Learning to Cooperate in a Socially Optimal Way in Swarm Robotics

Paul Ecoffet[1], Jean-Baptiste André[2], Nicolas Bredeche[1]

[1]Sorbonne Université, CNRS, Institut des Systèmes Intelligents et de Robotique, Paris, France
[2]Institut Jean Nicod, Département d'études cognitives, ENS, EHESS, PSL Research University, CNRS, Paris France
paul.ecoffet@sorbonne-universite.fr

Abstract

This paper addresses the problem of learning cooperative strategies in swarm robotics. We are interested in heterogeneous swarms, in which each robot optimizes its individual gain. For some tasks, the problem is that the optimal strategy requires to cooperate and may be counter-selected in favor of a more stable but less efficient selfish strategy. To solve this problem, we introduce a mechanism of partner choice, which conditions of operation are learned. This mechanism proves surprisingly efficient, when the swarm size is large, and the duration of interactions is long. Beyond evolutionary swarm robotics, the results we present are relevant for other distributed on-line learning methods for robotics, and as a possible extension of existing evolutionary biology and social learning models.

Introduction

Nature abounds with impressive examples of swarm intelligence (Camazine et al., 2001), which have motivated researchers in robotics to devise methods to obtain self-organizing behaviors in collective of robots (Mataric, 1992; Beni, 2005; Brambilla et al., 2012; Bayindir, 2016; Hamann, 2018).

However, designing the behaviors of a robot swarm poses a challenge in itself as collective behaviors emerge from the multitude of interactions between robots, and are therefore difficult to predict and design. The use of automatic design methods can circumvent this problem to some extent, but is based on constraining assumptions as they generally assume that (1) the collective payoff is known and available and (2) the learning algorithm can be iterated in a centralized way.

This is the case in multi-agent reinforcement learning methods for decision making under uncertainty (Amato et al., 2015) and in evolutionary swarm robotics (Trianni, 2008). In the latter, these two assumptions enable to use homogeneous populations, ie. swarms of clones (Hauert et al., 2014; Trianni et al., 2007). Learning with clonal populations have been shown to provide several advantages: it can lead to purely altruistic behaviors (Waibel et al., 2011), it can deal well with credit assignment (Waibel et al., 2009), and it allows the acquisition of specialized behaviors even though all robots share the same control parameters (Tuci and Trianni, 2014; Ferrante et al., 2015).

In this paper, we lift the two previously mentioned hypotheses. We are interested in a population where all individuals are different and get individual payoffs (without knowing about the global payoff). While we use a classic evolutionary algorithm scheme, this setup is relevant for two other learning settings: individual learning facing collective tasks (Fudenberg, 1998) and distributed on-line reinforcement learning (Bredeche et al., 2018; Heinerman et al., 2015). In the class of problems addressed by either methods, cooperation is possible only when the individual's objective is aligned with the global objective, which requires a carefully designed individual objective function.

As this may not always be the case, we address the following question in this paper: **how to enable each robot in a swarm to learn the *socially* optimal behavior when this behavior is *individually* sub-optimal?**

This problem has been extensively studied in game theory and evolutionary biology (Axelrod and Hamilton, 1981). Whenever the accomplishment of a task by a group of individuals is *not* aligned with each individual's objective, maximizing one's own payoff will interfere with the execution of the collective task unless explicitly constrained otherwise.

Several mechanisms have been identified that allow the alignment between the individual's and the global objectives (West et al., 2007). Among these mechanisms, partner choice is revealed to be particularly efficient. If all individuals can choose with whom to cooperate, then it is in everyone's interest not only to choose the best partner, but also to cooperate so as to be chosen. In other words, there is a selection pressure that favours those individuals who are able to make a good compromise between self-interest and common interest.

Earlier results obtained in theoretical biology have shown that for partner choice to be effective, the time spent searching for a partner compared to the time spent interacting with partners should be as short as possible (Debove et al., 2015). However, all studies so far consider tightly controlled conditions, either with learning partner choice occurring in a

well-mixed population (McNamara et al., 2008) or with the agents moving in a discrete grid world but without learning (Aktipis, 2011).

We propose an implementation of the mechanism of partner choice for evolutionary swarm robotics, the use of which is learned by the robots depending on the interactions between robots and the task to be performed. We also propose a study of the necessary conditions for partner choice to enable the learning of a socially optimal cooperative strategy when such a strategy is not naturally stable (i.e. not a Nash Equilibrium). In agreement with theoretical results from evolutionary biology, we show that the use of partner choice in a robot swarm can shift the Nash Equilibrium from using a sub-optimal defective strategy to using a cooperative strategy. However, we also show that severe constraints over the number of encounters and the duration of interactions are key order parameters to enable the learning of socially optimal cooperative strategies.

