Seeking Specialization Through Novelty in Distributed Online Collective Robotics

Amine Boumaza

To cite this version:

Amine Boumaza. Seeking Specialization Through Novelty in Distributed Online Collective Robotics. EvoApplications 2022 - 25th International Conference on the Applications of Evolutionary Computation, Apr 2022, Madrid, Spain. pp.635-650, 10.1007/978-3-031-02462-7_40 . hal-03563761

HAL Id: hal-03563761

https://hal.inria.fr/hal-03563761

Submitted on 9 Feb 2022

Seeking Specialization Through Novelty in Distributed Online Collective Robotics

Amine Boumaza
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
amine.boumaza@loria.fr

February 9, 2022

Abstract

Online Embodied Evolution is a distributed learning method for collective heterogeneous robotic swarms, in which evolution is carried out in a decentralized manner. In this work, we address the problem of promoting reproductive isolation, a feature that has been identified as crucial in situations where behavioral specialization is desired. We hypothesize that one way to allow a swarm of robots to specialize on different tasks is through the promotion of diversity. Our contribution is twofold, we describe a method that allows a swarm of heterogeneous agents evolving online to maintain a high degree of diversity in behavioral space in which selection is based on originality. We also introduce a behavioral distance measure that compares behaviors in the same conditions to provide reliable measurements in online distributed situations. We test the hypothesis on a concurrent foraging task and the experiments show that diversity is indeed preserved and, that different behaviors emerge in the swarm; suggesting the emergence of reproductive isolation. Finally, we employ different analysis tools from computational biology that further support this claim.

1 Introduction

Embodied evolutionary robotics (EER), aims to design collective behaviors for a swarm of heterogeneous agents evolving online [8]. In a nutshell, these are algorithms in which evolution is carried out in a decentralized manner, where each agent, typically a mobile robot, runs an EA on board and exchanges genetic material with other agents when they meet. Selection and variation are performed locally by the agent. As such, there is no central process that governs evolution, in contrast with traditional evolutionary robotics (ER). These algorithms have been successfully applied in different contexts and have been shown to be robust in open-ended environments [2]. However, due to their distributed nature and the fact that they operate online, some problems remain challenging for EER. For example, evolving task specialization or division of labor has been very challenging and remains an open problem.

It has been shown that evolving specialized behaviors requires reproductive isolation and tailored selection operators [16]. The fact that agents can spread their genetic material to any other agent does not help the emergence and the conservation of specialized behaviors. One way to solve this issue is to limit mating encounters trough geographical isolation (*Allopatric speciation*) where contact between agents from different regions is limited [20]. However, forcing geographical isolation limits the range of problems and environments that the swarm can tackle. One of the objectives of online EER is long-term adaptation in open-ended environments; having such a requirement can be a big limitation.

In the present work, our goal is to propose a new hypothesis that could facilitate behavior specialization. Our main argument can be stated as follows: if reproductive isolation is a requirement to behavior specialization, can this isolation be enforced in other ways than geographically? For instance, can we favor this isolation during the reproduction phase by choosing the "right" genetic material? This type of isolation can be considered as an instance of *Sympatric speciation*, akin to what exists in nature where reproductive isolation evolves within a population sharing the same environment, allowing the divergence into different species [15]. There exist in nature some evidence that behavioral separation can be a plausible mechanism for promoting *Sympatric speciation.* We propose here to explore the idea of promoting isolation by promoting behavioral diversity.

Ever since the introduction of novelty search [14], there has been a large body of work on the idea of searching for originality; disposing of objectives. It opened the way to many clever and powerful algorithms in evolutionary robotics (ER) and later lead to quality diversity (QD) algorithms which elegantly combine the search for novel and fit solutions [21, 17].

Although very efficient in traditional ER, their application in Online collective robotics raises few challenges. For instance, they rely on centralized archives that are incrementally filled with newly discovered behaviors, whereas EER are decentralized and emphasize local information exchange and no history maintenance. Furthermore, the property of EER that is probably the greatest challenge, is the fact that they operate online. Individual behaviors resulting from the interactions of the agents with their environment or other agents, cannot be predicted and cannot be reproduced, which make their comparison difficult.

The search for novelty and the promotion of diversity, have been previously introduced in collective robotics. For instance, [9] applied novelty search on a swarm of agents on an aggregation task. In that instance, the authors used NEAT [22] a centralized off-line algorithm and all agents shared the same controller (homogeneous agents). It showed that novelty can improve exploration for solutions in swarm robotics. In the case of online collective robotics, [12] introduced a decentralized instance of MAP-Elites [17] where agents exchange locally behavioral maps. That work showed that behavioral diversity can be possible without reproductive isolation, however the algorithm needs task specific behavioral descriptors. Furthermore, measuring the behavior of solutions on agents in specific situations may not generalize well to other situations due to the dynamics of the environment.

1.1 Objectives

As said previously, evolving specialized behaviors requires reproductive isolation. In some cases, choosing specially crafted selection operators, and fitness assignment schemes can also help for the emergence of specialization. For instance, [11] showed that using a "market" mechanism that favors the least prevalent behavior by artificially increasing its fitness value, allows the algorithm to balance between two behaviors. One can even argue that the "market" mechanism creates some level of isolation, as it devalues behaviors based on their frequency in the population and then selection pressure limits their spread. The central theme of this paper follows a similar general idea.

