\mathrm{exp}}}^2)$, restricted between [0, 1]
10     Clear recorded $x_m$ and $\Pi_m$
```

genomes increase (line 7). In this way, an agent selects the optimal genome using the fitness adjusted with the affinity factor (line 8). Therefore, the agents are encouraged to select genomes close to their own. This enforces a static reproductive isolation and is able to encourage the whole population to diverge into multiple subgroups to suit the requirements of an environment. Similar use of affinity bias is popular in embodied evolution to evolve divergent cooperative behaviors and is used in various state-of-the-art literature (Trueba et al., 2013; Trueba & Prieto, 2018).

4.4 Bi-objective embodied evolution

In order to perform embodied evolution in the considered task allocation scenario, we introduce an alternative algorithm as shown in Algorithm 4. Besides x_n, we introduce another variable $y_n \in \{-1, 1\}$ that indicates an agent's preferred direction to change the current genome x_n. $y_n = 1$ means that agent n prefers to increase x_n and $y_n = -1$ means that agent n prefers to decrease x_n. The estimated quality of preferred direction is denoted by ρ_n.

In this algorithm, a collective decision-making process which determines the more suitable y_n value is running simultaneously along with the optimization of genome x_n. Optimization of x_n is done in two alternating states, indicated by $state_n$. In State 0 (line 12–17), the agent performs a random walk similar to Algorithm 1. However, the agent will only sample a new value of x_n in the preferred direction indicated by y_n. In State 1 (line 32–39), the agent receives x_m from its M neighbors similar to Algorithms 2 and 3, but will select a random genome within its preferred direction indicated by y_n. The random walk in State 0, ensures that the random selection process in State 1 produces better offspring than the previous generation.

The preferred direction y_n is determined in another decision-making process (line 22–30) that is also executed in State 1. The agents exchange their current y_n values and their corresponding qualities ρ_n with their neighbors. Each agent then computes y_{Mode}, which is the prevailing y value in all neighboring agents and the agent itself. y_{Mode} is used to estimate the prevailing y in the whole swarm. Π_{mean} is computed as the mean of current total fitness obtained by all agents in the vicinity of agent n. The change in Π_{mean} (denoted as $\Delta_{\mathrm{fitness}}$) between consecutive control loops is then computed. $\Delta_{\mathrm{fitness}}$ is used to estimate the effect of the current prevailing y on the total fitness. It is expected that when the current prevailing y in the whole swarm indicates a direction that improves $g(\bar{x})$, agents are more likely to experience an increase in their own total fitness. Also, if the agent's individual y_n is the same as the prevailing y in the whole swarm, then it has contributed positively to the

Algorithm 4: Bi-Objective EE

```
Input  : Initial $x_n, y_n \forall n$
Output: Optimized $x_n^* \forall n$
$\rho_\mathrm{n} = 0$: Estimated Quality of Preferred Direction
Define $\sigma_\mathrm{exp}$
$\Pi'_n = -\infty$
$\Pi_{mean} = 0, \Pi'_{mean} = -\infty$
$x'_n = 0$
$state_n = 0$
while True do
    Current Weighted Total Fitness: $\Pi_n = (1-w)f(x_n) + wg_n^*$
    Broadcast $x_n$, $y_n$, $\Pi_n$, $\rho_n$ to neighbors
    if $state_n$=0 then
        # Local random walk, searching for local optimum
        if $\Pi_n \geq \Pi'_n$ then
            $x'_n = x_n$
            $\Pi'_n = \Pi_n$
            $x_n = N(x_n, {\sigma_\mathrm{exp}}^2)$, if $y_n = -1$ restricted between $[0, x_n]$, else between
              $[x_n, 1]$
        else
            $x_n = N(x'_n, {\sigma_\mathrm{exp}}^2)$, if $y_n = -1$ restricted between $[0, x_n]$, else between
              $[x_n, 1]$
        $state_n = 1$
    else
        Collect $x_m$, $y_m$, $\Pi_m$, $\rho_\mathrm{m}$ from $M$ neighbors in communication radius
        # CDM on expected direction of option change that leads to improvement
        $y_\mathrm{Mode} = Mode(y_{j,\, j\, =\, 1, ..., M}, y_n)$
        $\Pi_{mean} = Mean(\Pi_{m, m=1,...,M}, \Pi_n)$
        $\Delta_\mathrm{fitness} = \Pi_{mean} - \Pi'_{mean}$
        $\rho_\mathrm{n} = \Delta_\mathrm{fitness} \times y_n \times y_\mathrm{Mode} \times M$
        $\Pi'_{mean} = \Pi_{mean}$
        if $Max(\rho_{m,\, m\, =\, 1, \cdots, M}) > \rho_n$ then
            $y_n = y_m, m = argmax_m(\rho_{m,\, m\, =\, 1, \cdots, M})$
        if $\Delta_{fitness} < 0$ and $y_n = y_{Mode}$ then
            $y_n = -y_n$
        # CDM on individual Option
        if $y_n = -1$ then
            $\mathbf{m}_\mathrm{available} = \mathbf{m}[x_m < x_n]$
        else
            $\mathbf{m}_\mathrm{available} = \mathbf{m}[x_m > x_n]$
        if $\boldsymbol{m}_{available} \neq \emptyset$ then
            $x'_n = x_n$
            $\Pi'_n = \Pi_n$
            $x_n = x_m, m = RandomChoice(\mathbf{m}_\mathrm{available})$
        $state_n = 0$
        Clear recorded $x_m$, $y_m$, $\Pi_m$ and $\rho_m$
```

effect of the latter, while if they disagree, then the contribution is negative. Thus, we design the decision-making mechanism such that when y_n is the same as y_{Mode}, Δ_{fitness} contributes positively to the computation of y_n's quality and vice versa. The quality of y_n (denoted as ρ_n) is computed in line 25 as follows:

$$\rho_n = \Delta_{\text{fitness}} \times y_n \times y_{\text{Mode}} \times M \tag{4}$$

ρ_n scales with both Δ_{fitness} and the number of neighbors M. The latter is because sampling the y values of more neighbors leads to more accurate estimation of the prevailing y of the whole population, therefore making the estimation of the quality of y_n more accurate. After that, an optimal new y_n is chosen from the y values of an agent's neighbors and that of itself based on their qualities ρ (line 27–28). Finally, if $\Delta_{fitness}$ is negative, and y_n is equal to the prevailing y, the performances of the whole swarm is likely decreasing, and the agents are moving away from optimal genome selections. Therefore, y_n value would be flipped to reverse this trend (line 29–30).