Methods

Environment

We define a collective foraging task where N robots move and consume resources in a circular arena (see Fig. 1). The environment in which the robots move is continuous. The robots are subject to a simple kinematic model, and can control their translation speed and angular speed. Resources are small objects spread randomly throughout the arena, and can be consumed *only* if two robots are into contact with the resource at the same moment. Once a resource is consumed, it disappears and a new resource appears at a random location in the environment to ensure that the density of resources in the environment remains constant over time.

In order to consume a resource, both robots in a pair must invest some amount of energy, which is learned and may differ from one robot to another (see Section Learning). Each robot receives a payoff based on its own investment and that of its partner, for each resource harvested. This means that each robot has to make a compromise between the effort made to harvest the resource and the expected payoff.

We define the experimental setup so that a robot may either cheat (minimizing its own investment while maximizing its gain) or cooperate (maximizing the gain of the pair). This is achieved by defining a payoff function such that the Nash Equilibrium corresponds to a selfish behavior, while the social optimum where robots cooperate is not stable (see Section Payoff function).

Payoff function

When two robots interact with each other, they earn a gain determined by the investment of the two agents. The gain of an agent a_i investing x_i with its partner a_j investing x_j is determined by the function $P(x_i, x_j)$ described in the equation 4.

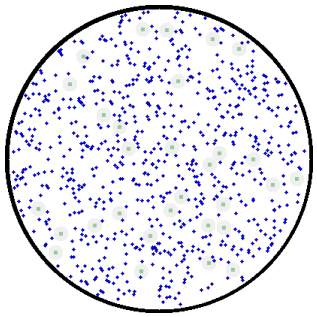

Figure 1: The environment is a circle arena. Blue dots are robots. Green dots are resources. Robots can see the resources, and when two robots are close enough (light grey area), they may interact together to forage the resource. The Roborobo simulator is used (Bredeche et al., 2013).

$$PG(x_i, x_j) = \frac{a}{2}(x_i + x_j) \quad (1)$$

$$PD(x_j) = \frac{b}{2}(x_j) \quad (2)$$

$$C(x_i) = \frac{1}{2}x_i^2 \quad (3)$$

$$P(x_i, x_j) = PG(x_i, x_j) + PD(x_j) - C(x_i) \quad (4)$$

This function is a mixture of a public good (PG, modulated by a) and a prisoner's dilemma (PD, modulated by b) and a quadratic cost C. For a_i to maximize its individual gain ($P(x_i, x_j)$), the optimal investment is $x_d = \frac{a}{2}$, which corresponds to the defective behavior. For the group to maximize their total gain, both agents must invest $\hat{x} = a + \frac{b}{2}$, which corresponds to the cooperative behavior. By using a combination of two classical social dilemma games, we ensure that (i) the optimal selfish individual investment x_d is greater than zero (which remove possible boundary effects) and (ii) there is a clear-cut difference between cooperative and selfish strategies.

Figure 2 plots the payoff function with different partner's investment values. The maximum payoff for the focal robot is always obtained for $x_d = \frac{a}{2}$, notwithstanding the contribution of the partner. However, if both agents play the same strategy, the maximum payoff for the group is found at $\hat{x} = a + \frac{b}{2}$. We define x_d as the contribution corresponding to a defective strategy, and $\hat{x}$ as corresponding to a cooperative strategy.

In the following, we fixed a and b so that the focal robot will have to invest a value of $\hat{x} = 6.5$ in order to hope to obtain the best possible collective gain, but this gain will only be obtained if her partner consents to the same effort (cf. green curve). However, in the presence of a cooperative partner, it is more interesting for the focal robot to cheat by investing less. In this case, the optimal investment value of our

cheating robot is $x_d = 2.5$ (cf. orange curve). Nevertheless, in this case, the partner has no interest in cooperating, and both of our cheaters will eventually get a sub-optimal gain (cf. blue curve). Therefore, the latter situation arises that the only possible Nash Equilibrium is sub-optimal: robots could both get more *and* a robot cannot deviate from this strategy without a loss.

For clarity, we use the following terms in the rest of the paper: a robot investing $x_d = 2.5$ will be said to play a *defective strategy*. A robot investing $\hat{x} = 6.5$ will be said to play the *"optimal cooperative strategy"*. Any robot playing $x > 2.5$ will be said to play a *cooperative strategy*, even if it is not the optimal one.