We argue that promoting diversity, as it is done in Novelty Search [14] for example, can increase reproductive isolation and thus favors the emergence of specialization. To this end, we propose a selection scheme that promotes diversity by selecting solutions based on their originality in behavioral space. The selection scheme respects the online and distributed nature of EER, it operates and uses generic behavioral descriptors measures locally on the agents. The experiments we present aim at verifying the following hypotheses claiming that:

H.1 it is possible to promote diversity with the proposed selection scheme and,

H.2 promoting diversity favor reproduction isolation.

In the following, we start by describing the $(\mu, 1)$-ONLINE EEA and the different selection schemes we used. We then describe our experimental procedure along with the tested scenarios and finally, conclude with a discussion of the results.

Algorithm 1: $(\mu, 1)$-ONLINE EEA.

```
for $1 \leq j \leq \lambda$ in parallel do
    $\mathbf{x}^j \leftarrow$ random()
    $a^j$ is active
    $\sigma^j \leftarrow \sigma_0$
repeat
    for $1 \leq j \leq \lambda$ in parallel do
        $t \leftarrow 0$, $f^j \leftarrow 0, L^j \leftarrow \emptyset$
        while $t < t_{\max}$ do
            $t \leftarrow t + 1$
            if $a^j$ is active then
                execute($\mathbf{x}^j$)
                update($f^j$)
                if $t > \tau\, t_{\max}$ then
                    broadcast($\mathbf{x}^j$, $f^j$, $\sigma^j$)
            $L^j \leftarrow L^j \cup$ listen()
        if $L^j \neq \emptyset$ then
            $\bar{\mathbf{x}} \leftarrow$ select($L^j$)
            $\sigma \leftarrow$ adapt($\sigma^{\bar{\mathbf{x}}}$)
            $\mathbf{x}^j \leftarrow$ mutate($\bar{\mathbf{x}}, \sigma$)
        else $a^j$ is not active
until termination condition met
```

2 Methods

2.1 The $(\mu, 1)$-ONLINE Embodied EA

The main inspiration of the $(\mu, 1)$-ONLINE EEA is the original version of minimally Environment-driven Distributed EA (mEDEA) [3] to which with we add a selection operator that we will describe later (Algorithm 1). The algorithm considers a swarm of λ mobile agents a^j with $j = 1, \dots, \lambda$ each executing a neuro-controller whose parameters are $\mathbf{x}^j$ (the active genome). Each agent maintains a list L^j, initially empty, in which it stores other genomes that it receives from other agents.

At each time step $t < t_{\max}$, an agent executes its active controller and broadcasts its genome within a limited range. In parallel, it listens for genomes originating from other agents, and when a genome is received (a mating event), it is stored in the agent's list L^j (its local population). This procedure is executed in parallel on all agents during $t_{\max}$ steps, the evaluation period or one generation.

During the evaluation phase, fitness values are assigned to the individual agents based on their own performance with regard to the given task and continuously updated (line 12). The fitness values are transmitted along the genome and stored on the receiving end and if an agent receives an already seen genome, it updates the genome's fitness. To further ensure that fitness values are accurate, a *maturation age* is required of the agent before broadcasting [13]. This is set as $\tau \; t_{\max}$ with $\tau < 1$.

At the end of a generation, agents select a genome $\bar{\mathbf{x}} \in L^j$, and replace their active genome with a mutated copy of the selected one. Their list is then emptied and a new generation begins. In this work, the genome is a vector $\mathbf{x} \in \mathbb{R}^N$, which represents the weights of the neuro-controller. Only the weights undergo evolution (fixed-topology).

Each genome has its own mutation step-size σ whose initial values is σ_0, and when broadcasting, this value is also sent along with the genome and its fitness. On the receiving end, when a genome is selected from the local list it is mutated using its σ value. Step-sizes are adapted before the mutation takes place, where each agent chooses with probability (0.5) to either increase or a decrease the step-size. This rule increases the likelihood that the most adapted value between the two, will survive and spread in the swarm. The update rule is defined as:

$$\sigma = \begin{cases} \min\left(\sigma(1+\gamma),\ \sigma^+\right) & \text{if increase} \\ \max\left(\sigma(1-\gamma),\ \sigma^-\right) & \text{if decrease} \end{cases}$$

where σ^+ and σ^- are the allowable upper and lower bounds and γ some positive constant (Table 1). Mutation is Gaussian using the updated step-size:

$$\mathbf{x}^j := \bar{\mathbf{x}} + \sigma^2 \times \mathcal{N}(1, 0). \tag{1}$$

In the event where an agent had not mating opportunities and finishes its evaluation period with an empty list $L^j = \emptyset$, it becomes inactive; a state during which the agent is motionless and which can last multiple generations. During this period, the inactive agent continues to listen for incoming genomes from other agents passing by, and once $L^j \neq \emptyset$ the agent becomes active again at the beginning of the next generation.