Overall, the proposed bi-objective embodied evolutionary algorithm does not freely copy high fitness genomes from other individuals in a population, as in traditional embodied evolutionary algorithms. In contrast, agents use the genome and fitness information from their neighbors to form an estimation of the more potentially optimal direction y_n to change its own genome x_n, and thus restrict the neighbors available for selection. This enforces a dynamic reproductive isolation which is both able to select a genome with high combined fitness and to maintain a separation of specialized behaviors if required.

5 Experiments

Our experiments are run on 20 simulated mobile robots. The specification of the mobile robots is set to model e-puck robots (Mondada et al., 2009). The arena is set to be 2-dimensional with size of 2 m × 2 m. The default maximum communication distance is set to be 0.5 m. In order to simulate the distributed and asynchronous process of embodied evolution, the length of a control loop is set to be an exponentially distributed random variable exp (0.1) s. To mimic the sporadic connectivity among the agents, the simulated robots are constantly performing random walk in the arena. Each robot would first move forward in its current direction for a random amount of time sampled from an exponential distribution with a mean of 40 s. When it senses another robot or the edge of the arena 0.1 m in front of it, or the timer runs out, it will turn in a random direction for a random amount of time sampled from a uniform distribution between 0 and 4.5 s. If no obstacles are detected when the timer runs out, the robot reverts back to moving forward, otherwise the robot will keep turning with a reset timer. Robot controls are actuated instantaneously, and no collision occurs in our simulations.

In the rest of this section, we will show our experiments and results. We first determine the optimal parameter settings for the considered algorithms. Then, we vary w between 0 and 1 to observe the performance of all considered algorithms with different levels of consideration of global and local fitness. At each environmental configuration, 20 experiments are conducted. We measure the performance via the peak total local fitness $\Sigma f(x_n)$ and total global fitness $Ng(\bar{x})$. The performances of considered algorithms will be compared among each other and with the Pareto frontier produced by NSGA-II, which is used to determine the theoretical optimum of our experiment scenarios. We will also compare the average

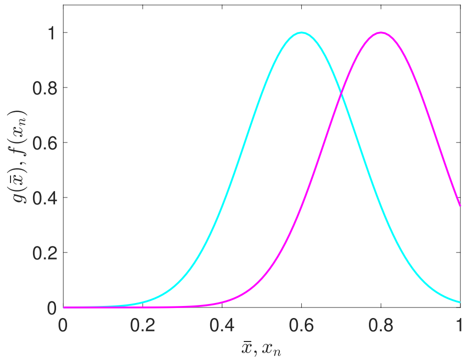


Fig. 1 Illustration of the global $g(\bar{x})$ (magenta) and the local $f(x_n)$ (cyan) fitness functions in Scenario 1 (Color figure online)

total fitness obtained over a 60s period to determine the long-term performances of the considered algorithms. We then add a stochastic term into the objective functions to see the performances of considered algorithms when facing uncertainties and noise.

5.1 Selection of σ_{exp}

An important parameter for all considered algorithms is σ_{exp}. It is selected in the following scenario. Both $f(x_n)$ and $g(\bar{x})$ are set to be simple unimodal functions defined as follows and shown in Fig. 1:

$$g(\bar{x}) = \exp\left(-\left(\frac{\bar{x} - 0.8}{0.2}\right)^2\right) \tag{5}$$

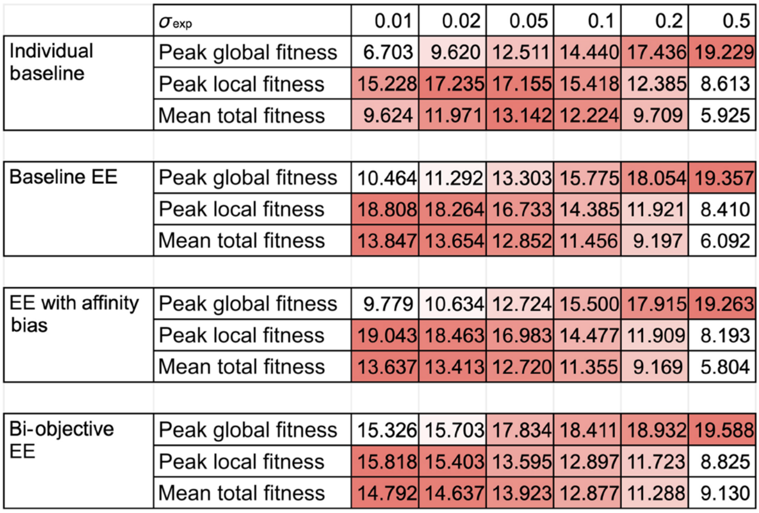
$$f(x_n) = \exp\left(-\left(\frac{x_n - 0.6}{0.2}\right)^2\right) \tag{6}$$

This scenario is meant to model task allocation problems in an environment that does not favor specialization. Since both of these objective functions have only a single optimum which are at different positions, we have a clear trade-off. An agent which moves its option closer to the optimum of $f(x_n)$, its corresponding $\bar{x}$ will move further away from the optimum of $g(\bar{x})$.

In order to find the optimal σ_{exp} value, we measure both the peak performances and the continuous performances of considered algorithms. The peak performances are measured via the global fitness and the local fitness when the total fitness obtained by the population reaches maximum during an experimental run. The weight of global fitness w is set to be 0.5. Therefore, we also seek to select a σ_{exp} value that can deliver results with balanced global and local fitness. The performances at different σ_{exp} values are shown in Table 1.

It can be observed in Table 1 that the peak global fitness and the peak local fitness are conflicting objectives with respect to the selection of σ_{exp}, as for all considered algorithms, a higher σ_{exp} produces a higher peak global fitness and a lower peak local fitness and vice versa. This is because a high local fitness requires all individuals to have similar genomes that are close to the optimum of the local objective function $f(x_n)$. This is hard to achieve when the individuals undergo drastic random mutations with a high σ_{exp} after every control

Table 1 Fitness performances at different σ_{exp} for all considered algorithms, $w = 0.5$ (Color table online)



	σ_{exp}	0.01	0.02	0.05	0.1	0.2	0.5
Individual baseline	Peak global fitness	6.703	9.620	12.511	14.440	17.436	19.229
	Peak local fitness	15.228	17.235	17.155	15.418	12.385	8.613
	Mean total fitness	9.624	11.971	13.142	12.224	9.709	5.925
Baseline EE	Peak global fitness	10.464	11.292	13.303	15.775	18.054	19.357
	Peak local fitness	18.808	18.264	16.733	14.385	11.921	8.410
	Mean total fitness	13.847	13.654	12.852	11.456	9.197	6.092
EE with affinity bias	Peak global fitness	9.779	10.634	12.724	15.500	17.915	19.263
	Peak local fitness	19.043	18.463	16.983	14.477	11.909	8.193
	Mean total fitness	13.637	13.413	12.720	11.355	9.169	5.804
Bi-objective EE	Peak global fitness	15.326	15.703	17.834	18.411	18.932	19.588
	Peak local fitness	15.818	15.403	13.595	12.897	11.723	8.825
	Mean total fitness	14.792	14.637	13.923	12.877	11.288	9.130

Table 2 Choice of σ_{exp} for different algorithms

	Individual	Base EE	EE affinity	Bi-ob
σ_{exp}	0.05	0.02	0.02	0.02

loop. On the other hand, a high σ_{exp} enables the population to explore the global objective function $g(\bar{x})$ more effectively and hence attain a higher global fitness. Given the conflicting nature of the two objectives, we seek to balance the two fitness obtained when the weight w is set to 0.5.