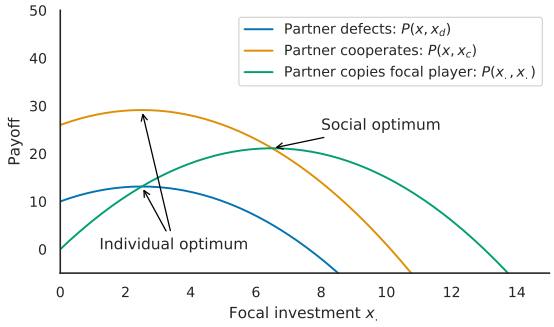

Figure 2: Payoff function with different partner's investment value. The individually optimal investment is $x_d = \frac{a}{2}$ whatever the constant value the partner invests, which corresponds to a defective strategy. If both robots invest the same value, then the socially optimal investment is $\hat{x} = a + \frac{b}{2}$, which corresponds to a cooperative strategy behavior.

Partner Choice

We give our robots the ability to perform partner choice as a cooperative mechanism to solve the social dilemma we just described. Partner choice makes it possible to escape the sub-optimal selfish behavior of partners by enforcing individuals to act as "good" partners, rather than just optimizing their own self interest. This is made possible by setting up a game during which potential partners announce their respective investment in advance, allowing everyone to decide whether or not to continue the cooperation. As a result, it can supposedly lead to shifting the Nash Equilibrium to the socially optimal strategy (ie. both partners must cooperate).

Partner choice has been studied in theoretical biology. (Debove et al., 2015) explored learning partner choice in repeated ultimatum games with both field studies with humans and numerical simulations. They showed that the efficiency of partner choice depends on the **meeting probability** of an agent (β) and the **split probability** of an interaction (τ). Both β and τ are expressed from the viewpoint of a focal agent. The meeting probability β determines how fast an agent will find a potential partner (whether it then chooses to interact or not). If the pair interacts, the split probability τ is used to determine when the interaction ends (smaller τ means longer interactions).

If the meeting probability is large compared to the split probability, i.e. β/τ is large, then partner choice is a viable strategy and can emerge. Indeed, for partner choice to be effective, when an agent refuses to interact with a partner, it must do so because its expectation of gain in finding a better partner outweighs the gain missed by rejecting the interaction with the wrong partner and the implied cost paid by looking for a new partner. Thus, if search time is short (ie. larger β) compared to interaction time (ie. smaller τ), it is profitable to spend more time searching for a good partner than interacting with more uncooperative partners.

The β parameter is determined by the ability of the robots to meet on a resource and varies as the robots evolve, but also depending on the density of robots in the arena, and especially the robots that are also seeking for a partner. Both β and τ are fixed in theoretical models. This is not the case in our robotic model, as the meeting probability β is indirectly controlled by how experimental conditions are set (in particular the number of robots and the size of the arena) *and* by how the robots move, which is learned.

Robotic Behaviors

We consider a swarm of heterogeneous robots, meaning that robots may act differently when facing a similar situation depending on their personal learning experience.

Each robot alternates between two behaviors. Firstly, a **foraging** behavior, which is learned. It is in charge of both exploration and partner choice. It will be described in Sections Controller and Representation and Learning; Secondly, a **wandering** behavior, which is hard-coded, that simply moves the robot forward while avoiding obstacles of any kind (walls, resources and other robots alike). This behavior is used *for some time* after a resource has been successfully harvested (see details below).

Partner choice is implemented such that when two robots meet on a resource, each robot executes the following algorithm:

1. The robot announces to its potential partner the effort it is willing to make to capture the resource. This is when the robot can choose to cooperate (to maximize the overall gain) or cheat (to maximize its own gain at the expense of its partner);

2. Based on the effort announced by its potential partner, the robot decides whether to pursue the interaction further;

3. If (and only if) both robots agree to continue the interaction, the resource is harvested and each robot's payoff is computed accordingly. The robot's payoff then depends on both its own investment and the overall investment. As

such, two robots may have different payoffs depending on each individual effort.

Whether the interaction is successful or not, the resource disappears and is relocated elsewhere. In case of a successful interaction, both robots switch to the *ad hoc* wandering behavior for a certain period of time which depends on the split probability τ (cf. Section Partner Choice), before returning to its nominal (learned) behavior. The larger the τ value, the faster the robot should start to search for a new resource.

While wandering, the robot no longer participates in the cooperative game as it only avoids obstacles (for example, this corresponds to time taken to process the resource (e.g.: digesting or retrieving the resource). At each time step, each wandering robot has a probability of τ to switch back to the foraging behavior. The wandering behavior is used to simulate the expected duration of an interaction, which value is thus $1/\tau$. τ is fixed for a given experiment, and the influence of several particular values will be explored in Section Results. In the particular case where $\tau = 0$ (ie. interaction time is infinite), then robots that both accepted to interact will be wandering around until the end of the current generation.

