Novelty-driven Cooperative Coevolution

Jorge Gomes   jgomes@di.fc.ul.pt
Instituto de Telecomunicações, Lisbon, Portugal
BioMachines Lab, Lisbon, Portugal
BioISI, Faculdade de Ciências, Universidade de Lisboa, Portugal

Pedro Mariano   plmariano@fc.ul.pt
BioISI, Faculdade de Ciências, Universidade de Lisboa, Portugal

Anders Lyhne Christensen   anders.christensen@iscte.pt
Instituto de Telecomunicações, Lisbon, Portugal
BioMachines Lab, Lisbon, Portugal
Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal

Abstract

Cooperative coevolutionary algorithms (CCEAs) rely on multiple coevolving populations for the evolution of solutions composed of coadapted components. CCEAs allow, for instance, the evolution of cooperative multiagent systems composed of heterogeneous agents, where each agent is modelled as a component of the solution. Previous works have, however, shown that CCEAs are biased towards stability: the evolutionary process tends to converge prematurely to stable states instead of (near-)optimal solutions. In this study, we show how novelty search can be used to avoid the counterproductive attraction to stable states in coevolution. Novelty search is an evolutionary technique that drives evolution towards behavioural novelty and diversity, rather than exclusively pursuing a static objective. We evaluate three novelty-based approaches that rely on, respectively (i) the novelty of the team as a whole, (ii) the novelty of the agents' individual behaviour, and (iii) the combination of the two. We compare the proposed approaches with traditional fitness-driven cooperative coevolution, in three simulated multirobot tasks. Our results show that team-level novelty scoring is the most effective approach, significantly outperforming fitness-driven coevolution at multiple levels. Novelty-driven cooperative coevolution can substantially increase the potential of CCEAs, while maintaining a computational complexity that scales well with the number of populations.

Keywords

Cooperative coevolution, multiagent systems, neuroevolution, novelty search, convergence to stable states, behaviour exploration.

1   Introduction

Cooperative coevolutionary algorithms (CCEAs) are capable of evolving solutions that consist of coadapted, interacting components (Potter and De Jong, 2000). Such approaches are promising because they potentially allow for large problems to be decomposed into smaller and more tractable sub-problems. In a typical CCEA, each component of the solution is evolved in a separate population. Components are evaluated as part of a complete solution that consists of one component from each population. The individual components are thus scored based on the performance of the complete solution as a whole rather than their individual performance.

A common application of CCEAs, which we focus in this study, is the evolution of multiagent behaviours (Potter et al., 2001). The natural decomposition of the problem into sub-components makes multiagent systems a good fit for cooperative coevolution: each agent can be represented as a component of the solution, and the coevolutionary algorithm evolves a set of agent behaviours that solve the given task. In this way, coevolution allows for the synthesis of heterogeneous multiagent systems, where each individual agent can evolve a specialised behaviour (see for instance Potter et al., 2001; Yong and Miikkulainen, 2009; Nitschke et al., 2009).

Cooperative coevolutionary algorithms are, however, plagued by a number of issues, among which premature convergence to stable states stands out (Wiegand, 2004; Panait and Luke, 2005a; Popovici et al., 2012). In a CCEA, the individuals of a given population are scored to optimise team performance, in the context of the individuals (team members) found in the other populations. But as the other populations are also evolving, the fitness of each individual is subjective: it depends on the collaborators with which it is evaluated. CCEAs therefore tend to gravitate towards stable states (Panait, 2010), regardless of whether or not such states correspond to optimal solutions for a given problem.

Convergence to stable states in CCEAs is substantially different from convergence to local optima in non-coevolutionary algorithms (Panait et al., 2006b). Contrary to non-coevolutionary algorithms, it has been show that even under ideal conditions of populations with an infinite number of individuals, CCEAs are not necessarily attracted to the global optimum (Panait, 2010). The two pathologies correspond to different types of *deception*: in non-coevolutionary algorithms, evolution might be deceived by the fitness gradient generated by the fitness function (Whitley, 1991), while in CCEAs each population might be deceived by the fitness gradient resulting from the choice of collaborators from the other coevolving populations (Wiegand et al., 2001).

Lehman and Stanley (2008, 2011a) recently proposed an evolutionary approach aimed at avoiding deception in non-coevolutionary algorithms, called *novelty search*. In novelty search, candidate solutions are scored based on the novelty of their behaviour with respect to the behaviours of previously evaluated individuals, and not based on a traditional, static fitness objective. Given the dynamic nature of the objective, novelty search has the potential to avoid deception and premature convergence. The approach has attained considerable success in many different domains (Doncieux and Mouret, 2014), especially in evolutionary robotics, both in single-robot systems (Mouret and Doncieux, 2012) and in multirobot systems (Gomes et al., 2013).

In this paper, we show how novelty search can be applied to cooperative coevolution in the embodied multiagent domain. We propose three algorithms to accomplish novelty-driven coevolution: novelty search at the team behaviour level (*NS-Team*), at the individual agent behaviour level (*NS-Ind*), and a combination of the two (*NS-Mix*). In *NS-Team*, the novelty score assigned to an individual is based on the behaviour displayed by the team in which it is evaluated — regardless of the individual's specific contribution. In *NS-Ind*, the novelty score assigned to an individual is solely based on the behaviour of that individual agent when collaborating with a team. We conduct a detailed analysis of the proposed algorithms in the evolution of neural controllers for predator agents in a predator-prey pursuit task (Nitschke et al., 2012a; Yong and Miikkulainen, 2009), a benchmark task that requires a high degree of cooperation. The proposed algorithms are additionally evaluated in a simulated multi-rover task (Nitschke et al., 2009), and in a herding task (Potter et al., 2001).

We evaluate current fitness-based techniques for overcoming convergence to sub-

optimal equilibria, and show how they often become ineffective in more complex setups that involve more than two populations. We show how novelty-driven coevolution, due to its dynamic novelty objective, can overcome the problem of convergence to poor equilibrium states, and display a superior performance in almost all experimental setups. We additionally study the proposed novelty-driven algorithms according to behaviour space exploration and convergence, diversity of solutions, and scalability with respect to the number of populations.

This paper is a substantially revised version of (Gomes et al., 2014b), where we originally proposed novelty-driven cooperative coevolution. In this paper, we have refined and improved the previously proposed algorithms, and we perform a comprehensive evaluation at multiple levels. The key novel contributions of this paper include: (i) we introduce *NS-Mix*, a combination of team and individual novelty; (ii) we use a multi-objective algorithm to combine the novelty and fitness scores; (iii) we study the phenomenon of premature convergence using techniques found in previous works; (iv) we compare novelty-driven coevolution with previously proposed techniques for overcoming premature convergence to stable states; (v) we analyse the scalability of *NS-Team* regarding the number of populations; (vi) we show the diversity of team behaviours that *NS-Team* can evolve; (vii) we study a variation of *NS-Team* where only the novelty scores are used; and (viii) we evaluate the proposed approaches in two new tasks, the herding task and the multi-rover task, and with a more capable neuroevolution algorithm (NEAT).

2 Background

2.1 Cooperative Coevolution

In the domain of embodied multiagent systems, such as multirobot systems, several tasks have been solved with cooperative coevolutionary algorithms. Some examples include: (i) the *predator-prey* task (Nitschke et al., 2012a; Yong and Miikkulainen, 2009; Rawal et al., 2010; Gomes et al., 2014b), a common testbed in coevolution studies where a team of predators must cooperate to catch a fleeing prey; (ii) the *herding* task (Potter et al., 2001; Gomes et al., 2015b), in which a team of shepherds must cooperate to drive a sheep towards a corral, while at the same time keep a fox at distance; (iii) the *gathering and collective construction* task (Nitschke et al., 2012b), in which robots must carry different types of building blocks to a construction zone and place them in a specific sequence; (iv) the *multi-rover* task (Nitschke et al., 2009), which consists of a team of simulated vehicles (rovers), that must detect different types of features of interest spread throughout the environment; (v) *keepaway soccer* (Gomes et al., 2014b), a simplified version of robot soccer in which multiple keepers have to maintain possession of the ball against one or more takers; and (vi) an *item collection* task (Gomes et al., 2015a), where a ground robot cooperates with an aerial robot to collect items in the environment.

The classic cooperative coevolution architecture (Potter and De Jong, 2000) models a system comprising two or more species, with each species isolated in a separate population. This means that individuals only compete and reproduce with members of their own species. Due to complete separation of populations, it is possible to have different evolutionary algorithms and different individual representations for each population (Gomes et al., 2015a) (for example, neural networks could be evolved in one population while program trees could be evolved in another). At every generation of the evolutionary process, each population is evaluated in turn. To evaluate an individual from one population, teams are formed with one representative from each of the other populations. The resulting teams are then evaluated by a fitness function in the task domain,

and the individual under evaluation receives the fitness score obtained by the team as a whole. If the same individual is evaluated in multiple different teams, the scores are aggregated by taking the average or the maximum value (Wiegand et al., 2001).