We also want the considered algorithms to stay on high fitness genomes continuously, given that the algorithms are designed to run online. Therefore, we consider the long-term continuous performance of the algorithms at different parameter settings and look at the mean total fitness during a 60-s period. It can be observed that the individual random walk baseline algorithm reaches maximum mean total fitness at $\sigma_{\text{exp}} = 0.05$, while for all variants of embodied evolution, mean total fitness decreases as σ_{exp} increases. The former is because the agents in individual random walk baseline do not copy genomes from each other, and thus all have to approach the optimum via exploratory mutations. Therefore, σ_{exp} needs to have a larger value to ensure a fast convergence. For other embodied evolution algorithms, the ability to copy genomes from other individuals ensures that convergence is quick even with a small σ_{exp}, while a large value of σ_{exp} can introduce instability to the population and reduce the mean total fitness.

Taking an integrated look at both aspects of the algorithms' performances, we pick σ_{exp} values for our algorithms that do not compromise the mean total fitness too much while seeking a balance between global and local peak fitness. The chosen σ_{exp} values are shown in Table 2.

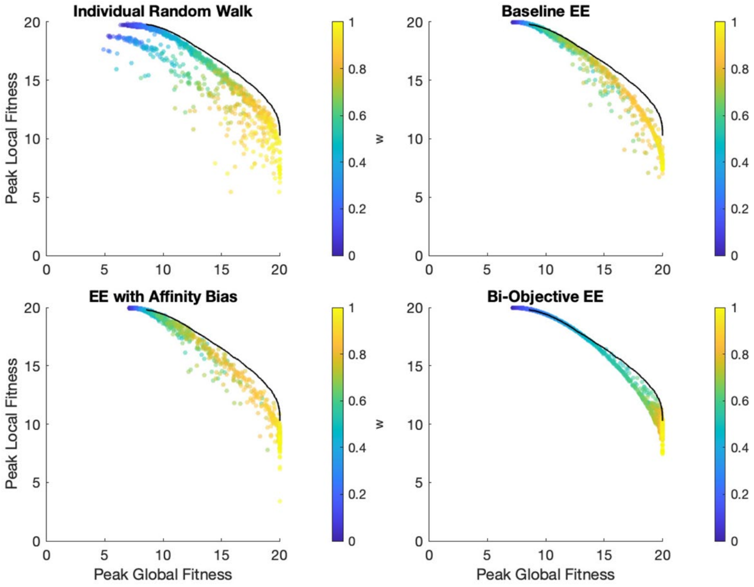


Fig. 2 Optimal global and local fitness obtained during each experimental run for all considered algorithms in Scenario 1, color scale indicates weight of global fitness *w*, black line indicates the Pareto frontier computed by NSGA-II (Color figure online)

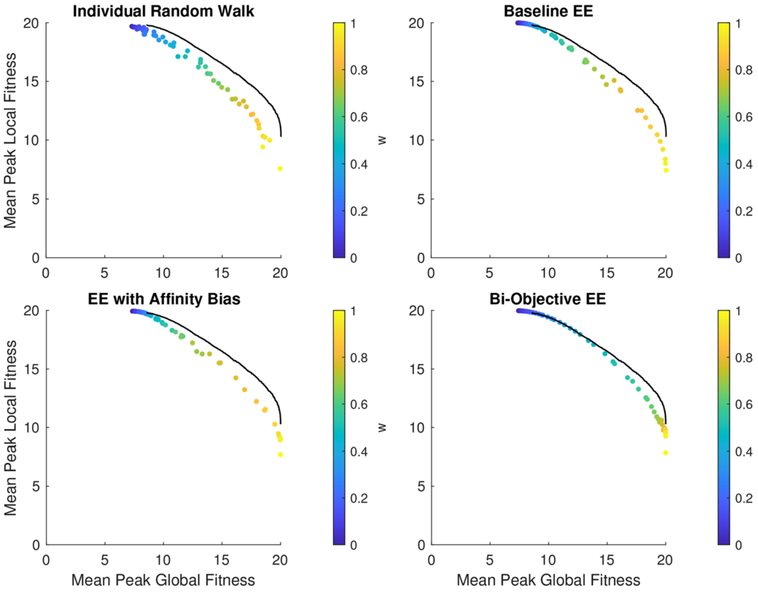


Fig. 3 Average optimal global and local fitness at each *w* value in Scenario 1 (Color figure online)

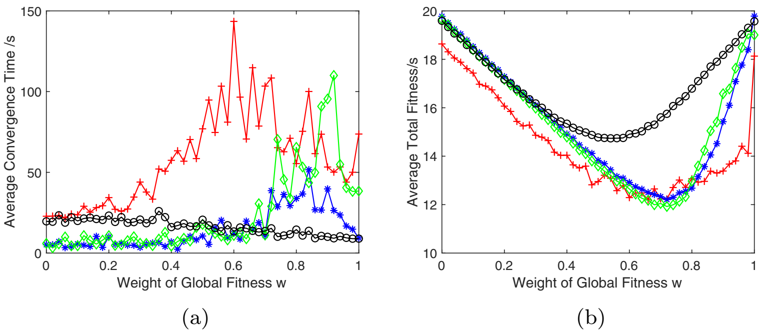


Fig. 4 Average convergence time (**a**) and average total fitness obtained/sec (**b**) of all considered algorithms in Scenario 1; red+: individual BL, blue*: BL EE, green◇: EE affinity, black○: bi-objective EE (Color figure online)

5.2 Optimal task allocation

In this part of the experiments, we examine the impact of the full range of w from 0 to 1 in an interval of 0.02 and compare the local and global fitness obtained when the total fitness reaches maximum across all considered algorithms.