Controller and Representation

The robots' control architecture is decomposed, for each robot, into three parts: (1) The **investment value** represents what the robot is willing to pay to cooperate. It is defined in $x \in [0, 10]$; (2) Control parameters for the **partner choice module**. It is used when a robot pairs with another to harvest a resource, to decide whether to accept interaction or not depending on each partner's investment values; (3) Control parameters for the **movement module**. It is used to move the robot around (e.g. avoid obstacles, finding a resource, finding a partner).

Both modules are artificial discrete neural networks, using a $\tanh$ activation function. The details of the inputs of each network are given in Table 1. Both networks takes an additional bias value of 1.0 as input. The bias neuron projects on the hidden and output neurons.

The partner choice module is active only when a robot pairs with another on a resource. It takes three input values: the robot's investment value, that of its partner, and a bias neuron. The hidden layer is composed of 3 neurons, and the network produces a single output value ($a \in]-1, 1[$), which determines if cooperation is accepted ($a > 0$) or not ($a <= 0$). The hidden layer is composed of three neurons.

The movement module takes $8 * 4$ sensory input neurons and a bias neuron as input. These are projected on one hidden layer with 10 neurons. Then, two output neurons are defined in $(-1, 1)$, and scaled to determine the proper translation and rotation speeds. The $8 * 4$ sensory inputs are provided by 8 sensors, placed uniformly around the robots. Each sensor gives four elements of information: (a) distance to nearby robot (if any), (b) distance to wall (if any), (c) distance to resource (if any) and (d) if a resource is detected, presence of another robot on the resource ($= 1$) or not ($= 0$).

Input	Value
Movement module	
Per sensor ($\times 8$)	
Distance to Robot	$[0, 1)$ if in range else 1
Distance to Wall	$[0, 1)$ if in range else 1
Distance to Resource	$[0, 1)$ if in range else 1
Partner on the Resource	0 or 1
Partner choice module	
Partner's investment	$[0, 10)$
Robot's own investment	$[0, 10)$

Table 1: Neural Networks inputs

Learning

Learning is performed for all individuals, using an evolutionary algorithms as a direct policy search method (Doncieux et al., 2015). Neural weights and investment value are optimized according to an objective function that rewards the ability to forage resources. Each robot is described by its own unique genome containing 369 values, decomposed as follow: (1) the neural weights of for the movement module, i.e. 352 real values, with each value initialized in the range $[-1, 1)$ and bounded in $[-10, 10]$ throughout learning; (2) the neural weights for the partner choice module, i.e. 16 real values, initialized and bounded similarly; (3) the investment value g_x, a real value defined in $[0, 1)$. At run time, the investment level x of the robot is set to $x = 10 \times g_x$.

The fitness function for each robot used is formalized in Eq. 5.

$$F_i = \sum_{j=0}^{n} P(x_i, x_j) \tag{5}$$

The fitness value F_i for a given robot i is computed as the sum of its payoffs $P(x_i, x_j)$ obtained during evaluation, with x_i the robot's investment value (which remains constant through evaluation), and x_j the investment of its partner at interaction j^{th}.

Fitness proportionate selection, which can maintain diversity in collective evolutionary robotics setups, is used to build a new population. Mutation is applied to the genome of the selected individuals. Each gene g_k of a robot has a probability $\mu = 0.01$ to mutate. If the gene is selected for mutation, then it has a probability of 0.1 to mutate according to a uniform distribution $\mathcal{U}([-10, 10))$ and a probability of 0.9 to mutate according to a normal distribution $\mathcal{N}(g_i, \sigma)$ with $\sigma = \sigma_w = 0.1$ for the weight genes and $\sigma = \sigma_x = 0.1$ for the investment gene. The new generation then performs the task and the process is repeated for $G = 200$ generations (see Table 2 for a list of all the parameters).

Param	Description	Value
Payoff		
a	Public good weight	5
b	Prisoner's dilemma weight	3
Environment		
T	Number of iterations per generation	100 000
G	Number of generations per run	200
	Arena diameter	400px
	Robot size	4px
	Robot sensor range	96px
	Robot max speed	2px/iteration
ω	Number of ressources	30
	Ressource radius	3px
	Ressource footprint radius	10 px
τ	End of interaction probability	
Evolution hyper-parameters		
μ	mutation probability	0.01
σ_w	mutation strength of weight genes	0.1
σ_x	mutation strength of investment gene	0.1

Table 2: Experimental parameters

Results

Experimental setup

The environment is a circular arena with a diameter of 400px. The robots are 4px[1] diameter disks. The robots have 8 equally distributed sensors with a range of 96px giving them information about their surroundings, such as the presence of other robots, of a resource or of a wall. The robots move through the environment at a maximum translation speed of 2px/iteration and a rotational speed of 30°/iteration. N robots are spread randomly in the environment and 30 resources are randomly scattered throughout the arena. Each generation lasts $T = 100\,000$ iterations. The environment is represented in Figure 1.