2.2 Convergence to Equilibrium States

In a cooperative coevolutionary algorithm, the fitness landscape of each population is defined (and limited) by the behaviour of the team members. The fitness landscape is thus constantly changing, as the individuals from the other populations evolve. The fitness of an individual can vary significantly depending on with which collaborators it is evaluated. It is therefore easy for a population to be misled by a particular selection of collaborators from the other populations. CCEAs are naturally attracted to equilibrium states where each population is perfectly adapted to one other, such that changing one of the team members would result in a lower team performance. There is, however, no guarantee that such stable states correspond to globally optimal solutions (Panait, 2010). A related problem is *relative overgeneralisation*, which occurs when populations in the system are attracted to regions of the search space in which there are many strategies that perform well with the individuals from the other populations (Panait et al., 2004; Wiegand and Potter, 2006).

Due to these pathological dynamics, it has been shown that, in many cases, CCEAs are actually attracted to suboptimal regions of the search space (Panait, 2010; Jansen and Wiegand, 2004). Premature convergence to equilibrium states should be distinguished from the typical local convergence problems that plague non-coevolutionary algorithms (Panait et al., 2006b). While under ideal conditions a genetic algorithm is theoretically attracted to the global optimum, the same does not hold for cooperative coevolutionary algorithms. Panait (2010) has shown that even under ideal conditions of infinite populations, the CCEA might still not be attracted to the optimum. The lack of bias towards optimal solutions compromises the effectiveness of cooperative coevolutionary algorithms (Panait and Luke, 2005a).

A number of strategies have been proposed to overcome convergence to suboptimal equilibria in two-population CCEAs. Panait et al. (2004); Wiegand et al. (2001); Popovici and De Jong (2005) demonstrated that an optimistic reward scheme can be used to bias coevolution towards globally optimal solutions. The optimistic scheme evaluates an individual, not with only one collaboration, but in N trials, each with a randomly formed collaboration, and only the maximum reward obtained is considered. Panait (2010) showed that this optimistic scheme guarantees convergence to a global optimum if given enough resources, i.e., sufficiently large populations, and a sufficiently high number of collaborations N. This scheme can naturally have a drastic impact on the computational complexity of the algorithm. To reduce the number of necessary evaluations, Panait and Luke (2005b) proposed a variation of the optimistic reward scheme, wherein the number of collaborations N decreases with time.

Panait et al. (2006a) presented an archive-based algorithm called *iCCEA*, in which the number of evaluations is reduced by maintaining an archive of *informative collaborations* for each population. *iCCEA* builds an archive of collaborators which produce the same ranking of individuals in the other population as they would receive if they were tested against the full population of collaborators. The authors, however, acknowledge that the archive can become large, which makes evaluation computationally expensive, and that it is unclear if/how the algorithm would scale to complex domains.

The maximum of N collaborations scheme was extended by Panait et al. (2006b): the fitness is based partly on the maximum score obtained in N collaborations with ran-

domly chosen partners, and partly on the reward obtained when partnering with the *optimal* collaborator, i.e., the collaborator with which the individual under evaluation would receive the highest possible fitness score. The results showed that computing the fitness of an individual based on its performance with the optimal collaborator can significantly increase the performance of the algorithm. The assumption that the optimal collaborator is known is, however, largely unrealistic for most domains, and as such heuristic methods would be necessary for estimating the optimal collaborator.

Existing studies on premature convergence to suboptimal equilibria are, however, mostly focused on function optimisation (e.g. Wiegand et al., 2001; Panait et al., 2006b; Popovici and De Jong, 2005) and evolutionary game-theory (e.g. Panait, 2010; Wiegand et al., 2002; Wiegand and Potter, 2006), and always with only two coevolving populations. It is thus unclear whether these methods for overcoming convergence to suboptimal equilibria are efficient and effective in the embodied multiagent domain. In this domain, systems are often composed of more than two agents, and existing methods rely on the use of large numbers of collaborations to assess the fitness of each individual. When evolving controllers for embodied agents, the number of generations can only be reduced to some extent — a large number of generations is typically needed for fine-tuning the controllers (which can have hundreds of parameters in the case of neural networks), even with a perfect fitness gradient. An increase in the number of collaborators therefore results in a steep increase of computational complexity that is not viable in domains that rely on time-consuming simulations for evaluating the individuals. In previous works on embodied multiagent systems, collaborations are typically formed with only the best individual of each population (e.g. Potter et al., 2001; Yong and Miikkulainen, 2009; Nitschke et al., 2012b).

To the best of our knowledge, the issue of premature convergence to suboptimal equilibria has not been directly studied in the domain of embodied multiagent systems. The principles of the pathology, however, apply to any cooperative coevolution application (Panait, 2010). In a number of studies that apply CCEAs to embodied multiagent systems, problem decomposition techniques (Panait and Luke, 2005a), incremental evolution (Gomez and Miikkulainen, 1997) in particular, are used to achieve successful solutions in reasonable time (Nitschke et al., 2009, 2012a,b; Yong and Miikkulainen, 2009). The described necessity of using problem decomposition can potentially be ascribed to the tendency of traditional CCEAs to converge to suboptimal equilibria.

2.3 Novelty Search

Novelty search (Lehman and Stanley, 2011a) is a recently proposed evolutionary approach to overcome the problem of deception. Novelty search drives evolution towards behavioural novelty instead of a pre-defined goal. The distinctive aspect of novelty search is how the individuals of the population are scored. Instead of being scored according to how well they perform a given task, which is typically measured by a static fitness function, the individuals are scored based on their behavioural novelty according to a dynamic *novelty metric*, which quantifies how different an individual is from other, previously evaluated individuals. This reward scheme therefore creates a constant evolutionary pressure towards behavioural innovation, and actively tries to avoid convergence to a single region in the solution space.

Novelty search has mainly been studied in the evolutionary robotics domain, including the evolution of: (i) single-robot controllers (e.g. Mouret and Doncieux, 2012; Lehman and Stanley, 2011a), (ii) controllers for homogeneous multirobot systems (Gomes et al., 2013), (iii) two-population competitive coevolution (Gomes et al., 2014a),

(iv) morphologies (Lehman and Stanley, 2011b), and (v) plastic neural networks (Risi et al., 2010). A few applications of novelty search outside the robotics domain can also be found in the literature, for instance in machine learning (Naredo et al., 2013; Naredo and Trujillo, 2013) and game content generation (Liapis et al., 2015). The previous works have shown that novelty search is able to find good solutions faster and more consistently than fitness-based evolution in many different applications. Novelty search is particularly effective when dealing with deceptive domains (Lehman et al., 2013). It has also been shown that novelty search can evolve a diverse set of solutions in a single evolutionary run, as opposed to fitness-based evolution that typically converges to a single region in the solution space (Gomes et al., 2013).

Implementing novelty search requires little change to any evolutionary algorithm aside from replacing the fitness function with a domain-dependent novelty metric (Lehman and Stanley, 2011a; Gomes et al., 2015c). To measure how far an individual is from other individuals in behaviour space, the novelty metric relies on the average behaviour distance of that individual to the k-nearest neighbours:

$$\rho(x) = \frac{1}{k} \sum_{i=1}^{k} dist(x, \mu_i) \ ,$$

(1)

where μ_i is the ith-nearest neighbour of x with respect to the distance metric $dist$. Potential neighbours include the other individuals of the current population and a sample of individuals from previous generations, stored in an archive. The archive can be composed stochastically, with each individual having a fixed probability of being added, or by adding the behaviours that are sufficiently different from the ones already there (Gomes et al., 2015c). The function $dist$ is a measure of behavioural difference between two individuals, which will be further discussed in the next section.

2.3.1 Behaviour Distance Measures

The behaviour of each individual is typically characterised by a real-valued vector. The behaviour distance $dist$ is then the distance between the corresponding characterisation vectors. The design of a behaviour characterisation has direct implications on the effectiveness of novelty search. An excessively detailed characterisation can open the search space too much, and might cause evolution to focus on regions of the behaviour space that are irrelevant for solving the task (Cuccu and Gomez, 2011). On the other hand, an incomplete or inadequate characterisation can cause counterproductive conflation of different behaviours (Kistemaker and Whiteson, 2011).

Task-specific distance measures

Most previous works on behavioural diversity rely on behaviour characterisations designed specifically for the given task. These characterisations are composed of behavioural traits that the experimenter considers relevant for describing agent behaviour in the context of the given task. In the evolutionary robotics domain, the characterisations typically have a strong focus on the spatial relationships between entities in the task environment, or the location of the robots in the environment (Gomes et al., 2014c). They typically comprise only a small number of different behavioural traits (up to four), and are based either on the final state of the environment, or a single quantity sampled or averaged over the evaluation trial.