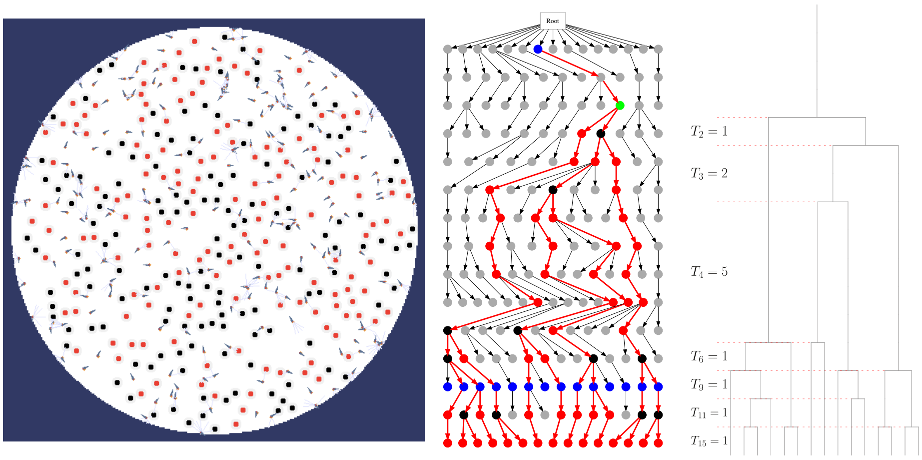


Figure 1: A screenshot of the simulator (left), black and red circles represent items to be collected, the rest are agents. An example of a phylogenetic tree for 15 agents and 14 generations (middle) and the corresponding coalescence tree (right). Gray nodes represent active genomes, black nodes represent coalescence events and the green node the MRCA of the surviving genomes. The generation at which the takeover event occurs is where all active genomes are colored in blue (all descendants of a single initial genome, also colored in blue.) The coalescence tree is a different representation of the sub-tree that is colored in red.

The number of genomes the agents collects $\mu^j = |L^j|$ $(0 \leq \mu^j \leq \lambda)$ is conditioned by its mating encounters. Since the communication range is limited, agents that travel long distances will increase their probability of mating. We should note that mating encounters allow agents to spread their active genome and to collect new genetic material. The algorithm stores only one copy of the same genome[1] if it is received more than once.

The main difference between mEDEA and Algorithm 1 is that we do not consider a listening phase. Agents broadcast if they are active and listen all the time, whereas in mEDEA, agents must be in a specific listening state to record incoming genetic material.

2.2 Seeking new behaviors

In novelty base search [14], the goal of the evolutionary algorithm is to discover solution that exhibit new unseen behaviors. These solutions are recorded in an archive which prevents the algorithm to "rediscover" similar solutions later. New solutions are compared to ones in the archive, and if novel enough they extend the archive. In our context, the notion of novelty, is defined differently. Since agents empty their genome list after selecting a new active genome, we do not have an archive. The selection scheme we propose selects solutions at the agent level by choosing the solutions that have the most different behavior from other solutions in the local list. The open question now is how do we measure

---

[1] The term same is here used in the sense "originating from the same agent".

behavior? Traditionally, selection based on behavioral distances requires the definition of a so-called behavioral descriptor; a set of features that capture the behavior of solutions, in a given environment. These descriptors allow to compute a distance between behaviors which is the basis for selection. These behaviors descriptors fall generally in two categories: 1) task related, requiring expert knowledge about the task or 2) generic descriptors, that can be applied on a large class of problems. This latter category often relies on sensor and actuator data to characterize behaviors [10, 19].

In addition to the lack of an archive, we believe that measuring the behavior during the lifetime of the agent online is not a reliable basis for comparison. Because agents live specific experiences depending on their encounters, it is difficult to generalize a measurement outside from the situation that it was measured in. Consider for example, a genome that realized a given behavior during its lifetime, the same genome on a different vehicle would most certainly exhibit a different behavior. It would have lived a different experience, crossing the path of different agents and obstacles at different times. If we measure its behavior on its first experience, it will not be a reliable measure on which we could base a behavioral distance. Furthermore, since genomes get only one chance to be evaluated, reevaluations in different conditions is not possible.

To overcome these limitations, we propose to estimate the behavior of the genomes regardless of their agents' experience. If the goal is to identify the most different behavior in the agent's selection pool, we propose to simulate behaviors and measure their difference on fake sensory data and use these measurements as a selection criterion. The important step here is to present all genomes in the selection pool the same "fake" sensory data, as if they experienced the same situation, and compare their "would be" response to that fake situation. We use the terms "fake" and "would be" to emphasize the fact that the sensory data is not measured by agents moving in the environment, but randomly generated and, the responses are not executed by the agents, they are just the outputs of the neuro-controller.

To be more specific, let us consider one agent of the swarm, let it be a. We note $L^a = \{\mathbf{x}_1 \dots \mathbf{x}_\mu\}$ the set of genomes it collected during its last lifetime. Our goal is to identify $\bar{\mathbf{x}} \in L^a$ that has the most different behavior.