The results presented below are obtained by observing of the final generation (i.e.: 200^{th} generation). We ran 24 simulations per condition in all experiments. The code used to generate all results is freely available[2].

In this Section, we explore whether cooperation can easily be learned, or not. Our hypothesis is that while partner choice should enable cooperative behaviors that is socially optimal, such cooperative behavior may be hindered by other factors such as the number of opportunities to meet other robots.

In the following, the influence of several factors are explored that may facilitate (or not) the emergence of partner choice and cooperation behaviors:

- the effect of population size (Section Learning Cooperation and Population Size). Hypothesis: a robot does not have time to search for a "good" partner if the population is small, as all pairing will be quickly made;
- the effect of the duration of interactions by changing the split probability τ (Section Learning Cooperation and Interaction Length). Hypothesis: exploration will be favored when interactions are long as there is a strong cost to cooperate with a "bad" partner.

Section Learning Cooperation and Population Size also presents the main control experiment, i.e. removing entirely partner choice, to demonstrate that partner choice is indeed mandatory to attain efficient cooperative foraging under the right experimental conditions.

In addition, three control experiments are described that explore the sensibility of results with respect to our particular experimental settings:

Section Effect of Mutation Strength (Control) explores the impact of both weaker and stronger mutation strengths applied on the investment gene σ_x. In particular, a weaker mutation may hinder the possibility to innovate towards better cooperators. We show that this is not the case: results are robust w.r.t. mutation.

Section Population Size vs Generations (Control) explores the influence of the parameters chosen for the evolutionary algorithm. Given a constant evaluation budget, a different balance between the number of generations (which is fixed to 200) and the population size may have an impact. For setups that use a small population, it is possible that better results may be obtained by using more generations. To evaluate this, we adjust the setup with the smallest population to an evaluation budget that matches that of the setup with the larger population, by augmenting the number of generations. We show that the evolutionary algorithm is robust w.r.t. the parameters used.

Section Wandering and Relocation (Control) focuses on the possible bias due to the particular implementation of the *ad hoc* wandering behavior. The wandering behavior acts as a diffusion process for the robots, but it is clear that diffusion is neither anisotropic nor provides uniform relocation due to the multiple collisions that can occur with obstacles in the arena. We show that using an unrealistic "teleportation" behavior instead, which ensures pure uniform relocation, actually does *not* change the results obtained before, thus confirming that our particular implementation of the wandering behavior does *not* bias the outcome.

Learning Cooperation and Population Size

To test how partner choice enables the emergence of cooperative behavior, we set $\tau = 0$ and the evaluation duration to $T = 100\,000$. This supposedly corresponds to a favorable setup as robots will benefit from a long search time and a very engaging commitment (i.e. only one pairing is possible) if they accept to interact. Figure 3 provides the results for different population sizes, from 50 to 1000 robots.

At $N = 50$, robots plays the defective strategy. With 50 robots in the arena, the robots are unable to meet and

[1] px is short for pixels, the basic unit length used in the simulator

[2] `http://pages.isir.upmc.fr/~bredeche/Experiments/ALIFE2020_coopPC_code.zip`

sample enough partners to be selective before the end of the generation. Moreover, the robots are racing to find a partner quickly. Indeed, with $\tau = 0$, the more the task advances in time, the fewer robots are available in the arena and thus the more β decreases throughout the evaluation.

On the other hand, robots evolve a cooperative behavior for N sufficiently large as larger population also implies a higher probability of encounters β. With a population of $N = 1\,000$, the average investment level is close to the social optimum.

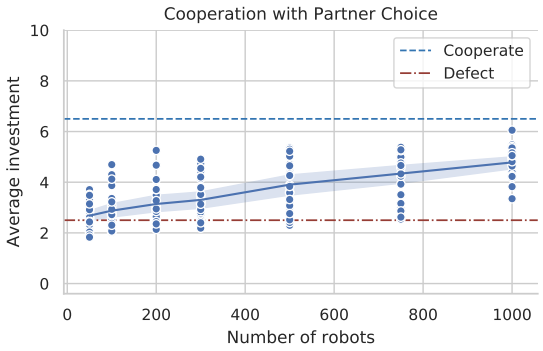

Figure 3: Evolution of cooperation with partner choice for a split probability $\tau = 0$ and mutation strength for the investment gene $\sigma_x = 0.1$. For each setup, 24 independent runs are performed (less is shown due to overlaps). Results are compiled from 168 runs obtained from 7 different experimental setups. For each setup, learning is performed for 200 generations with a given population size (x-axis). The values for population size are: 50, 100, 200, 300, 500, 750, 1000. In addition, the blue line shows the average values for each setup with a confidence interval $CI_{0.95}$.