Generic distance measures

Doncieux and Mouret (2010) proposed generic behaviour similarity measures to overcome the necessity of manually designing behaviour characterisations in single-robot

tasks. The proposed measures are exclusively based on the sensor-effector states of the agent. They rely on comparisons between the sequences of all binary sensor and effector values of the agent through time, or counting how many times the agent was in each possible sensor-effector state. These generic similarity measures were extended by Gomes and Christensen (2013), making them applicable to multiagent systems, and to non-binary sensors and effectors. While generic measures are widely applicable, they can result in a very large behaviour space (Cuccu and Gomez, 2011; Mouret, 2011). To address this issue, Gomes et al. (2014c) proposed a middle ground between generic and task-specific characterisations: *systematically derived behaviour characterisations* (SDBCs), which are based on behaviour features systematically extracted from a formal description of the agents and their environment. Such features include, for instance, average distances between the agents, agents' average speed, energy levels, and so on.

Mouret and Doncieux (2012) compared task-specific and generic characterisations in a comprehensive empirical study with a number of single-robot tasks. Doncieux and Mouret (2013) showed how different similarity measures (generic or task-specific) can be combined, either by switching between them throughout evolution or by calculating the behaviour distance based on all similarity measures.

2.3.2   Balancing Novelty and the Objective

Previous works have found that combining the exploratory character of novelty search with the exploitative character of fitness-based evolution often is an effective way to apply novelty search (Lehman et al., 2013; Gomes et al., 2015c). A number of techniques have been proposed to accomplish this combination. One possible way is to establish a task-dependent minimal criterion that the individuals must meet in order to be considered viable for selection. This minimal criterion can either focus on a certain aspect of the individual's behaviour and be provided by the experimenter (*MCNS*, Lehman and Stanley, 2010), or be dynamic and calculated based on the fitness scores of the current population (*PMCNS*, Gomes et al., 2014d).

Cuccu and Gomez (2011) proposed to score each individual based on a *linear scalarisation* of its novelty and fitness scores, with a parameter ρ controlling the relative weight of fitness and novelty. Mouret (2011) proposed novelty-based multi-objectivisation, which we use in this study, where a novelty objective is added to the task objective (fitness function) in a multi-objective evolutionary algorithm. Gomes et al. (2015c) showed that the multi-objectivisation was one of the most effective approaches for combining novelty and fitness, and it has the advantage of not relying on additional parameters.

3   Novelty-driven Cooperative Coevolution

We propose three distinct approaches based on novelty search to overcome convergence to stable states in multi-population cooperative coevolution. The first approach, *NS-Team*, is based on traditional cooperative coevolution principles: an individual's novelty score is calculated based on the behaviour of the team in which it participated, without any discrimination of the individual agent behaviours. The second approach, *NS-Ind*, is based on the typical implementation of novelty search in non-coevolutionary algorithms: individuals are rewarded for exhibiting novel individual behaviours with respect to the other individuals in their population, thus maintaining behavioural diversity inside each population. The third approach, *NS-Mix*, is a combination of the first two: individuals are rewarded for displaying both novel individual behaviours and causing novel team behaviours.

Algorithm 1 *NS-Team*: Novelty-driven cooperative coevolution based on team-level behaviour characterisations.

1: Let P be the set of all populations in the coevolutionary system.
2: Let A be an archive of behaviour characterisations, initially empty.
3: **for each** generation **do**
4:     **for each** population $P_n \in P$ **do** choose one representative individual r_n.
5:     **for each** population $P_n \in P$ **do**
6:         **for each** individual $i \in P_n$ **do**
7:             Form one team t with individual i and the representatives r_m from each other population $P_{m \neq n}$.
8:             Evaluate team t, obtaining a fitness score f_t and a team behaviour characterisation b_t.
9:             $f_i \leftarrow f_t$, and $b_i \leftarrow b_t$.
10:         **for each** individual $i \in P_n$ **do** compute the novelty score ns_i, comparing b_i to the other individuals in P_n and behaviours in A.
11:         Apply to P_n the selection, mutation, and crossover operators, based on the scores f and ns.
12:     Update the archive A, using the characterisations b_i of individuals i in P.

3.1 Team-level Novelty

Team-level novelty (*NS-Team*) is described in Algorithm 1. In *NS-Team*, as in a typical cooperative coevolutionary algorithm, the evaluation of each individual begins with the formation of one team (joint solution) composed of that individual and representative individuals from each of the other populations (step 7). The chosen representative of each population is the individual that obtained the best team fitness score in the previous generation (step 4), or a random one in the first generation. The collective performance of the team is then assessed by evaluating it in the problem domain (step 8). *NS-Team* relies on the characterisation of the behaviour of a team as a whole. The novelty score of each individual is computed based on the team-level characterisation of the team with which it was evaluated (step 10). The novelty of the individual thus corresponds to the novelty of the team as a whole. This process is analogous to typical CCEAs, in which an individual receives the fitness of the team in which it participated, without discriminating the individual's contribution.

Besides the team's novelty score (step 10), the team's fitness score is also taken into consideration in the selection process (step 11). The motivation for such combination is to drive evolution towards the exploration of *valuable* behaviour regions (see Section 2.3.2). The key difference is that while the team fitness measure is static, the team novelty measure is dynamic. Contrary to what happens in fitness-driven CCEAs, in *NS-Team* the attractors keep changing throughout evolution: what is novel in one generation will only remain so for a few generations. The evolutionary process is constantly led towards novel regions of the team behaviour space, which can avoid premature convergence to a single region of the solution space.

It should be noted that the method used to compute of the novelty scores (step 10), the technique used to combine novelty and fitness (step 11), and the implementation of the archive update (step 12) are independent of *NS-Team*. We followed implementations for these steps commonly found in novelty search applications (Gomes et al., 2015c), described in Section 2.3. In our experiments, the combination of novelty and

team fitness objectives is achieved with the NSGA-II multiobjective algorithm (Deb, 2001). The proposed algorithm only concerns the evaluation phase of the evolutionary algorithm, and therefore any underlying evolutionary algorithm can theoretically be used (step 11).

Team behaviour can be characterised using the design principles proposed by Gomes et al. (2013): the behaviour characterisation focuses on the team as a whole, without directly discriminating between the respective contributions of individual agents. Previous studies have shown that such team-level characterisations can be crafted with task-specific knowledge (Gomes et al., 2013) or without it (Gomes and Christensen, 2013), see Section 2.3.1. Regarding task-specific characterisations, which are used in the experiments presented in this paper, we have shown (Gomes et al., 2014c) that team-level characterisations can be based on measures of how the team influences the task environment, or by averaging agent's behavioural traits over all the members of the team.

3.2 Individual-level Novelty

In domains where a high degree of cooperation is required for a joint solution to be successful, it may not be possible to assess the contribution of each agent to the success of the team. This issue is commonly known as the *credit assignment problem* (Potter and De Jong, 2000). Nonetheless, it is possible to describe the behaviour of each individual agent when participating in a team, ignoring to some extent whether the agent's actions are harmful or beneficial with respect to the team's objectives.

We study a novelty-based coevolutionary algorithm that uses individual agent behaviour characterisations, instead of the team-level behaviour characterisations used in *NS-Team*. In *NS-Ind*, individuals are rewarded for displaying novel agent behaviours, regardless of the behaviour of the teams in which the individuals were evaluated. The objective of *NS-Ind* is to directly promote behavioural diversity inside each population, thus preventing premature convergence of the evolutionary process, following the previous successes of novelty-based techniques in single-population evolutionary algorithms (Lehman and Stanley, 2011a; Gomes et al., 2013; Mouret and Doncieux, 2012).

The implementation of *NS-Ind* is similar to the novelty search implementation in non-coevolutionary algorithms, with one novelty archive for each population, and the novelty scores are computed inside each population. *NS-Ind* is detailed in Algorithm 2. During the evaluation of an individual, the behaviour of that individual in the context of a team is characterised (step 8). This characterisation is then used to compute the novelty of the individual, by comparing it with the other behaviours observed in the respective population, and in the archive of that population (step 10). The novelty of the individual is then combined with the fitness of the team in which it participated (step 12), in order to drive the coevolutionary system towards novel, high-quality solutions. As in *NS-Team*, in our experiments we use a multiobjective algorithm to combine the individual novelty and team fitness scores.

3.3 Mixed Novelty

We additionally propose and evaluate *NS-Mix*, which combines *NS-Team* and *NS-Ind*. In *NS-Mix*, the individuals are rewarded both for causing novel team behaviours *and* novel agent behaviours. The implementation relies on *NS-Team* and *NS-Ind*. The team-level novelty scores (ns_{team}) are calculated according to Algorithm 1, while the individual-level novelty scores (ns_{ind}) are calculated according to Algorithm 2. These two sets of novelty scores, together with the team fitness scores, are used to select and

---

Algorithm 2 *NS-Ind*: Novelty-driven cooperative coevolution based on agent-level behaviour characterisations.