At the selection step, agent a creates the set $\mathcal{I}$ consisting of K "fake" input vectors such as $\mathcal{I} = \{\mathbf{I}_k | \mathbf{I}_k \sim \text{Uniform(0,1)}^{|s|}\}$, where $|s|$ is the number of sensors and $[0, 1]$ is the range of values they can take. Let $g : \mathbb{R}^N \times [0, 1]^{|s|} \to [-1, 1]^{|e|}$ such as, $\mathbf{O}_{i,k} = g(\mathbf{x}_i, \mathbf{I}_k)$ be the function that computes the "would be" output of a neuro-controller with weights $\mathbf{x}_i$ on "fake" input $\mathbf{I}_k$. We define the sequence $\mathcal{B}_{\mathbf{x}_i} = (\mathbf{O}_{i,1}, \mathbf{O}_{i,2}, \dots, \mathbf{O}_{i,K})$ as the behavior of genome $\mathbf{x}_i$, note that $\mathcal{B}_{\mathbf{x}_i}$ contains $K \times |e|$ individual output values and is considered as a vector. The genome with the most different behavior is defined as the one whose behavior vector has the largest average distance to all others or:

$$\bar{\mathbf{x}} = \underset{\mathbf{x} \in L^a}{\arg\max} \left(\frac{1}{|L^a|} \sum_{\mathbf{y} \in L^a} ||\mathcal{B}_{\mathbf{x}} - \mathcal{B}_{\mathbf{y}}|| \right).$$

At each generation, each agent generates its own set $\mathcal{I}$, present it to all genomes in its list and then selects its next active genome using the above distance. In the following, we name this selection scheme (BS) for behavioral selection.

Table 1: Simulation parameters.

Arena diam.	1000 pix.	λ	200
Nb. items	300	$t_{\max}$	2000 tics
Sens./com. range	16 pix.	$g_{\max}$	500 generations
Agent/item diam.	6 pix.	σ^0	0.25
Max trs. vel.	2 pix. / tic	(σ^-, σ^+)	(0.01, 0.5)
Max rot. vel.	30 deg / tic	γ	0.35
Init. weights	$[-2, 2]$	τ	0.2
Nb. runs	64	Nb. fake inputs	128

2.2.1 Objective based selection

As a baseline for the experiments, we use a traditional fitness based selection scheme. Different fitness-based selection operators have been applied in EER contexts and it has been shown in multiple instances that their level of selection pressure is correlated with the performance of the swarm on the task [7]. Here, we chose two objective base selection schemes: on the one hand fitness proportionate selection (FPS) whose benefits on behavior specialization have been discussed in [4] and on the other hand, and elitist selection (ES).

2.2.2 Multi-objective selection

Having defined the above selection schemes, here we follow the ideas of [18] and introduce an instance of "diversity selection" in which selection is based on multi-objective criteria using fitness and novelty (MOBS). In this case, the set of Pareto-optimal solutions[2], with respect to fitness and novelty, in the agent's local population constitutes the selection pool. The agent then selects a random genome from the front. Here the novelty objective is the same as defined above.

3 Experiments

3.1 Simulation

We considered a concurrent foraging task in which agents have to collect items placed randomly in the environment (Figure 1). There exist two types of items (red and black) and agents are rewarded one unit of fitness every time they collect one item regardless of its type. This task has been extensively studied in the context of task specialization [11, 12]. Collecting one type of items or the other is considered two distinct tasks (although similar in nature), because agents perceive the types of items trough different sensors.

The experiments were performed on the Roborobo simulator [5] an environment that allows to run experiments on large swarms of agents[3]. In this simulator, agents are e-puck like mobile robots with limited range sensors and two differential drive wheels. Sensors are placed at 12 locations around the agent's body (7 facing the front, 3 facing the back and two facing each side). Agents

---

[2] Recall that the Pareto set is the set of all non-dominated solutions and that, x dominates y imply that x in is not worse than y with respect to all objectives and x is strictly better than y with respect to at least one objective.

[3] Roborobo3 at commit `f108c030f51a991e8fabd92aaaecb87d5ad7032a`

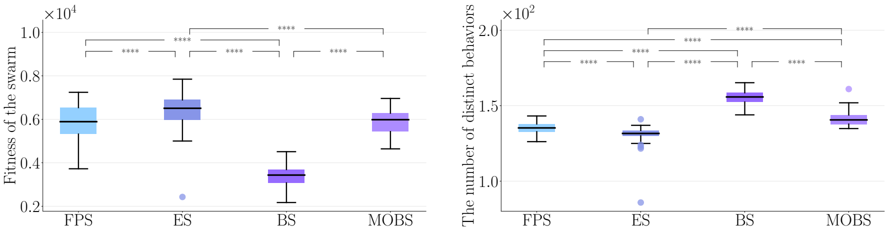


Figure 2: The fitness of the swarm (left) and the number of distinct behaviors (right). Box-plots of 64 values (runs) where individual values in each are averages over all generation. Statistical significance indicated by the number of asterisks (n asterisks indicate a p-value $< 10^{-n}$).

can perceive 4 different things in the environment: obstacles, other agents and the 2 types of items, and can move using 2 motors. All sensor values are in $[0, 1]$ and actuator values are in $[-1, 1]$. To collect items, agents must bump onto them, and when collected, the items reappear at some random location. There is always the same number of items in the environment.

The neuro-controller we consider here is a simple feed-forward perceptron with a hyperbolic tangent activation function. The genome encodes the weights ([1 bias neuron + 12 sensors × (2 types of items + 1 agent + 1 obstacle)] × 2 outputs = 98 weights) of this controller as a real vector.

All instances of the algorithm[4] were run on the exact same environment, with the same conditions and runs were repeated 64 times. All the parameters of the experiments are summarized in Table 1.

3.2 Measures

We present here the measures that were used in the experiment. To estimate the quality of the foraging, we define the swarm fitness at generation g as $\hat{f}(g) = \sum_{j=1}^{\lambda} f^j(g)$ where $f^j(g)$ is the fitness of agent j at generation g.