To validate the importance of partner choice in the evolution of a cooperative behavior, we build a control experiments where we deactivate the robots' ability to know their partner's investment in order to accept or not accept an interaction. In this condition, whatever the number of robots in the environment, the average investment level always converge to x_d, that is a defective behavior (see Fig. 4). In this situation, robots have no way to be selective and cannot choose a cooperative robot over a non-cooperative one. Thus, cooperative robots are not preferentially selected as partners and there is no incentive to invest more than the individual optimum.

Learning Cooperation and Interaction Length

Figure 5 shows how the value of the split probability τ affects learning cooperation. When the split probability τ is null or low ($\tau < 10^{-3}$), the robots invest in a collectively optimal way and adopt a cooperative strategy. The robots plays systematically a defective strategy when $\tau \geq 10^{-3}$.

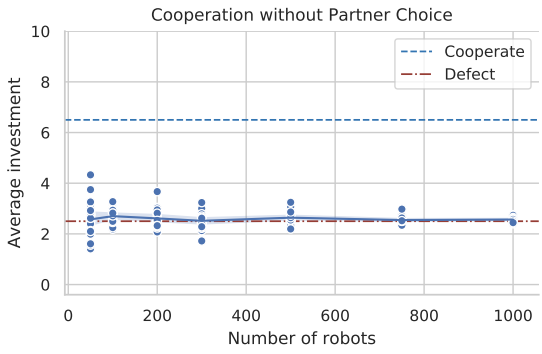

Figure 4: Evolution of cooperation *without partner choice* for a split probability $\tau = 0$ and mutation strength for the investment gene $\sigma_x = 0.1$. Technical details are identical to those of Fig. 3 (see caption).

Thus, decreasing the split probability (ie. increasing the interaction time) has a positive effect on the acquisition of a cooperative strategy by partner choice, in accordance with previous theoretical results.

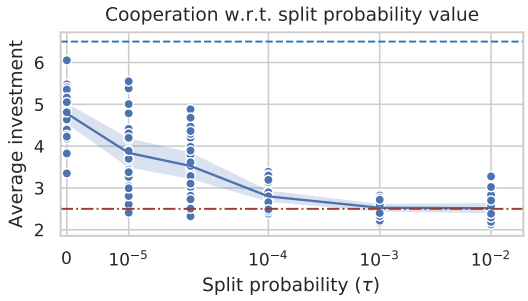

Figure 5: Evolution of cooperation with partner choice for a population size of 1000 robots and a mutation strength for the investment gene $\sigma_x = 0.1$. For each setup, 24 independent runs are performed. Results are compiled from 144 runs obtained from 6 different experimental setups. For each setup, learning is performed for 200 generations with a given split probability τ (x-axis). The values for τ are: 0, 10^{-5}, 5×10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}. In addition, the blue line shows the average values for each setup with its 95% confidence interval.

Effect of Mutation Strength (Control)

Previous results in theoretical biology have shown the importance of variability in the level of investment in the population (McNamara et al., 2008). In a population with increased phenotypic diversity, each individual can select part-

ners from a pool of individuals with different strategies. Cooperators may then be chosen over non-cooperators, just because they are available. As a consequence, this can bootstrap the emergence of cooperative strategies.

We test the influence of phenotypic variability (which, in our case, is directly linked to the investment value g_x), induced by smaller or larger mutation rates. To do so, we modify the strength σ_x of the Gaussian mutation on the gene encoding the robot investment level.

Figure 6 shows that differences are minor in the average investment level between the different simulations when using larger and smaller mutation strengths (to be compared with the original results in Fig.3). However, there is less variability between simulations when the mutation level is high ($\sigma_x \geq 0.1$), which can be explained by a more rapid convergence towards the optimal investment level.

The fact that the variability of investment in the environment plays very little role in our task may be due to the presence of individuals with various levels of investment in the initial population. The ability to be selective in the choice of partner may therefore emerge before the population is completely homogeneous and thus generating phenotypic variability becomes an unnecessary feature.

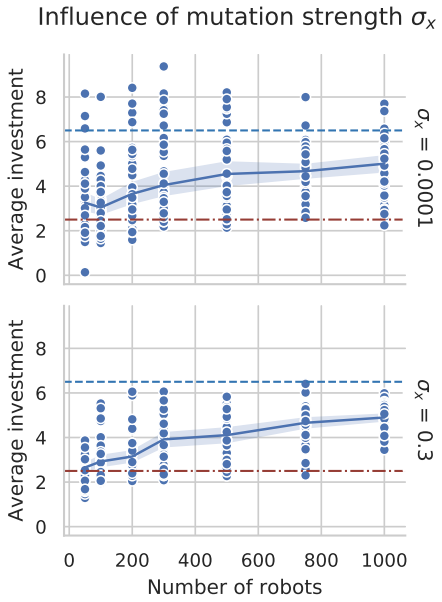

Figure 6: Evolution of cooperation with partner choice for a split probability $\tau = 0$ and mutation strength for the investment gene of $\sigma_x = 0.0001$ (top) and $\sigma_x = 0.3$ (bottom). Technical details are identical to those of Fig. 3 (see caption), which showed results for $\sigma_x = 0.1$.