1: Let P be the set of all populations in the coevolutionary system.
2: Let $A_1 \cdots A_{|P|}$ be a list of archives of behaviour characterisations, all initially empty.
3: **for each** generation **do**
4:     **for each** population $P_n \in P$ **do** choose one representative individual r_n.
5:     **for each** population $P_n \in P$ **do**
6:         **for each** individual $i \in P_n$ **do**
7:             Form one team t with individual i and the representatives r_m from each other population $P_{m \neq n}$.
8:             Evaluate team t, obtaining a fitness score f_t and one behaviour characterisation b_i describing the individual behaviour of i.
9:             $f_i \leftarrow f_t$.
10:         **for each** individual $i \in P_n$ **do** compute the novelty score ns_i, comparing b_i to the other individuals in P_n and behaviours in A_n.
11:         Update A_n, using the characterisations b_i of individuals $i \in P_n$.
12:         Apply to P_n the selection, mutation, and crossover operators, based on the scores f and ns.

---

breed the individuals of each population. In the experiments described in this paper, we implement *NS-Mix* with a multiobjective algorithm, maximising the three scores: team novelty, individual novelty, and team fitness score.

4 Behaviour Exploration Analysis

The analysis of the individuals in coevolutionary algorithms can provide a valuable insight into potential coevolutionary pathologies. Previous works have focused on the analysis of the best individuals evolved in each population, at every generation (*best-of-generation* individuals) (Popovici and De Jong, 2006). By plotting the trajectory of such individuals over the evolutionary run, it is possible to visualise to which regions of the solution space the coevolutionary process is converging. Since we study problems with more than two populations, and since individuals have multidimensional genomes/behaviours, the previously proposed methods cannot be directly applied. Instead, we rely on the behaviour of the team that obtained the highest fitness score in a given generation. The multidimensional behaviour space is reduced to two dimensions with Sammon mapping (Sammon, 1969) in order to produce a graphical representation. Besides the graphical representation, we define one metric for quantifying the behavioural dispersion of the best-of-generation teams (*BoG team dispersion*).

Previous works in novelty search have also shown that analysing the behaviour space exploration, based on all the evolved individuals, can be an important tool to uncover the evolutionary dynamics. For instance, Lehman and Stanley (2011b); Gomes et al. (2013) analyse the exploration of the behaviour space to discover the diversity of solutions for a given task. In a novelty-based evolutionary process, looking only at the best-of-generation individuals can be misleading, as a great deal of the exploration of the behaviour space can correspond to solutions with lower fitness scores. To this end, we implemented two measures of exploration, to cover both the team behaviour space (*All team dispersion*) and the individual behaviour space (*Individual dispersion*).

For all three metrics, dispersion is given by the mean difference between behaviour

characterisations. Considering a set of teams/individuals φ, the mean difference (MD) is given by:

$$MD(\varphi) = \sum_{i \in \varphi} \sum_{j \in \varphi} \frac{dist(i,j)}{|\varphi|(|\varphi| - 1)} \qquad (2)$$

where $dist$ is the Euclidean distance between the respective behaviour characterisation vectors. The mean difference is a non-parametric measure of statistical dispersion, that is not defined in terms of a measure of central tendency. A low mean difference value means that most teams/individuals have very similar behaviour characterisations, while a high value means the teams/individuals are well dispersed in the behaviour space. We defined the following metrics based on the MD:

BoG team dispersion: The behavioural dispersion of the best-of-generation teams evolved during a given evolutionary run (φ_b). The *BoG team dispersion* is given by $MD(\varphi_b)$, where the distance between any two teams, $dist$, is given by the Euclidean distance between the respective team behaviour characterisations. This metric is intrinsically related to convergence – a low value means that the highest scoring teams always displayed very similar team behaviours, which suggests that evolution converged to a specific region of the team behaviour space.

All team dispersion: Similar to BoG teams dispersion, but considering all teams evaluated during a given evolutionary run (φ_a). *All team dispersion* is thus given by $MD(\varphi_a)$, with $dist$ calculated based on the team behaviour characterisations.

Individual dispersion: The mean dispersion of the individual (agent) behaviours evolved in each population, averaged over all the populations. The distance $dist$ between any two individuals is given by the Euclidean distance between the respective *individual* behaviour characterisations. Considering P as the set of populations in the coevolutionary system, and φ_p the set of individuals evolved in population p, *individual dispersion* is given by:

$$D_i(\varphi) = \sum_{p \in P} \frac{MD(\varphi_p)}{|P|} \qquad (3)$$

5 Predator-prey Experiments

Predator-prey pursuit is one of the most common tasks studied in multiagent coevolution, both in cooperative coevolution (Nitschke et al., 2012a; Yong and Miikkulainen, 2009; Rawal et al., 2010) and in competitive coevolution (Nolfi, 2012; Rawal et al., 2010). Pursuit tasks involve a number of agents (predators) chasing a prey. The predators cannot move faster than the prey, and they therefore need to cooperate in order to successfully capture the prey. In cooperative coevolution studies (Nitschke et al., 2012a; Yong and Miikkulainen, 2009), only the team of predators is evolved, while the prey has a pre-specified fixed behaviour. The predator-prey task is especially interesting in cooperative coevolution studies because heterogeneity in the predator team is required to effectively catch the prey, along with a tight coordination among the predators.

5.1 Task Setup

The predators are initially placed in linear formation at one end of the arena, in the slots depicted in Figure 1 (a). We defined task variants with different number of predators, from two to seven. A single prey is randomly placed near the centre of the arena.
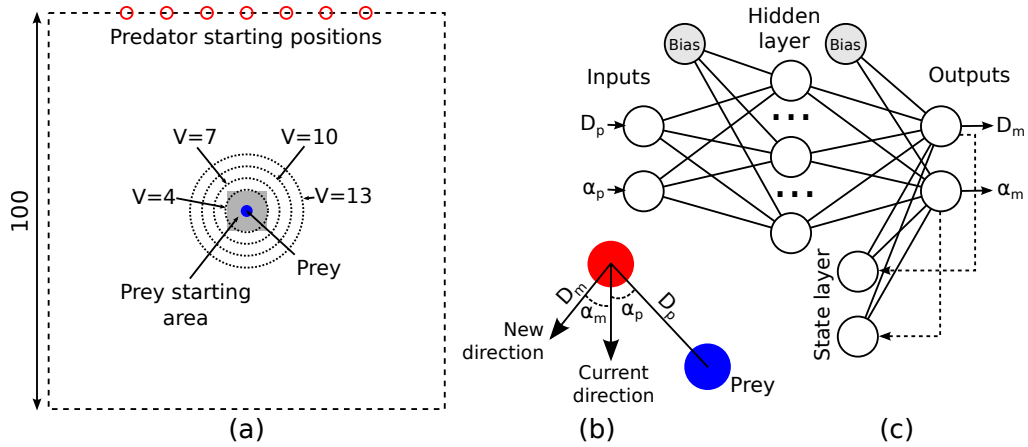
Figure 1: Predator-prey task setup. (a) Initial conditions of the simulation, with the possible prey vision ranges (V) and the possible predators' starting positions (circles at the top). (b) Sensors and effectors of each predator: the predator senses the distance D_p and relative orientation α_p of the prey, and the effectors control the speed D_m and turning angle α_m. (c) The structure of the neural network controller of each predator.

The arena is not bounded by walls, and if the prey escapes the arena, the trial ends. The task parameters are listed in Appendix A. We use a version of the task where the predators cannot communicate nor sense one another (Yong and Miikkulainen, 2009; Rawal et al., 2010). Each predator is controlled by a neural network that receives only two inputs: (i) the distance to the prey (D_p), and (ii) the relative orientation of the agent with respect to the prey (α_p). These inputs are normalised before being fed to the neural network, and the network's two outputs control the speed (D_m) and the rotation (α_m) of the agent, see Figure 1 (b). The neural network that controls each predator is a Jordan network (Jordan, 1997), a simple recurrent network with a state layer connected to the output neurons, see Figure 1 (c). The network has two inputs, eight hidden neurons, two outputs, and is fully connected[1].

The predators move at most at the same speed of the prey (1 unit/step). The behaviour of the prey consists of moving away from any predator within a radius of V around the prey. If there are no predators within the radius of V, the prey does not move. Otherwise, the prey moves at a constant speed, with a direction opposite to the centre of mass of the nearby predators. We use task variants with different values for the radius V, ranging from four to 13 units. The prey is captured if a predator collides with it. A trial ends if the prey is captured, escapes the arena, or if T=300 simulation steps elapse. Each team of predators is evaluated in five simulation runs, varying the starting position of the prey.