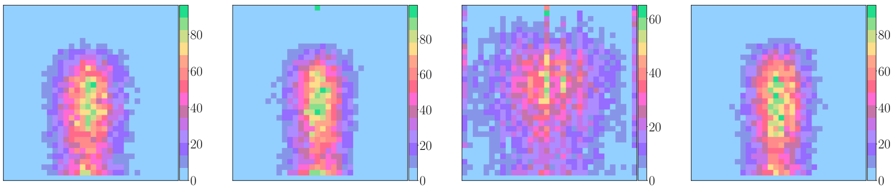
3.2.1 Assessing behavioral diversity.

To measure the behavioral diversity of the swarm, we use two behavioral descriptors: the ratios of red to black items collected and the max distances traveled. The former measures the diversity in terms of items collected, and the latter in terms of exploration range. This range is measured as the maximum distance the agent had during its lifetime from its starting position[5]. These behavioral descriptors are discretized into a two-dimensional map on which the agents of the swarm are binned; i.e. each cell of the map counts the number of agents that expressed the behavior represented by the cell's coordinates. The larger the number of occupied cells in the map, the larger the number of distinct behaviors in the swarm. These types of behavioral measures were also used in [12]. They are not considered during the selection step of the algorithm, they are computed off-line, after the simulation.

---

[4] The code can be downloaded at: `https://gitlab.inria.fr/boumaza/public-code`

[5] The exploration range is bounded by the diameter of environment



FPS ES BS MOBS

Figure 3: Behavioral Maps with 32 × 32 bins. x-axis represent the item type ratio (leftmost bins represent behaviors that collect only black items, those that collect equally are in the middle, and those that collect only red are on the right) the y-axis represent the maximum traveled distance (zero distance at the topmost, maximum distance at the bottom). Maps aggregate behaviors from the last 10 generations.

3.2.2 Assessing diversity using phylogeny.

One way to measure if the population maintains its diversity throughout time is to inspect the population dynamics. For that, we take a gene perspective and study their genealogy independently of the agents. At each generation of Algorithm 1, we record the descendants of the genes and construct a *phylogenetic tree*. Nodes in this tree represent active genomes (those that get selected on agents) and edges represent parenthood relations (child nodes are one mutation away from their parents). Edges relate genomes regardless of their vehicles; i.e., parents and offspring may have existed on different agents.

All the initial genomes (the ones created randomly) are children of a “root” node[6]. Each node, that is neither the root nor an initial genome, has at most one parent, and can have at most λ children (the number of agents). Finally, the depth of the tree is at most $g_{\max}$, and at each level there are at most λ nodes. Since inactive agents do not have active genomes, they are not present in the tree. A simplified phylogenetic tree with 15 initial genes over 14 generations is shown in the center of Figure 1. From a phylogenetic tree, we extract the corresponding *coalescence tree* which synthesize the genealogy of the last surviving genomes (the population at the last generation). This tree is constructed from the last generation up to the first where, at each step lineages are merged whenever two or more genomes share the same parent (a coalescence event).

The most recent common ancestor (MRCA) is the youngest genome from which all the end-survivors descend. Given a phylogenetic tree, it can be identified going back in time from the lowest leaves up following their ancestors until we find the sole ancestor which can either be, a regular node or the root node. The time of the most recent common ancestor (TMRCA) is the height of the MRCA (12 on Figure 1). This time indicates the genetic closeness, since it is related to the number of generations and thus the number of mutations between the MRCA and a current population. The larger it is, the farthest apart are the genomes in the population, the smaller it is, the more its descendants are related. Another measure we use, is the branch lengths of the coalescence tree.

---

[6] This allows to construct a connected graph to ease the analysis and does not affect the results.

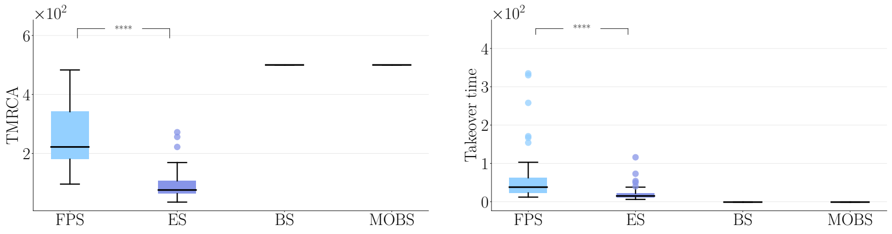


Figure 4: TMRCA (left) and Takeover time (right). Flat boxes indicate the event did not occur. Box-plots of 64 values (runs) where individual values in each are averages over all generation. Statistical significance indicated by the number of asterisks (n asterisks indicate a p-value $< 10^{-n}$.

It gives a sense of how much history genomes share. Genomes would share the least history if they come from a common ancestor far back in time and then evolved along distinct lineages. Furthermore, the less history genomes share, the more reproductive isolation is present. Assuming there are $r \leq \lambda$ survivors at the last generation, the branch length is defined as:

$$L = \sum_{k=2}^{r} k \cdot T_k, \tag{2}$$

where T_k is the number of generations there were k distinct lineages in the tree (Figure 1, right).