Population Size vs Generations (Control)

The difference in population sizes between low (50 robots) and large (1000 robots) populations could be explained by the smaller evaluation budget used with small populations. Indeed, given the number of generations is constant ($G = 200$), the number of evaluations when considering a population of 50 robots is $50 \times 200 = 10000$, while a population with 1000 robots gets $1000 \times 200 = 200000$ evaluations. This difference in the number of evaluations could explain why cooperative behavior has evolved in the conditions where N is large and not in those where N is small.

We test the impact of the number of evaluations by running a new control condition of 24 simulations with $G = 4000$ for a population of $N = 50$ robots, offering 200000 evaluations. Fig. 7 shows the difference between this new setup ($N = 50, G = 4000$) and both the previous setup with a similar population size but fewer generations ($N = 50, G = 200$) and the previous setups with similar number of evaluations but a larger population ($N = 1000, G = 200$).

The difference between the ($N = 50, G = 200$) setup and the ($N = 50, G = 4000$) setup turns out to be marginal, while the difference with the ($N = 1\,000, G = 200$) setup using larger population remains largely significant. We conclude that adding more generations does not improve the level of cooperation achieved for conditions with a small population. These results confirm that smaller meeting probability β is responsible for blocking the emergence of cooperative behavior under these conditions.

Wandering and Relocation (Control)

In order to account for a possible bias due to the *ad hoc* wandering behavior used, an unrealistic "off-grid" behavior is introduced in place of the wandering behavior. The off-grid behavior is a mechanism that simply removes a robot from the environment after a successful interaction, and relocates it at a random position after some time depending on the split probability τ. It is actually closer to the abstract process used in numerical simulation in evolutionary biology models on partner choice, where space is ignored (Debove et al., 2015).

Figure 8 shows results obtained with the off-grid behavior for various values of split probability τ (instead of the wandering behavior, as used for results previously shown in Figure 5). Using either the off-grid behavior or the wandering behavior produces similar results. When the split probability is low ($\tau < 1 \times 10^{-5}$), robots tend to be more cooperative (investment value > 2.5). When it is large ($\tau > 10^{-3}$), robots' investment values converge to the 2.5, which means that defection is the rule with either behavior.

Using the off-grid behavior instead of the wandering behavior does actually provide an advantage for intermediate investment values, as robots remain cooperative for larger τ values. This can be explained by the fact that the arena

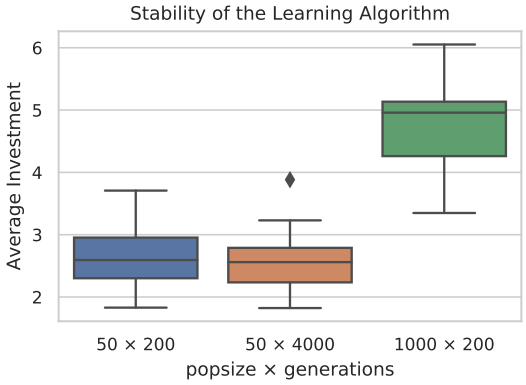

Figure 7: Performance is compared using different budget balance between population size and number of generations. From left to right: (1) population=50, generations=200; (2) population=50, generations=4000; (3) population=200, generations=1000. Results from (1) and (3) are taken from Fig. 3 and uses different number of evaluations, but the same number of generations (200). Results from (2) are obtained with an evaluation budget similar to (3), i.e. $50 \times 4000 = 1000 \times 200 = 200\,000$, but with a population similar to (1), i.e. 50 robots.

is less crowded than in the wander condition due to the removal of robots from the arena. Indeed, a robot necessarily crosses *only* potential partners, and is not blocked by robots wandering around that cannot be available for interaction. In other words, the β encounter probability is greater with the off-grid behavior than with the wander behavior (see Section Partner Choice).

Conclusion

We have shown that partner choice in evolutionary swarm robotics with heterogeneous population is a key mechanism to overcome deceptive social dilemma. We have also shown that efficient partner choice can be learned, but that its success strongly depends on environmental conditions. In particular, the number of encounters should be high, and the impact of interaction during or after cooperation should be long in duration.

Acknowledgements

This work was supported by the MSR project funded by the Agence Nationale pour la Recherche under Grant No ANR-18-CE33-0006.