The fitness function F_{pp} is based on previous works (Nitschke et al., 2012a; Yong and Miikkulainen, 2009). If the prey was captured, F_{pp} increases as the time to capture the prey (τ) decreases, otherwise it increases as the average final distance from the

---

[1]The structure of the neural network and the number of hidden neurons were tuned empirically. Feed-forward networks and Elman networks (Elman, 1990) were also tested, and the number of hidden neurons was varied from three to ten.

predators to the prey (d_f) decreases:

$$F_{pp} = \begin{cases} 2 - \tau/T & \text{if prey captured} \\ max(0, (d_i - d_f)/size) & \text{otherwise} \end{cases}, \qquad (4)$$

where T is the maximum simulation length, d_i is the average initial distance from the predators to the prey, and $size$ is the side length of the arena.

The behaviour characterisations were defined based on systematically derived behaviour characterisations (SDBC) (Gomes et al., 2014c). We chose a subset of the extracted features, based on the estimated relevance of the features with the predator-prey task. The team-level behaviour characterisation $\beta_{pp}(t)$ is a vector of length 4. The agent-level characterisation of an agent ($\beta_{pp}(a)$) is based on the same behaviour features of $\beta_{pp}(t)$, but measured for a specific agent instead of the whole team. The characterisations are described in Table 1.

Table 1: Behaviour characterisations used in the predator-prey domain. All features have values normalised to the range [0,1].

Team-level characterisation $\beta_{pp}(t)$	Individual-level characterisation $\beta_{pp}(a)$
– Whether the prey was captured or not	– Whether agent a captured the prey
– Average final distance of the predators to the prey	– Final distance of a to the prey
– Average distance of each predator to the other predators over the simulation	– Average distance of a to the other predators over the simulation
– Simulation length	

5.2 Evolutionary Setup

We use a canonical genetic algorithm to evolve the neural networks that control the agents. The weights of the networks are directly encoded in the chromosomes. The algorithm uses tournament selection, the genes (weights) are mutated individually with a fixed probability, and we apply one-point crossover. The elite of each population passes directly on to the next generation. The parameters of the algorithm were tuned in preliminary experiments using the predator-prey task with three agents and $V = 7$. See Appendix A for parameter values.

Novelty search is implemented as described in Section 2.3, and configured according to the results presented by Gomes et al. (2015c). Individuals are randomly added to the archive, and the archive size is bounded for computational and memory efficiency: after reaching the size limit, random individuals are removed to allow space for new ones. The novelty objectives are combined with the team fitness objective with the multi-objective optimisation algorithm NSGA-II, as proposed by Mouret (2011).

Each experimental treatment was repeated in 30 evolutionary runs. In all experiments, the highest scoring team of each generation was re-evaluated *a posteriori* in 50 simulation trials. As the initial position of the prey is stochastic, the re-evaluation yields a more accurate estimate of the team fitness. All the team fitness plots presented in the paper are based on the scores obtained in this post-evaluation.

5.3 Base Fitness-driven Cooperative Coevolution

In the first set of experiments, we analyse how fitness-based coevolution performs when faced with varying degrees of task difficulty. We vary the difficulty of the task by varying the prey's visual range (V). Increasing V allows the prey more room and time to escape from the predators. As such, a higher degree of cooperation, as well as a more fine-tuned strategy, are required in the team of predators in order to successfully catch the prey. In setups with high V values (*V10, V13*), only one non-cooperating agent might be sufficient to compromise the performance of the whole team, as it can drive the prey away or leave room for it to escape. Figure 2 shows performance achieved with fitness-driven coevolution for three predators and with values of V varying from 4 to 13.
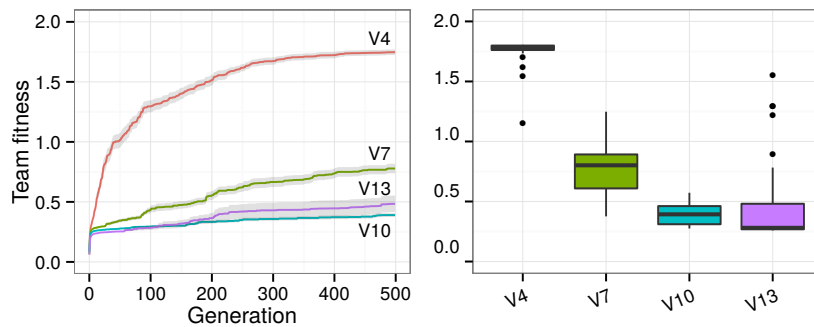
Figure 2: Team fitness scores achieved with fitness-based evolution in task setups with varying prey vision (V). Left: highest fitness scores achieved so far at each generation, averaged over 30 runs for each setup. The grey areas depict the standard error. Right: boxplots of the highest scores achieved in each evolutionary run, for each task setup. The whiskers represent the highest and lowest value within 1.5 IQR, and the dots indicate outliers.

The results show that fitness-driven coevolution (*Fit*) is only able to consistently evolve effective solutions in the easiest setup ($V4$). In the other setups, *Fit* rarely reaches high-quality solutions. It should be noted that it is possible to find effective solutions for all these task setups, as it will be shown in the following sections. To determine the reason for failure, we analyse the best-of-generation (BoG) teams, as described in Section 4. Figure 3 shows the behaviour of the BoG teams in representative evolutionary runs[2], along with the mean value of the BoG team dispersion (D) for each task setup. These results show that in the easier task setup (*V4*), coevolution can consistently explore the behaviour space and reach regions of the behaviour space where high-quality solutions can be found. In the other task setups, however, coevolution converges prematurely to a narrow region of the behaviour space, resulting in a relatively low degree of BoG team dispersion (D).

5.4 Increasing the Number of Collaborations

We experimented with techniques studied in previous works to try to overcome convergence to suboptimal solutions in fitness-driven coevolution. As suggested by Wiegand et al. (2001), we increased the number of random collaborations with which each

---

[2]For each setup, we chose the evolutionary run that had a value of *BoG dispersion* (D) closest to the mean of all runs in that setup.
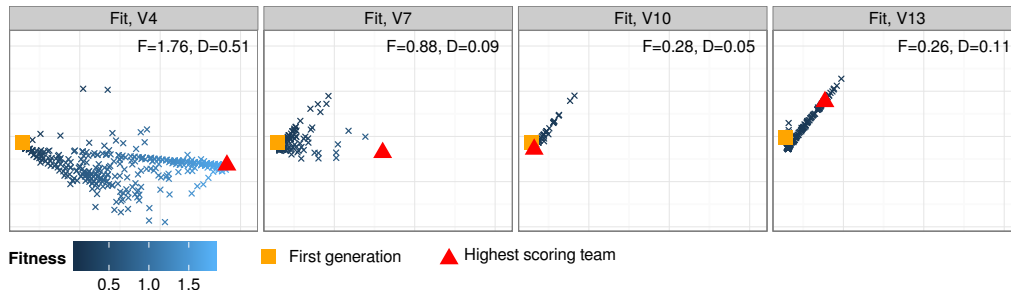
Figure 3: Behaviour of the best-of-generation teams in representative evolutionary runs of fitness-driven coevolution. Each cross represents one team, mapped according to its team behaviour. The four-dimensional behaviour space was reduced to two dimensions using Sammon mapping, see Section 4. D is the BoG team dispersion of the respective run, and F is the highest fitness score achieved.

individual is evaluated. The fitness assigned to an individual is the maximum reward it obtained with any collaboration. To evaluate each individual, $N + 1$ collaborations are formed: one with the best individuals from the previous generation, and N collaborations composed with randomly chosen collaborators. According to previous results (Panait, 2010; Wiegand et al., 2001; Popovici and De Jong, 2005), increasing the number N of collaborations should increase the likelihood of the coevolutionary algorithm to converge to the global optimum. We therefore evaluated how varying the number N impacts the performance of fitness-based coevolution. Since this scheme has, to the best of our knowledge, only been used in coevolutionary setups with two populations, we also experimented with a predator-prey setup with only two predators (*V4/2*) to establish a fair basis for comparison. In the remaining setups, three predators are used, and the random collaborations are formed by partnering the individual that is being evaluated with one randomly chosen individual from each of the other populations.

Figure 4 (left) shows the effect of increasing N in the different task setups. It should be noted that the number of generations was the same (500) in all evolutionary configurations, meaning that the number of evaluations increases linearly with N. The results show that using random collaborations can significantly improve the performance of fitness-based coevolution in the two-population setup (*V4/2*, $p = 0.005$, Kruskal-Wallis test), with respect to the highest team fitness scores achieved. This result is coherent with previous studies performed with two-population setups. The results obtained in the three-agent setups, however, reveal a substantially different trend. In *V7*, no significant differences in the performance were found ($p = 0.389$, Kruskal-Wallis), and in *V4*, *V10* and *V13*, increasing the number of collaborations can actually result in a lower performance ($p = 0.019$, $p < 0.001$, and $p = 0.006$ respectively).