5.2.1 Scenario 1: unimodal $f(x_n)$

In the first scenario, we use the same unimodal $f(x_n)$ and $g(\bar{x})$ as in Sect. 5.1. The results are shown in Figs. 2, 3 and 4.

As shown in Fig. 2 (top left), the Individual Random Walk approach is sometimes able to find the optimal global and local fitness quite well in its decision-making process. However, it also frequently converges to a suboptimal solution, indicated by the scattered data points below the Pareto frontier. Therefore, as shown in Fig. 3 (top left), the mean output at medium to high w values is quite far from the Pareto frontier. Regarding the convergence time (Fig. 4a), Individual Random Walk is usually slower than other algorithms, and its convergence time increases with w. Its long convergence time is due to the fact that the agents do not learn genomes with high fitness from each other, and they have to approach the optimal point using the random exploration steps. Its increase with respect to w is because for the total fitness $(1-w)f(x_n) + wg_n^*$, x_n has a much larger impact on $f(x_n)$ than on g_n^*. Therefore, the higher w is, the less impact x_n has on the total fitness and less accurate an estimate of the quality improvement of an exploration step is.

The peak performances of the Baseline EE (Fig. 2 top right) are mostly clustered together with high local fitness and low global fitness, except at high w values beyond 0.7. This is due to the fact that the decision mechanism of traditional evolutionary algorithms relies on comparing the fitness among the individuals. In this case, the agents compare the total reward received $\Pi_n = (1-w)f(x_n) + wg_n^*$. Since g_n^* values are very close within a local group of agents, it is often ignored in the comparison. Thus, the agents will frequently converge to the solution that maximizes $f(x_n)$. At high w values beyond 0.7, the high weight causes the small differences in g_n^* to be magnified and enables the algorithm to consider the global reward. As shown in Fig. 4a, Baseline EE is the fastest algorithm

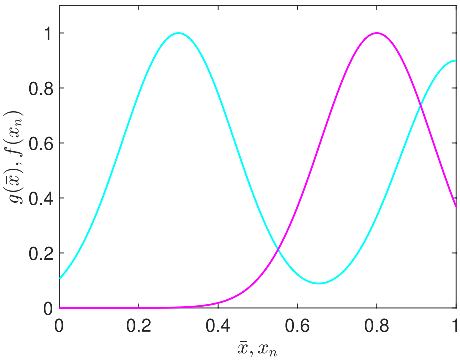


Fig. 5 Illustration of the global $g(\bar{x})$ (magenta) and local $f(x_n)$ (cyan) fitness functions in Scenario 2 (Color figure online)

for most of the cases, since the agents learn good options from each other and will quickly converge to an optimal option as long as it is reached by one agent.

EE with Affinity Bias (Fig. 2 bottom left) has similar peak performances to Baseline EE, as the affinity bias is mainly designed to create behavioral specialization and does not affect the evolution process significantly when the environment does not encourage specialization (Prieto et al., 2009).

Finally, Bi-Objective EE (Fig. 2 bottom right) obtains peak performances that are very close to the Pareto frontier. The results are considerably more consistent than those of Individual Random Walk. They are also more balanced than those produced by Baseline EE and EE with Affinity Bias, as the results at different w values are evenly spread along the Pareto frontier. Bi-objective EE avoids excessively focusing on local fitness like the other two variants of EE. Since it simultaneously decides on the preferred direction to change the agents' options y_n as well as the agents' genomes x_n. The decision-making process on y_n is able to keep track of the temporal changes of the agent's own fitness and the distribution of allocated tasks in the agent's locality $\bar{x}_{n*}$, and therefore is able to efficiently optimize g^*_n. As shown in Fig. 4a, Bi-Objective EE is faster than the individual baseline, but is slightly slower than the other two variants of EE except at high w settings.

The average total fitness obtained per second over a whole experimental run of 60s by all considered algorithms are shown in Fig. 4b. They are plotted against the weight of global fitness function w. It can be observed that all 4 algorithms have performances that are close to each other at both extreme ends of the values of w, while Bi-Objective EE outperforms the other algorithms, especially both variants of EE, at medium to high values of w from 0.4 to 1. As shown in Figs. 2 and 3, these are the values where the peak performances of both variants of EE are skewed toward the local fitness.

Overall, it can be observed that when the characteristics of the agents do not favor specialization, the agents need to be able to find an optimal balance between their individual local reward function and the global reward function. Therefore, although the two existing variants of embodied evolution are able to exceed the performance of the individual baseline when w is at either extreme ends of its values, they are unable to effectively consider the interactions between the two objective functions, and therefore are unable to achieve the optimal behaviors when w is at a medium value. On the contrary, Bi-Objective EE is able to attain the optimal behaviors consistently. It outperforms the two existing variants of embodied evolution in its bi-objective performances, while outperforms Individual Random Walk in reliability.

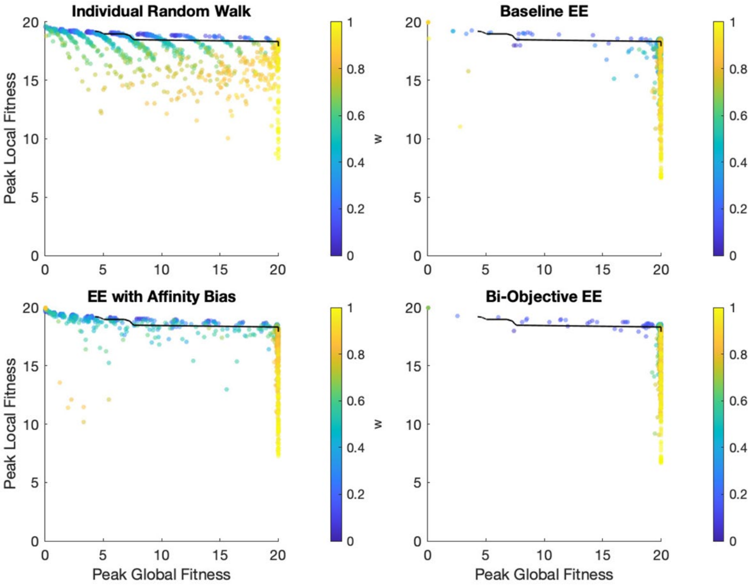


Fig. 6 Optimal global and local fitness obtained during each experimental run for all considered algorithms in Scenario 2, color scale indicates weight of global fitness w, black line indicates the Pareto frontier computed by NSGA-II (Color figure online)

5.2.2 Scenario 2: bimodal $f(x_n)$

The second experiment scenario keeps $g(\bar{x})$ as in Scenario 1 and defines a bimodal $f(x_n)$ as follows (Fig. 5):

$$f(x_n) = \exp\left(-\left(\frac{x_n - 1}{0.2}\right)^2\right) * 0.9 + \exp\left(-\left(\frac{x_n - 0.3}{0.2}\right)^2\right) \tag{7}$$

This scenario significantly differs from Scenario 1 in two ways: First, it is not guaranteed that the agents reach the optimum of $f(x_n)$ by taking incremental steps of improvement from a random initial position. Second, this scenario is meant to model task allocation scenarios that favor specialization and test the ability of the agents to self-organize into multiple subgroups. Since there are 2 local optima in $f(x_n)$, the agents' x_n can be split between the two optima in a particular proportion so that $\bar{x}$ lands on the optimum of $g(\bar{x})$. In addition, given the fact that the optimization problem is multi-modal with a lot of variables, NSGA-II has some difficulties in reaching the true Pareto frontier.