References

Aktipis, C. A. (2011). Is cooperation viable in mobile organisms? Simple Walk Away rule favors the evolution of cooperation in groups. *Evolution and Human Behavior*, 32(4):263–276.

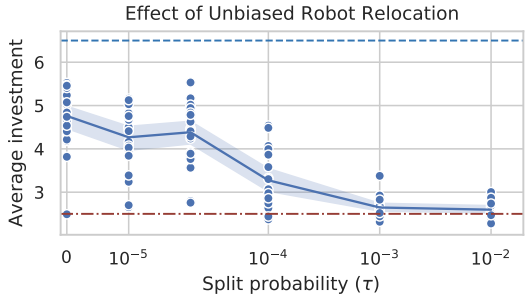

Figure 8: Evolution of cooperation *with off-grid behavior instead of the wandering behavior* with a population size of 1000 and mutation strength for the investment gene $\sigma_x = 0.1$. Technical details are identical to those of Fig. 5 (see caption).

Amato, C., Konidaris, G. D. G., Cruz, G., Maynor, C. A., How, J. P., and Kaelbling, L. P. (2015). Planning for decentralized control of multiple robots under uncertainty. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1241–1248. IEEE.

Axelrod, R. and Hamilton, W. (1981). The evolution of cooperation. *Science*, 211(4489):1390–1396.

Bayindir, L. (2016). A review of swarm robotics tasks. *Neurocomputing*, 172:292–321.

Beni, G. (2005). From Swarm Intelligence to Swarm Robotics. *Robotics*, 3342:1–9.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2012). Swarm robotics : A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41.

Bredeche, N., Haasdijk, E., and Prieto, A. (2018). Embodied Evolution in Collective Robotics: A Review. *Frontiers in Robotics and AI*, 5:12.

Bredeche, N., Montanier, J.-M., Weel, B., and Haasdijk, E. (2013). Roborobo! a Fast Robot Simulator for Swarm and Collective Robotics. pages 1–2.

Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). *Self-organization in biological systems*. Princeton University Press.

Debove, S., Andre, J.-B., and Baumard, N. (2015). Partner choice creates fairness in humans. *Proceedings of the Royal Society B: Biological Sciences*, 282:20150392.

Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. E. G. (2015). Evolutionary Robotics: What, Why, and Where to. *Frontiers in Robotics and AI*, 2(March):1–18.

Ferrante, E., Turgut, A. E., Duéñez-Guzmán, E., Dorigo, M., and Wenseleers, T. (2015). Evolution of Self-Organized Task Specialization in Robot Swarms. *PLoS Computational Biology*, 11(8):e1004273.

Fudenberg, D. (1998). *The theory of learning in games*, volume 36. MIT Press, Cambridge, MA.

Hamann, H. (2018). *Swarm Robotics: A Formal Approach*. Springer.

Hauert, S., Mitri, S., Keller, L., and Floreano, D. (2014). Evolving Cooperation : From Biology to Engineering. In *The Horizons of Evolutionary Robotics*, pages 203–217. MIT Press.

Heinerman, J., Drupsteen, D., and Eiben, A. E. (2015). Three-fold Adaptivity in Groups of Robots: The Effect of Social Learning. In Silva, S., editor, *Proceedings of the 17th annual conference on Genetic and evolutionary computation*, GECCO '15, pages 177–183. ACM.

Mataric, M. J. (1992). Integration of Representation Into Goal-Driven Behavior-Based Robots. *IEEE Transactions on robotics and automation*, 8(3).

McNamara, J. M., Barta, Z., Fromhage, L., and Houston, A. I. (2008). The coevolution of choosiness and cooperation. *Nature*, 451(7175):189–192.

Trianni, V. (2008). *Evolutionary swarm robotics: evolving self-organising behaviours in groups of autonomous robots*, volume 108. Springer.

Trianni, V., Ampatzis, C., Christensen, A. L., Tuci, E., Dorigo, M., and Nolfi, S. (2007). From Solitary to Collective Behaviours : Decision Making and Cooperation. pages 575–584.

Tuci, E. and Trianni, V. (2014). On the evolution of homogeneous two-robot teams: clonal versus aclonal approaches. *Neural Computing and Applications*, 25(5):1063–1076.

Waibel, M., Floreano, D., and Keller, L. (2011). A quantitative test of Hamilton's rule for the evolution of altruism. *PLoS biology*, 9(5):1–7.

Waibel, M., Keller, L., and Floreano, D. (2009). Genetic Team Composition and Level of Selection in the Evolution of Cooperation. *IEEE Transactions on Evolutionary Computation*, 13(3):648–660.

West, S. A., Griffin, A. S., and Gardner, A. (2007). Social semantics: Altruism, cooperation, mutualism, strong reciprocity and group selection. *Journal of Evolutionary Biology*, 20(2):415–432.