Figure 4 (right) shows the influence of N on behavioural convergence (as defined in Section 4). In the setups *V4*, *V10*, and *V13*, increasing the number of collaborations led to an increase in convergence to specific region of the solution space ($p < 0.001$, Kruskal-Wallis test), which in turn correlates with inferior performance. In the other setups, the influence of N is less clear.

Our results suggest that the traditional methods of overcoming convergence to stable states may not be effective in coevolutionary setups with more than two populations, and with a large number of individuals. Panait (2010) demonstrated that a CCEA converges to the global optimum if a *sufficient* number of collaborations are used to
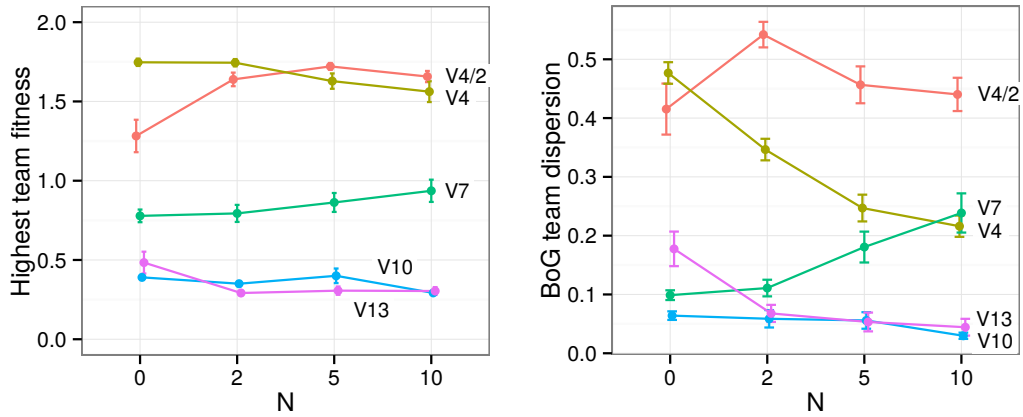
Figure 4: Left: Highest team fitness scores achieved in each evolutionary run, for each task setup with varying task difficulty (prey's vision range – V), and a varying number of random collaborators (N). The $V4/2$ setup uses only two predators, while the other setups use three predators. Right: Behavioural dispersion of the best-of-generation (BoG) teams. Standard error bars are shown.

evaluate each individual. An insufficient number of random collaborations might lead to poor fitness estimates that can result in convergence to suboptimal solutions. A *sufficient number of random collaborations* is, however, highly problem-dependent (Panait, 2010). As the number of possible collaborations increases exponentially with the number of populations and the number of individuals, the collaborations required to obtain a proper estimate of an individual's fitness may also increase significantly, maybe even exponentially. Unfortunately, our results do not provide a definite answer to this question, as exponentially increasing the number of collaborations is typically not computationally feasible in the domain of embodied multiagent systems.

5.5 Novelty-driven Coevolution

In this section, we analyse how novelty-driven cooperative coevolution can overcome the problem of premature convergence. We compare fitness-driven coevolution (*Fit*) with team-level novelty (*NS-Team*), individual-level novelty (*NS-Ind*), and a combination of the two (*NS-Mix*). In the novelty-based approaches, a multiobjective algorithm, NSGA-II, is employed to combine the novelty and fitness objectives, as described in Section 3. Based on the previously discussed results, we did not use random collaborations in any of the methods: each individual is evaluated together with the individuals of the other populations that obtained the highest fitness scores in the previous generation. Three predators were used in all experiments.

Overcoming premature convergence

Figure 5 (left) shows the highest team fitness scores achieved with each method, for each level of task difficulty. Figure 5 (right) shows the behavioural dispersion of the best-of-generation teams, which, as discussed above, can be a good indicator of premature convergence. Figure 6 shows the highest fitness score achieved at each generation, averaged over the 30 evolutionary runs.

As discussed in the previous section, the performance of fitness-driven coevolution drastically decreases as the difficulty of the task is increased. *NS-Team*, on the
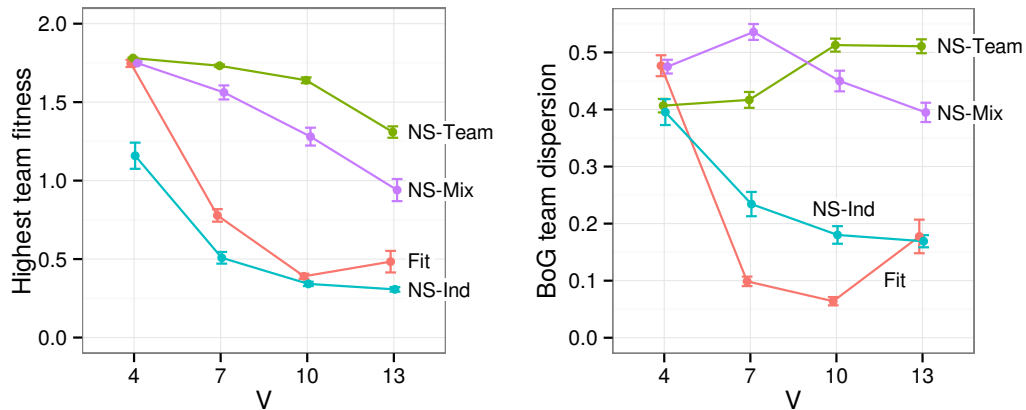
Figure 5: Left: Highest team fitness scores achieved in each evolutionary run with the different methods, for each task setup with varying task difficulty (V). Right: Behavioural dispersion of the best-of-generation teams. Standard error bars are shown.
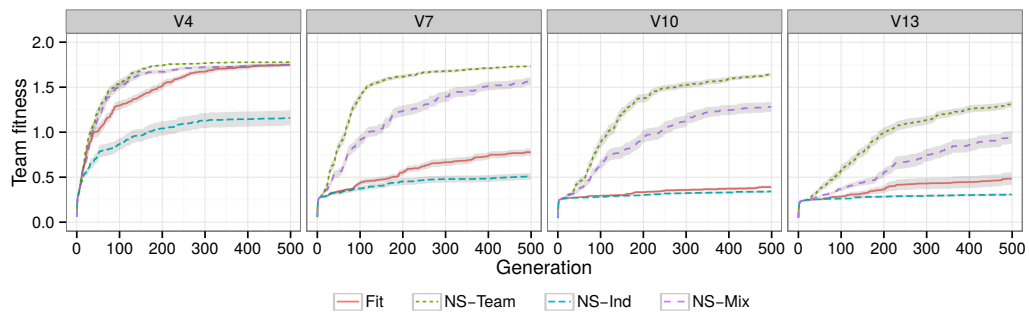

Figure 6: Performance of fitness-based evolution and the novelty-based approaches in each task setup. The plots show the highest team fitness scores achieved so far at each generation, averaged over 30 runs for each method. The grey areas depict the standard error.

other hand, can consistently evolve effective solutions (fitness > 1.0) in all task setups. The average team fitness of the best solutions evolved by *NS-Team* is significantly superior to the solutions evolved by *Fit* in all setups except the easiest one (adjusted[3] $p < 0.001$, Mann-Whitney test). *NS-Team* was clearly the highest performing approach among the novelty variants, while *NS-Ind* displayed the lowest performance. *NS-Ind* could not consistently evolve effective solutions, even in the easiest task setup, and was significantly inferior to all other novelty-based methods (adjusted $p < 0.001$). The performance of *NS-Mix* was significantly superior to *NS-Ind* in all setups, but it was inferior to *NS-Team* (adjusted $p < 0.01$). *NS-Team* could also achieve higher quality solutions earlier in the evolutionary process, see Figure 6.