In the peak performances produced by the Individual Random Walk (Fig. 6 top left), there are a large number of data points that are below the Pareto frontier and arranged in stripes. This is because of the characteristics of this multi-modal scenario. Since there are two local optima, the population needs to converge to one of them when w is low and achieves a particular proportion between them otherwise. Since the individuals can only modify their genomes via exploratory random mutations, if an incorrect proportion of agents converges to one local optimum of $f(x_n)$, there is no way to change an x_n from one local optimum to another. Therefore, from the perspective of the whole population, there are many local optima in terms of total fitness to which the swarm can converge.

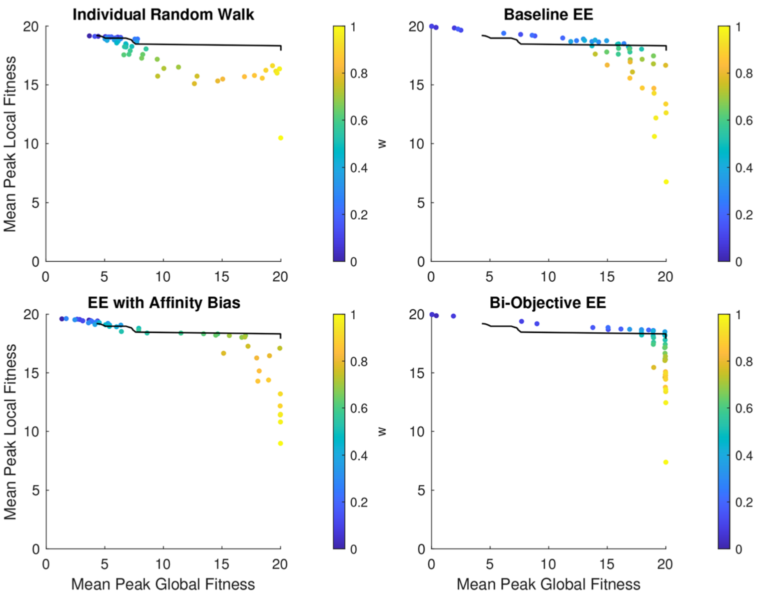


Fig. 7 Average optimal global and local fitness at each *w* value in Scenario 2 (Color figure online)

In Fig. 7 top left, we can observe that this approach works well at low *w* values but is unable to stay close to the Pareto frontier at high *w* values. This is a significant drop in performance compared to the results of the same algorithm in Scenario 1. It can be additionally observed that the Individual Random Walk fails to achieve the theoretical optimal performance at very low *w* values, which should have an optimal local fitness of close to 20 and global fitness of close to 0, compared to the results produced by Baseline EE and Bi-Objective EE.

Baseline EE (Fig. 6 top right) produces peak performances that mostly stay close to the Pareto frontier, with fewer suboptimal data points than Individual Random Walk. As can be seen in the plot of mean peak fitness (Fig. 7 top right), Baseline EE is able to reach optimal performances consistently at low *w* values. However, at medium and high *w* values, the results are unable to converge to the desired top right corner of the graph. Here, the same limitations in achieving optimal performances at high *w* values as in Scenario 1 are displayed for Baseline EE.

EE with Affinity Bias (Fig. 6 bottom left) produces less suboptimal data points than Individual Random Walk but more than Baseline EE. This is because the reproductive isolation created by the affinity bias limits the collective optimization among the agents and produces many potential local minima similar to Individual Random Walk. From the mean peak fitness plot (Fig. 6 bottom left), we can observe that EE with Affinity Bias only achieves optimal performances consistently at low *w* values, while the results can only approach the right section of the Pareto frontier at *w* values beyond 0.7.

As shown in Fig. 6 (bottom right), Bi-Objective EE has far less outlying data points below the Pareto frontier, with most data points clustering at either the top left and the top right corner of the graph. As can be seen in Fig. 7 (bottom right), Bi-Objective EE is not only able to consistently deliver optimal local fitness of 20 at very low *w* values,

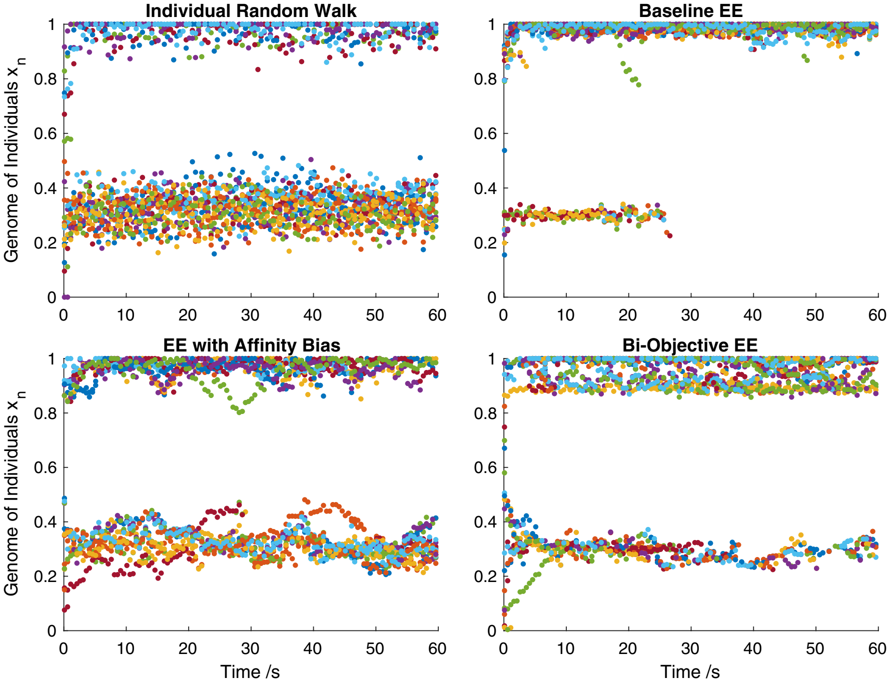
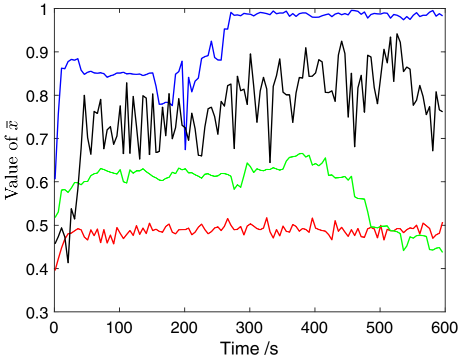


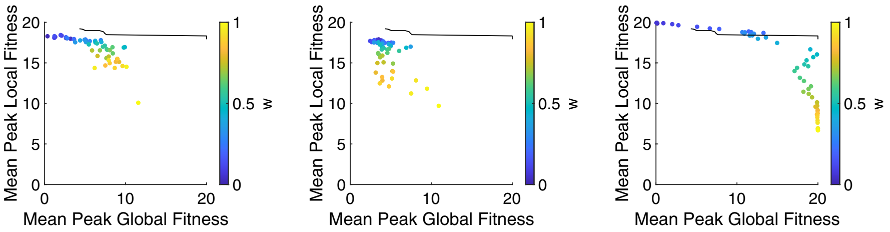
Fig. 8 Progression of all agents' genomes x_n through time ($w = 0.5$), colors represent different agents (Color figure online)

Fig. 9 Progression of $\bar{x}$ through time ($w = 0.5$); red: individual BL, blue: BL EE, green: EE affinity, black: bi-objective EE (Color figure online)



comparable to Baseline EE, but also to produce higher global fitness as w increases, which significantly outperforms other considered algorithms.