Finally, we define the takeover time as the generation at which all active gnomes in the population descend from only one initial genome [6, 1]. This can be seen on Figure 1 where the takeover event happens at the 12^{th} generation. Related to this, we also measure the survival rate of the initial population throughout the generations, which is defined as the proportion of the initial genomes that have offspring at depth d.

3.2.3 Statistical Significance

To compare all selection schemes, each instance is executed 64 times, and we compare the above measures using median values and percentiles. Furthermore, we perform a pairwise comparison using a Mann-Whitney U test with the null hypothesis being "*the samples of the results of both instances are drawn from the same distribution*". We establish that instance "A" outperforms instance "B" on a given measure, if the median value for instance "A" is better than the one for "B" and there is significant statistical difference between the distributions. The level of confidence at which we reject the null hypothesis is indicated by the number of asterisks on the figures, when applicable.

4 Results and Discussion

We begin our analysis by considering the population fitness on all instances (Figure 2, left). Recall that the fitness of one agent is the number of items it collects during its lifetime regardless of their color. As expected, the instance with the highest selection pressure (ES) evolves a swarm that collects the most items. If we look at the fitness proportionate selection (FPS), it comes in second, gathering slightly fewer items (the difference is statistically significant). This result is not new, for the collection task, several authors reported that increasing the selection pressure increases the fitness, see for example [7]. The instance that performed the worst is the one that disregards fitness (BS) which is also expected. In the middle comes the multi-objective selection schemes (MOBS) that performs as good as fitness proportionate selection in terms of the task.

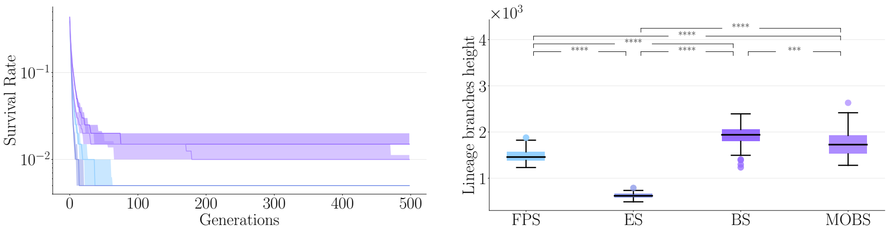


Figure 5: Survival rate (left) and branch length (right). Curve on the left represent median (solid line) and the range between the 25$^{\text{th}}$ and the 75$^{\text{th}}$ percentiles. Their colors follow the same scheme as the box-plots on the right. Statistical significance indicated by the number of asterisks (n asterisks indicate a p-value $< 10^{-n}$).

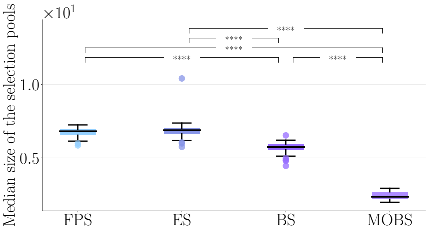
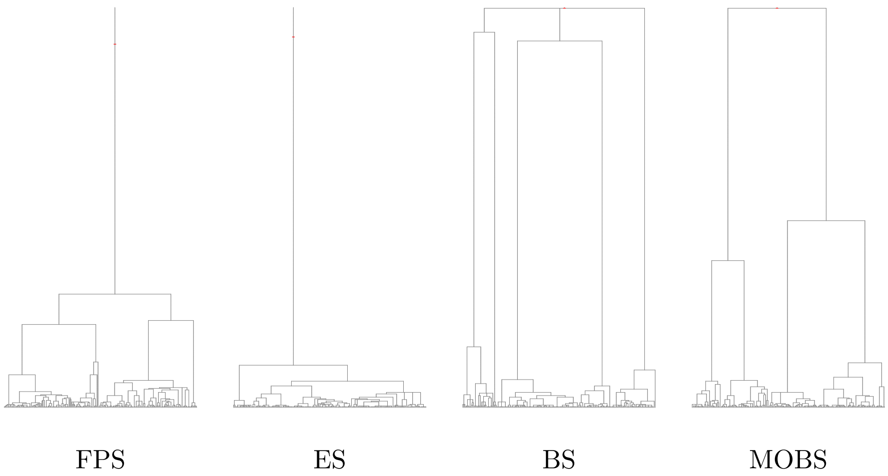


Figure 6: The size of the selection pool. Box-plots of 64 values (runs) where individual values in each are averages over all generation.

4.0.1 Does behavioral-selection promote diversity?

We start by looking at the number of distinct behaviors in the swarm (Figure 2, right). Here we see that behavioral-selection (BS) play its role and the number of distinct behaviors in this case is significantly higher than in the other instances. On the other hand, ES is the instance that create the least diversity. Here again MOBS comes in the middle.

The number of distinct behaviors and their characterization is better viewed on the behavioral maps (Figure 3). Here we see that the behavioral-selection scheme creates diverse behaviors on all the spectrum. We also notice that there are two groups of agents that only collect items of one type (leftmost or rightmost bins). Some agents specialize in collecting red items and other in collecting black items.



FPS ES BS MOBS

Figure 7: Coalescence trees of typical runs for each selection instance.

On the other hand, the behavioral maps of the other instances (ES, FPS, MOBS) do not show specialized behaviors. They are mostly clustered in the middle (ratio of $\frac{1}{2}$); behavioral specialization did not occur. Furthermore, in all instances, there is a group of agents that have a large exploration range (bottom of the map). This is the algorithm overcoming environmental pressure: controllers that travel long distances increase their chances of mating and survival.