As the results in Figure 5 (right) show, both *NS-Team* and *NS-Mix* were able to overcome the issue of premature convergence to stable states. In Figure 7, we also show the dispersion patterns of the best-of-generation teams, for a representative evolutionary run of each setup. Although it is possible to observe that *NS-Team* can still get attracted

---

[3]When multiple comparisons within the same set of results were made, the p-values were adjusted with the Holm-Bonferroni correction.
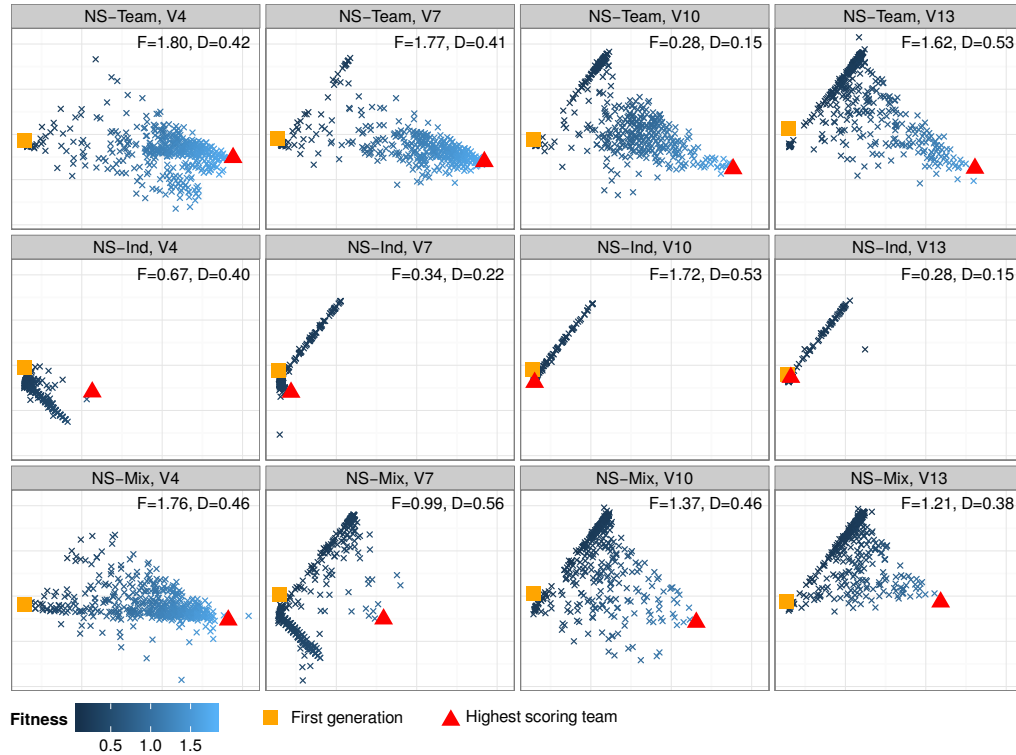
Figure 7: Behaviour of the best-of-generation teams in representative evolutionary runs. The behaviour space was reduced to a two-dimensional space with Sammon mapping.

to low-quality regions of the collaboration space (especially in *V10* and *V13*), *NS-Team* is ultimately capable of escaping these regions, and can reach high-quality collaborations. *NS-Ind*, on the other hand, was mostly ineffective. Rewarding novel individual behaviours was not an effective strategy to avoid premature convergence to narrow regions of the team behaviour space.

Exploration of the behaviour space

We resorted to the dispersion measures that use all teams (*all team dispersion*) and all individuals (*individual dispersion*), see Section 4, to better understand the difference between the proposed novelty search implementations. The results are shown in Figure 8. Fitness-driven coevolution always displays significantly inferior degrees of dispersion (adjusted $p < 0.001$, Mann-Whitney test) when compared to *NS-Team*, both in terms of the dispersion of team behaviours (Figure 8 left), and individual behaviours (Figure 8 right). These results confirm the attraction of fitness-based coevolution to stable states, already discussed above. Novelty-driven coevolution (*NS-Team*) displays substantially different evolutionary dynamics, and does not seem to converge to specific regions of the collaboration space. It explores a much wider range of collaborations (team behaviours), and can reach more collaboration regions associated with high-quality behaviours.

As previously mentioned, the performance of individual-level novelty (*NS-Ind*)
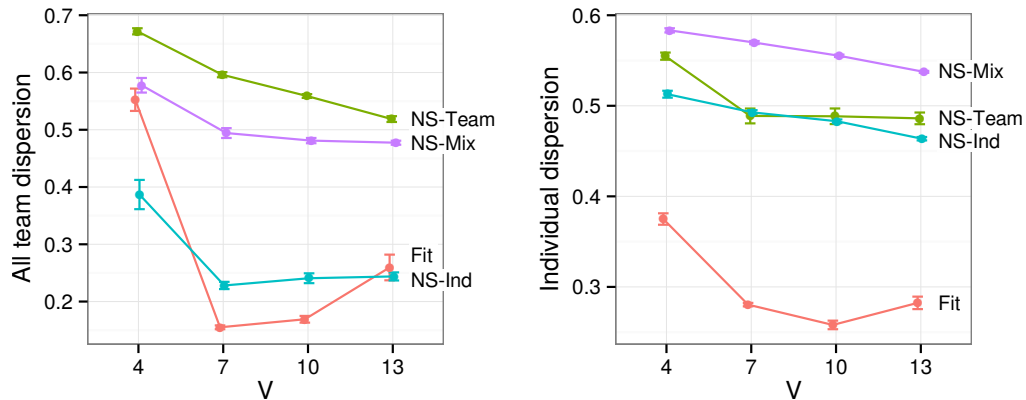
Figure 8: Analysis of team behaviour dispersion, considering all the evolved teams (left), and individual behaviour dispersion (right), with each evolutionary treatment, for task setups with varying difficulty (V).

was substantially inferior to *NS-Team*, failing to achieve effective solutions across all task setups. As the results in Figure 8 (right) show, *NS-Ind* is effective in discovering a reasonable diversity of agent behaviours, when compared to *NS-Team* and *Fit* (adjusted $p < 0.001$). However, this diversity of agent behaviours does not translate in the discovery of novel collaborations (see Figure 8, left). The individual novelty objective is not aligned with the team novelty objective, despite the similarity in the dimensions of the individual-level and team-level behaviour characterisations (see Table 1). Cooperation is not directly taken into account in *NS-Ind*, which results in poorly performing joint solutions.

The set of team behaviours discovered by *NS-Ind* was the least diverse, among the considered novelty-based treatments ($p < 0.001$). To directly encourage some degree of exploration of team behaviours, we also proposed *NS-Mix*: a combination of *NS-Team* and *NS-Ind* where the team novelty objective is added to the individual novelty and team fitness objectives. *NS-Mix* increased both the team and individual behavioural diversity when compared to *NS-Ind*. The diversity of team behaviours, however, was still significantly inferior to *NS-Team* ($p < 0.001$).

The results obtained with *NS-Ind* and *NS-Mix* suggest that favouring diversity of individual agent behaviours can actually be harmful. As each separate population is encouraged to constantly evolve towards individual behavioural novelty, it might be hard to form effective collaborations, as the individuals of a population do not have neither enough time nor incentive to adapt to the other populations. This evolutionary dynamic is contrary to what occurs in *NS-Team*, where each population can specialise in one area of the agent behaviour space at a time, thus allowing a better adaptation of the populations to each other. Overall, we showed that for the purpose of achieving effective solutions, novelty search with team-level characterisations was the most effective method of introducing novelty search in cooperative coevolutionary algorithms.

5.6 Diversity of Solutions

Besides the ability of overcoming premature convergence, novelty search has been shown capable of discovering a wide range of solutions for a given task (Gomes et al., 2013; Lehman and Stanley, 2011b). In this section, we present a qualitative analysis of

the exploration of the behaviour space and diversity of solutions evolved. To make a fair comparison between the diversity of solutions evolved, we used the task setup with three predators and $V = 4$, since this was the only setup where *Fit* and *NS-Team* achieved similar team fitness scores (see Figure 5).

The four dimensions of the behaviour characterisation were reduced to two dimensions using a Kohonen self-organising map in order to obtain a visual representation of the team behaviour space exploration. Kohonen (self-organising) maps produce a two-dimensional discretised representation of the input space, preserving the topological properties of the input space. The Kohonen map was trained with a sample (of size 25000) of all the behaviours found, and is depicted in Figure 9 (top). The individuals evolved in each evolutionary run were then mapped: each individual is assigned to the node (map region) with the closest weight vector. In Figure 9 (bottom), we show the behaviour exploration in a typical evolutionary run of *Fit* and *NS-Team*.