To understand the differences between the performances of considered algorithms, we take a look at the progression of individual genomes during an experimental run when $w = 0.5$, as shown in Fig. 8. The three approaches, Individual Random Walk, EE with Affinity Bias and Bi-Objective EE are able to split the whole population into 2 subgroups centering around the two local optima of the local fitness function in Fig. 5. In contrast, the whole population in Baseline EE eventually converges to one of the two



(a) Baseline EE (b) EE with Affinity Bias (c) Bi-Objective EE

Fig. 10 Average optimal global and local fitness at each w value in Scenario 2 under sparse communications, color scale indicates weight of global fitness w, black line indicates the Pareto frontier computed by NSGA-II (Color figure online)

local optima, thus ignoring the global fitness function entirely. This is why Baseline EE is not effective at dealing with a global fitness function.

On the other hand, although both the Individual Random Walk and the EE with Affinity Bias can split the individuals into two subgroups, the split does not consider the optimal proportion of agents in each subgroup and is thus heavily influenced by the initial distribution of genomes. As shown in Fig. 9, in both algorithms $\bar{x}$ remains around 0.5 despite the changes in individuals' genomes. This is because the genomes are initialized with a random value between 0 and 1, and the initial $\bar{x}$ is likely to be around 0.5. As the individuals converge to the local optima of the local fitness function closest to their initial genomes, the mean of all genomes $\bar{x}$ is unlikely to change much and thus still stays around 0.5, which is unlikely to be the optimal value for the global fitness function.

Finally, Bi-Objective EE is able to split the population into two subgroups and to regulate the proportion of individuals in each subgroup so that the global distribution of specialized behaviors is optimal. As shown in Fig. 9, $\bar{x}$ under Bi-Objective EE starts near 0.5 but moves to around 0.8, which is the optimal value of the global fitness function and oscillates around it. This ensures not only the individual genomes converge to optima of the local fitness function, but also the global distribution behavior is optimal as well.

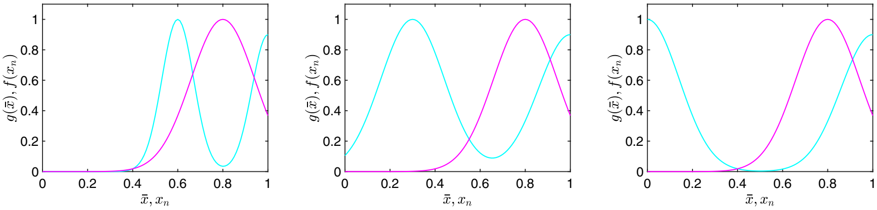
Overall, when the characteristics of the agents favor specialization, Bi-Objective EE outperforms the other considered algorithms in reaching the optimal behaviors. This is due to its ability to consider the two objectives while using a collective approach, therefore it overcomes the multi-modality of the local fitness function.

5.2.3 Performance under sparse communications

In order to investigate the performances of considered algorithms when communications are sparse, we reduce the communication range of the simulated agents to 0.1 m for this experiment and observe the optimal global and local fitness obtained. As shown in Fig. 10a, b, when communications are sparse, existing variants of embodied evolution experience a significant decrease in performance. This is because these algorithms rely on selecting the fittest genome from their neighbors to converge to an optimal behavior. When the communications become sparse, the number of available neighbors decreases, and thus it becomes harder for the whole population to reach a high total fitness.

Table 3 Local fitness functions in Scenarios 3.1–3.3

Scenario	Local fitness function
3.1	$f(x_n) = \exp\left(-\left(\frac{x-1}{0.1}\right)^2\right) * 0.9 + \exp\left(-\left(\frac{x-0.6}{0.1}\right)^2\right)$
3.2	$f(x_n) = \exp\left(-\left(\frac{x-1}{0.2}\right)^2\right) * 0.9 + \exp\left(-\left(\frac{x-0.3}{0.2}\right)^2\right)$
3.3	$f(x_n) = \exp\left(-\left(\frac{x-1}{0.2}\right)^2\right) * 0.9 + \exp\left(-\left(\frac{x-0.0}{0.2}\right)^2\right)$



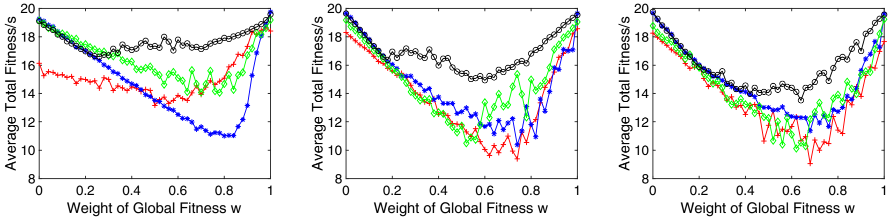
(a) Fitness Functions 3.1 (b) Fitness Functions 3.2 (c) Fitness Functions 3.3

Fig. 11 Illustrations of the global $g(\bar{x})$ (magenta) and local $f(x_n)$ (cyan) fitness functions in Scenario 3.1–3.3 (Color figure online)

On the other hand, as shown in Fig. 10c, Bi-Objective EE is able to maintain a high performance under sparse communications. Different from the other two embodied evolutionary algorithms, Bi-Objective EE does not directly copy high fitness genomes from their neighbors, but also relies on the temporal changes of their own fitness values to determine the optimal behavior. Therefore, it is more resilient to a reduction in number of neighbors. However, information from neighbors is important in Bi-Objective EE to accurately determine the quality of its y values. Thus, when comparing its performance here with in the previous section in Fig. 7 (bottom right), the performance here experiences a drop in both the global and the local fitness obtained, as fewer data points concentrate in the top right corner.