4.0.2 Does behavioral-selection promote isolation?

To answer this question, we compare the instances on the basis of the coalescence measures we described above. If we inspect the TMRCA (Figure 4, left), it is very clear that the most elitist instance (ES) has the lowest TMRCA. At each generation, only the few (locally) most fit solutions survive, reducing the genetic pool at the swarm level. FPS on the other hand allows (with a small probability) less fit solutions to survive which delays the appearance of the common ancestor. On the other hand, in all diversity-selection instances, there was no MRCA! This is indicated by the flat box-plot at the value 500 (the total number of generation). This reflects the fact that in those instances, there were distinct lineages that coexisted since the start of the simulation that never coalesced. Said differently, there were subsets of genomes that never shared any genetic material, which we may qualify as niches. This is an indication that reproductive isolation occurred.

How many niches where there? To answer this question, we can inspect the survival rate curves (Figure 5, left). The curves represent the rate, out of λ, of the initial solutions that have decedents at a given generation. We can see that for ES and FPS, the rate drops rapidly to $\frac{1}{\lambda}$. However for the diversity-selection instances, the drop is less rapid and converges between $\frac{2}{\lambda}$ and $\frac{5}{\lambda}$; i.e., between 2 or 5 niches. The rate at which the survival curves drops, indicates the rate at which the population becomes more homogeneous. This can also be seen if we look at the takeover-time (Figure 4, right). In the case of ES, the population is quickly taken-over by one of the initial solutions and in the case of FPS, the takeover happens a bit later. In the case of diversity-selection the takeover event never happened and, incidentally, this observation can be drawn directly from the lack of MRCA above. Finally, if we look at the branch length (Figure 5, right), we notice that the higher values are those for BS and MOBS indicating that the end-survivors of these instances share the least genetic history. To illustrate the above results, Figure 7 shows coalescence trees from typical runs of all the instances.

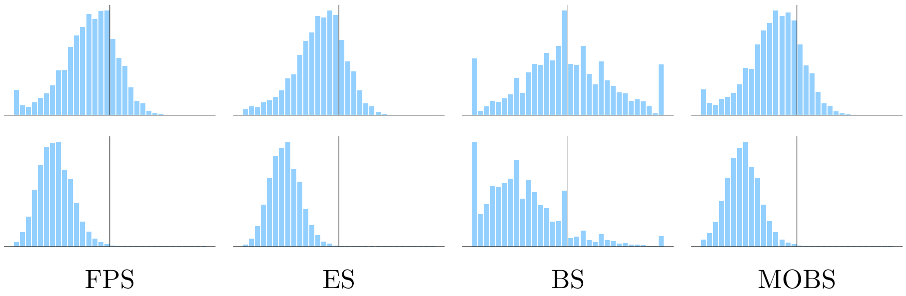


Figure 8: Histogram of red items ratios across the population. Each bar counts the number of agents collecting at that ratio. The range $[0, 1]$ is clustered in 32 bins. The vertical black line at value 0.5 indicates that equal amounts of black and red items are collected. The histogram aggregate counts from the final 10 generations. The top row is the case where black items reward more, and bottom row in the case were black items are more abundant.

We were puzzled at the poor performance in terms of diversity of the multi-objective instance. The phylogenetic measures indicate that some level of isolation occurred, however it may not be a direct result of behavioral diversity, this is at least not reflected in the behavioral maps. These algorithms have been reported to perform well on different settings, how come it is not the case here? We believe that the answer to this question lies in the size of the local populations collected by the agents. Indeed, these populations contain few solutions compared to the swarm size, and when the Pareto set is extracted, the selection pool becomes even smaller (Figure 6). Sampling randomly the next solution from such a small set does not leave enough room for creativity. Furthermore, since both the fittest and the most different solutions in the local population, are part of the Pareto set[7], on very small samples, the algorithm behaves as if it were a mix of both ES and BS.

Finally, fitness proportionate selection has been reported as very important in promoting behavioral specialization [4]. Why was it not the case here? The answer at this stage is still speculative and need further investigations. We believe that it is probably the case i.e., FPS favors behavioral specialization. However, for it to do so, may require reproductive isolation. In our simulation, there was none since our hypothesis was that it emerges from diversity. In [4], reproductive isolation was enforced geographically.

---

[7] They are both strictly better than the others on one objective.

4.1 Increasing the pressure from the environment

In the following, we briefly discuss cases where we modified the environment to create a bias toward favoring one of the two item types. Our goal is to investigate if diversity selection can cope with the added environmental pressure. On the first environment, we modified the reward system, where before, collecting items brings one unit of fitness regardless of their type, now black items reward more, 10 times more. On the second environment, we modified the proportions of each item type, now 75% of items are black and the rest are red. In both these modified environments, we expect the swarm will "prefer" to collect more black items since they are either more abundant thus easily found, or more rewarding. This was the case and can be seen on Figure 8 which presents the distribution of red item ratios across the population. In all instances with a task selection pressure (FPS, ES, MOBS) the swarm tends to favor black items and no agent specialized on red (no bars on the right). However, the behavioral-selection instance (BS) is, to some extent, able to cope with the added environmental pressure since some agents specialize on only red items.