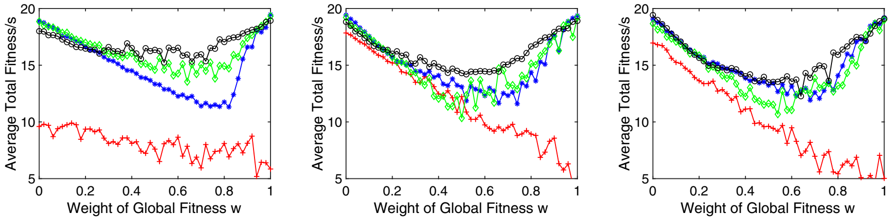
5.3 Long-term performances of specialized behaviors

In this subsection, we focus on the long-term performances of the considered algorithms in producing and maintaining specialized behaviors. This aspect of the algorithms is important given that they are designed to run online. In order to gauge the continuous performances, we compare the mean total fitness attained by the considered algorithms. As discussed in Montanier et al. (2016), behavioral specialization is more easily done when the targeted distribution is roughly equal. Therefore, we introduce three different scenarios that encourage a division of labor, but each requires progressively more imbalanced distribution of subgroups as shown in Table 3. The optimal proportion for the two subgroups are 1:1, 2:5 and 1:4, respectively. Illustrations of these fitness functions are shown in Fig. 11. We examine the average obtained reward by the considered algorithms in all 3 scenarios and with different w values. We additionally repeat the



(a) Scenario 3.1 (b) Scenario 3.2 (c) Scenario 3.3

Fig. 12 Average total fitness obtained/sec by the population at all w values for all considered algorithms; red+: individual BL, blue*: BL EE, green⋄: EE affinity, black∘: bi-objective EE (Color figure online)



(a) Scenario 3.1 (b) Scenario 3.2 (c) Scenario 3.3

Fig. 13 Average total fitness obtained/sec by the population at all w values for all considered algorithms with stochasticity added; red+: individual BL, blue*: BL EE, green⋄: EE affinity, black∘: bi-objective EE (Color figure online)

same experiments but with a stochastic term added to the fitness functions to measure the performances when facing uncertainty.

5.3.1 Average fitness in the absence of uncertainty

The mean total fitness obtained per second are shown in Fig. 12. We observe that both Baseline EE and EE with Affinity Bias outperform Individual Random Walk at small w values, but are overtaken at medium and large w values. This is because both variants of EE can quickly converge to the optimum of local fitness function, but they are unable to effectively consider the global fitness function, and therefore the mean fitness decreases steadily as w increases.

EE with Affinity Bias performs the best and significantly outperforms Baseline EE at medium w in Scenario 3.1 (Fig. 12a), where a balanced distribution of subgroups is encouraged. However, its advantage over Baseline EE diminishes in Scenario 3.2 and 3.3 (Fig. 12b, c), as there is no mechanism to regulate the distribution of individuals between subgroups, and the produced distribution is often not optimal.

The proposed algorithm Bi-Objective EE has slightly worse performances than other two EE variants in Scenario 3.1 (Fig. 12a) at low w values, but has comparable or better performances in all other situations. This demonstrates that Bi-Objective EE algorithm is able to deliver good continuous performances compared to other considered algorithms. However, its advantage over the other algorithms also diminishes as the desired distribution of tasks becomes more imbalanced. This shows that task allocation into imbalanced subgroups remains challenging for Bi-Objective EE.

5.3.2 Average fitness obtained with uncertainty

In the next step, we add a Gaussian noise with a standard deviation of 0.1 to the fitness functions measured by the individual agents. The performances of considered algorithms are shown in Fig. 13. Individual Random Walk's performance is reduced more significantly by the noise, compared to all EE variants. This is due to the fact that the collective learning in EE algorithms mitigates the effects of noise, as individual agents can pool their observations to gain more effective genome selection in the evolutionary process.

Bi-Objective EE is slightly more susceptible to the effects of noise than the other two variants, as it relies more on the individual observations. However, its performances are still consistently equivalent or better than other considered algorithms, especially at w values between 0.4 and 0.8.

6 Discussion

Taking an integrated look at the experimental results, we can make the following observations regarding the considered algorithms. Both variants of EE are effective at producing an optimal robot controller when only a local objective function is considered. Baseline EE is the best among the considered algorithms in such scenarios. It enables the individuals to quickly converge to the optimal genome regardless of the multi-modality of the considered objective function. Baseline EE also has full reproductive freedom among the agents, often producing better peak performances than algorithms with limited or no reproduction such as EE with Affinity Bias and Individual Random Walk, respectively. However, when a composite objective function is considered, the local component is given a much bigger emphasis than the global component. Furthermore, when a division of labor is encouraged by the interactions between global and local objective functions, Baseline EE still focuses on finding the singular optimum of the local objective function, often ultimately producing a total consensus rather than the desired division of labor and thus often has poor continuous performances.

EE with Affinity Bias is able to encourage a division of labor, and therefore, it sometimes has superior continuous behaviors than Baseline EE. However, it is still unable to effectively optimize the proportions of specialized behaviors according to the global objective function. Thus, it has an advantage over Baseline EE only when the desired distribution of tasks is roughly equal, but this advantage diminishes when the desired distribution becomes more imbalanced. At the same time, the static reproductive isolation limits the copying of genomes between the agents, causing EE with Affinity Bias to often have worse peak performance than Baseline EE.