5 Conclusions

It has been shown that in order to evolve specialized behaviors, reproductive isolation must be present. This isolation, can either be enforced geographically (Allopatric speciation) in the environment or through some other mechanism that limits reproduction between species sharing the same environment (Sympatric speciation). We proposed the idea of favoring isolation of the latter kind through the use of diversity-selection schemes and tested the hypothesis on a concurrent foraging task. The results suggest that it is the case, when diversity is enforced, a proportion of the population specialized on one of the two possible tasks. In order to measure behaviors reliably, we introduced a procedure that estimates behavioral distances on board the agents. This procedure compares the outputs of candidate controllers on the same reference input data. To verify if indeed isolation occurred, we used phylogenetic measures borrowed from coalescence theory. We also briefly investigate the resilience of the selection schemes in situations with added environmental pressure and the results were very promising.

The next steps are to test if the conclusions hold on different tasks and to further investigate the multi-objective selection method, as we believe it could be a good candidate to promote diversity and quality in EER settings. However, the issue of the archive or lack of archive, needs to be addressed. One possibility to increase the selection pool, could be by extending the lifetime or the number of agents which will increase the number of mating encounters. Furthermore, extending the behavioral comparison to neural architecture other than feed-forward networks can also be interesting to tackle more challenging tasks.

References

[1] Bäck, T.: Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In: Michalewicz, Z., Schaffer, J.D., Schwefel, H.P.,

Fogel, D.B., Kitano, H. (eds.) Proceedings of the First IEEE International Conference on Evolutionary Computation. pp. 57–62 (1994)

[2] Bredeche, N., Haasdijk, E., Prieto, A.: Embodied evolution in collective robotics: A review. Front, in Robo. and AI **5**, 12 (2018)

[3] Bredeche, N., Montanier, J.M.: Environment-driven Embodied Evolution in a Population of Autonomous Agents. In: PPSN 2010. pp. 290–299. Krakow (2010)

[4] Bredeche, N., Montanier, J.M., Carrignon, S.: Benefits of proportionate selection in embodied evolution: A case study with behavioural specialization (2017)

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

[6] E.Goldberg, D., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. Foundations of Genetic Algorithms, vol. 1, pp. 69–93. Elsevier (1991). https://doi.org/https://doi.org/10.1016/B978-0-08-050684-5.50008-2

[7] Fernández Pèrez, I.n., Boumaza, A., Charpillet, F.: Comparison of selection methods in on-line distributed evolutionary robotics. In: Proc. of Alife'14. pp. 282–289. MIT Press, New York (2014)

[8] Ficici, S., Watson, R., Pollack, J.: Embodied evolution: A response to challenges in evolutionary robotics. In: Proc. of the 8^{th} Europ. Wrks. on Learning Robots (1999)

[9] Gomes, J., Urbano, P., Christensen, A.L.: Introducing novelty search in evolutionary swarm robotics. In: Swarm Intelligence. pp. 85–96. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)

[10] Gomez, F.J.: Sustaining diversity using behavioral information distance. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. p. 113–120. Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1569901.1569918

[11] Haasdijk, E., Bredeche, N., Eiben, A.E.: Combining Environment-Driven Adaptation and Task-Driven Optimisation in Evolutionary Robotics. PLoS ONE **9**(6), e98466 (Jun 2014). https://doi.org/10.1371/journal.pone.0098466

[12] Hart, E., Steyven, A.S.W., Paechter, B.: Evolution of a functionally diverse swarm via a novel decentralised quality-diversity algorithm. In: Proc. of the Genetic and Evolutionary Computation Conference, Kyoto, Japan, July 15-19. pp. 101–108. ACM (2018). https://doi.org/10.1145/3205455.3205481

[13] Karafotias, G., Haasdijk, E., Eiben, A.E.: An algorithm for distributed online, on-board evolutionary robotics. In: Proc. of GECCO '11. pp. 171–178. ACM (2011)

[14] Lehman, J., Stanley, K.O.: Abandoning objectives: Evolution through the search for novelty alone. Evolutionary Computation **19**(2), 189–223 (Jun 2011). https://doi.org/10.1162/EVCO_a_00025

[15] Maynard Smith, J.: Sympatric speciation. The American Naturalist **100**(916) (1966)

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

[17] Mouret, J., Clune, J.: Illuminating search spaces by mapping elites. CoRR **abs/1504.04909** (2015), `http://arxiv.org/abs/1504.04909`

[18] Mouret, J.B., Doncieux, S.: Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In: Eleventh conference on Congress on Evolutionary Computation (CEC'09). pp. 1161–1168. IEEE Press, Trondheim, Norway (2009), `https://hal.archives-ouvertes.fr/hal-00473147`

[19] Mouret, J.B., Doncieux, S.: Encouraging behavioral diversity in evolutionary robotics: An empirical study. Evolutionary Computation **20**(1), 91–133 (2012)

[20] Prieto, A., Becerra, J.A., Bellas, F., Duro, R.J.: Open-ended evolution as a means to self-organize heterogeneous multi-robot systems in real time. Rob. and Auto. Sys. **58**(12), 1282 – 1291 (2010)

[21] Pugh, J.K., Soros, L.B., Stanley, K.O.: Quality diversity: A new frontier for evolutionary computation. Frontiers in Robotics and AI **3**, 40 (2016)

[22] Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation **10**(2), 99–127 (Jun 2002)