To improve the performances of existing embodied evolutionary algorithms in a similar task allocation scenario, the agents should take into account of both the local reward of their actions and the effectiveness of the cooperation between agents with different individual configurations. The embodied evolutionary algorithms implemented should also move beyond enforcing static reproductive isolation, such as the affinity bias used in this paper, to generate divergent behaviors. Our proposed Bi-Objective EE algorithm addresses these two points by using a decision-making mechanism on the preferred direction to change the genomes of individual agents, thus achieving dynamic reproductive isolation. This mechanism of optimization can better regulate the proportion of specialized behaviors than the static reproductive isolation achieved via the affinity bias. Based on our experimental results, Bi-Objective EE is consistently equal or better than other algorithms in terms of performance. Its peak fitness obtained is close to the Pareto frontier computed by NSGA-II. Its continuous performance is only overshadowed by other variants of EE when a high weight is given to the local fitness function as opposed to the global fitness function. Compared to the other variants of EE, the proposed algorithm's mechanism of genome optimization relies more on the individual agents' local information and especially on the temporal changes of the agents' obtained fitness values. Thus, it is more resistant to the effects of sparse communication but less resistant to stochastic fitness functions. Also, task allocation remains challenging when the desired distribution is imbalanced.

7 Conclusion

In this paper, we consider a novel task allocation scenario in swarm intelligence. We seek to use embodied evolution to simultaneously evolve optimal specialized behaviors on an individual level and converge to an optimal distribution of behaviors on a global level. We model the task allocation scenario as a bi-objective optimization problem with a composite objective function that includes a local and a global component. We have applied two existing variants of embodied evolution to this scenario. Their performances in bi-objective scenarios with different weights for the two objectives are measured and compared with an Individual Random Walk baseline algorithm. We have concluded that in such bi-objective situations, both variants of embodied evolution are generally not adequate. Baseline Embodied Evolution has very good performance when only the local objective function is considered, but is unable to effectively consider the global objective function. Embodied Evolution with an Affinity Bias is able to produce a division of labor, but is also unable to optimize the distribution of individuals in the two subgroups and unable to reach optimum of the global fitness function.

We have proposed Bi-Objective Embodied Evolution as a new algorithm that is better able to deal with a bi-objective scenario that considers both the individual and global fitness of genomes. We have demonstrated that the Bi-Objective EE is able to quickly and accurately converge to the optimal genomes for all agents in various bi-objective scenarios. We have shown that it has superior peak and continuous performances compared to the three aforementioned strategies. It also produces comparable results to the theoretical optima computed by NSGA-II along the Pareto frontier. In addition, it converges to high fitness genomes and keeps them continuously, thus it is well-suited for an online application. It is also resistant to the effects of sparse communications and, to a degree, stochasticity in the objective functions.

In future works, we plan to extend the Bi-Objective EE to scenarios with multiple dimensions of genomes and investigate if multiple bi-objective embodied evolutionary processes can run simultaneously. Eventually, we aim to apply the Bi-Objective EE to real swarm robotics tasks and produce more complex specialized cooperative behaviors.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Agarwal, M., Agrawal, N., Sharma, S., Vig, L., & Kumar, N. (2015). Parallel multi-objective multi-robot coalition formation. *Expert Systems with Applications, 42*(21), 7797–7811.

Bonabeau, E., Theraulaz, G., & Deneubourg, J. L. (1996). Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. *Proceedings of the Royal Society of London Series B: Biological Sciences, 263*(1376), 1565–1569.

Branke, J., Branke, J., Deb, K., Miettinen, K., & Slowiński, R. (2008). *Multiobjective optimization: Interactive and evolutionary approaches* (Vol. 5252). Springer.

Bredeche, N., Haasdijk, E., & Prieto, A. (2018). Embodied evolution in collective robotics: A review. *Frontiers in Robotics and AI, 5,* 12.

Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J., Bonabeau, E., & Theraula, G. (2003). *Self-organization in biological systems*. Princeton University Press.

Chen, R., Meyer, B., & Garcia, J. (2020). A computational model of task allocation in social insects: Ecology and interactions alone can drive specialisation. *Swarm Intelligence, 14,* 1–28.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Dias, M. B., Zlot, R., Kalra, N., & Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE, 94*(7), 1257–1270.

Ficici, S. G., Watson, R. A., & Pollack, J. B (1999). Embodied evolution: A response to challenges in evolutionary robotics. In *Proceedings of the eighth European workshop on learning robots* (pp. 14–22). Citeseer.

Gomes, J., Duarte, M., Mariano, P., & Christensen, A. L. (2016). Cooperative coevolution of control for a real multirobot system. In *International conference on parallel problem solving from nature* (pp. 591–601). Springer.

Gordon, D. M. (2016). From division of labor to the collective behavior of social insects. *Behavioral Ecology and Sociobiology, 70*(7), 1101–1108.

Gordon, D. M., Goodwin, B. C., & Trainor, L. E. (1992). A parallel distributed model of the behaviour of ant colonies. *Journal of Theoretical Biology, 156*(3), 293–307.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report, technical report-tr06. Erciyes University, Engineering Faculty, Computer.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J. C., Floreano, D., & Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions* (Vol. 1, pp. 59–65). IPCB: Instituto Politécnico de Castelo Branco.

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

Pacala, S. W., Gordon, D. M., & Godfray, H. (1996). Effects of social group size on information transfer and task allocation. *Evolutionary Ecology, 10*(2), 127–165.

Potter, M. A., & Jong, K. A. D. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation, 8*(1), 1–29.

Prieto, A., Becerra, J. A., Bellas, F., & Duro, R. J. (2010). Open-ended evolution as a means to self-organize heterogeneous multi-robot systems in real time. *Robotics and Autonomous Systems, 58*(12), 1282–1291.

Prieto, A., Bellas, F., & Duro, R. J. (2009). Adaptively coordinating heterogeneous robot teams through asynchronous situated coevolution. In *International conference on neural information processing* (pp. 75–82). Springer.

Şahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics* (pp. 10–20). Springer.

Shehory, O., & Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence, 101*(1–2), 165–200.

Sun, L., Lyu, C., & Shi, Y. (2020). Cooperative coevolution of real predator robots and virtual robots in the pursuit domain. *Applied Soft Computing, 89,* 106098.

Trueba, P., & Prieto, A. (2018). Improving performance in distributed embodied evolution: Distributed differential embodied evolution. In *Artificial life conference proceedings* (pp. 222–223). MIT Press.

Trueba, P., Prieto, A., Bellas, F., Caamaño, P., & Duro, R. J. (2013). Specialization analysis of embodied evolution for robotic collective tasks. *Robotics and Autonomous Systems, 61*(7), 682–693.

Vig, L., & Adams, J. A. (2007). Coalition formation: From software agents to robots. *Journal of Intelligent and Robotic Systems, 50*(1), 85–118.