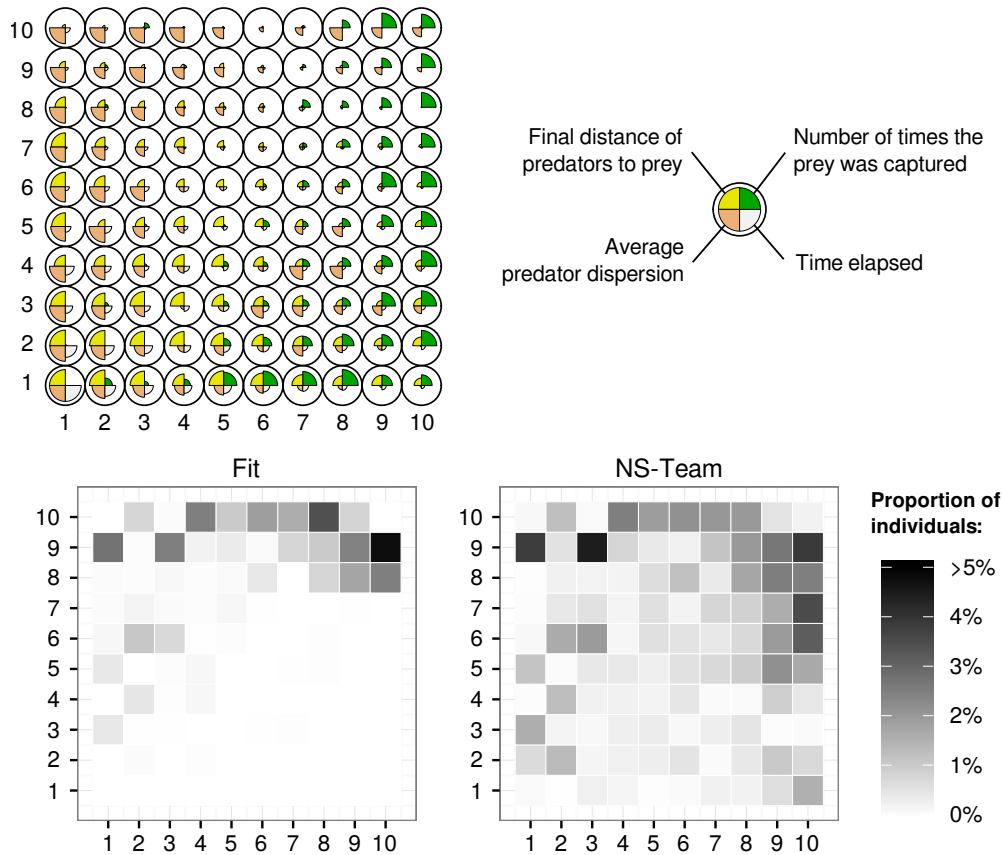As discussed in Section 5.5, fitness-driven coevolution often explores a relatively

Figure 9: Top: trained Kohonen map, where each unit represents a region of the team behaviour space. The high-quality behaviour regions (where the prey is caught most of the times) are found along column 10 and near it. Bottom: team behaviour exploration in a typical evolutionary run of fitness-based coevolution and *NS-Team*, with the easiest task setup (*V4*). The darker a region, the more individuals were evolved belonging to that behaviour region.
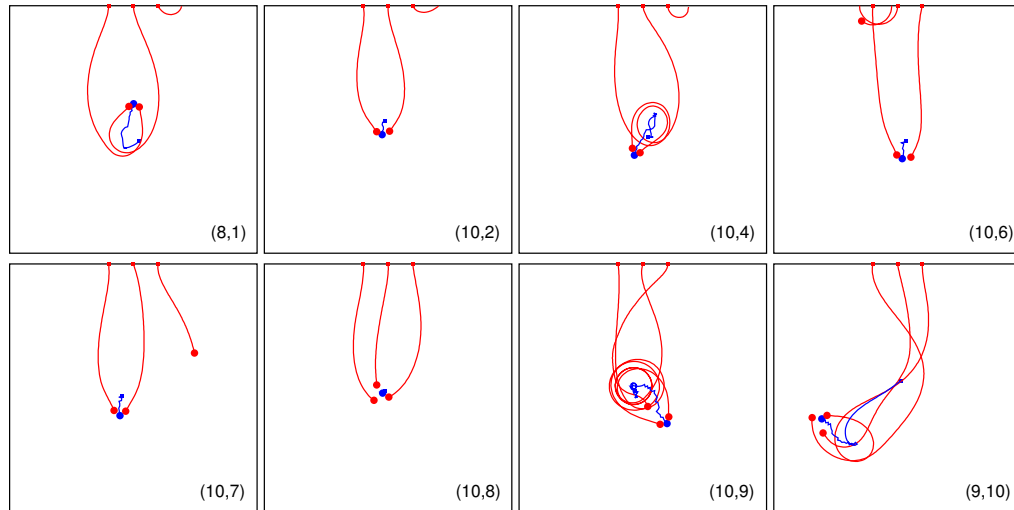
Figure 10: Examples of solutions evolved by *NS-Team* in the *V4* task setup, found in the behaviour regions associated with high-quality solutions (the numbers indicate the coordinates in the plots in Figure 9). The three preys (red) start at the top of the arena, and the prey (blue) starts in the centre.

narrow region of the behaviour space (corresponding to the top-right corner of the map, Figure 9), and converges to solutions in regions (10,9) and (10,8) (see Figure 9) in all evolutionary runs. *NS-Team* evolves individuals that cover a wider range of behaviour regions, and can find diverse high-quality solutions. These results are consistent with the exploration measures in Section 5.5. To confirm the diversity of solutions, we inspected the highest scoring solutions found in the high-quality behaviour regions. Figure 10 depicts typical movements of the predators and the prey in the different solutions. It is noteworthy that, for this task difficulty level ($V = 4$), *NS-Team* discovered solutions where only two predators actually chase the prey (see for instance regions (8,1) and (10,2)), which highlights the diversity of team behaviours that *NS-Team* can evolve.

5.7   Scalability with Respect to Team Size

We conducted evolutionary runs in setups with between two and seven predators to assess the scalability of *NS-Team* with respect to the number of populations, see Figure 11 (left). To assess if *NS-Team* is able to take advantage of the higher number of available agents, we analyse how many predators actually participate in catching the prey, compared to the total number of predators. We consider a predator as *participant* if it is near the prey (within $1.5 \times V$) in the moment the prey is caught, as the predators typically surround the prey in order to catch it (see for instance Figure 10). Figure 11 (right) shows the mean number of participant predators in each setup, considering the best-of-generation individuals only.

As the results in Figure 11 (left) show, adding more predators to the system never negatively impacts the performance of *NS-Team*. In the most challenging setups, *V10* and *V13*, adding more agents always resulted in a significant improvement of the team fitness scores achieved by *NS-Team* ($p < 0.05$, Mann-Whitney). The results in Figure 11 (right) confirm that the highest scoring solutions take advantage of the higher number of predators available, even though a smaller number of agents is often enough to solve
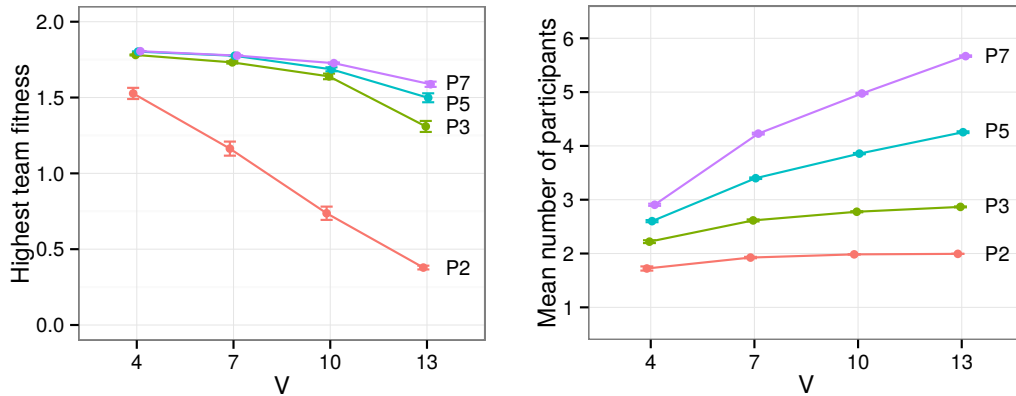
Figure 11: Left: Highest team fitness scores achieved with *NS-Team* in task setups with multiple combinations of number of predators and prey vision range V. Right: Mean number of participant predators in the best-of-generation solutions evolved in each setup.

the task. For a given task difficulty level (V), adding more predators always resulted in a significantly higher number of participant predators ($p < 0.001$). Overall, our results suggest that *NS-Team* can scale with the number of populations – it performed well with up to seven populations, and was able to take advantage of all or most of the agents available.

5.8 Combination of Novelty and Team Fitness

In all the experiments described so far in this section, the novelty-based approaches always consisted of one or two novelty objectives combined with the team fitness objective through a multiobjective algorithm, as described in Section 3. This choice was based on previous findings that show that the combination of novelty and fitness is the most effective way of applying novelty search in optimisation problems (Gomes et al., 2015c; Lehman et al., 2013). Nevertheless, a number of previous works also show that, in some situations, novelty search alone might suffice to solve challenging tasks (Lehman and Stanley, 2011a; Gomes et al., 2013). In this case, the only drive in the evolutionary process is behavioural novelty and the quality of the evolved solutions is completely ignored.

In this section, we evaluate the necessity of combining novelty with team fitness. We only focus on *NS-Team*, since it is clearly the best performing approach. We introduce *NS*-Team*, which is implemented similarly to *NS-Team* (see Algorithm 1), with the following differences:

1. The selection score of each individual is simply the team novelty score that individual obtained – the behavioural novelty of the team with which the individual was evaluated.

2. The representative of each population is the individual that obtained the highest novelty score in the previous generation.[4]

---

[4]Other possibilities for the selection of the representative individuals could be considered, for instance using the individual with the highest fitness score in the previous generation. We chose the most novel individual as the representative in order to avoid introducing any biases from the fitness function in the evolutionary process.
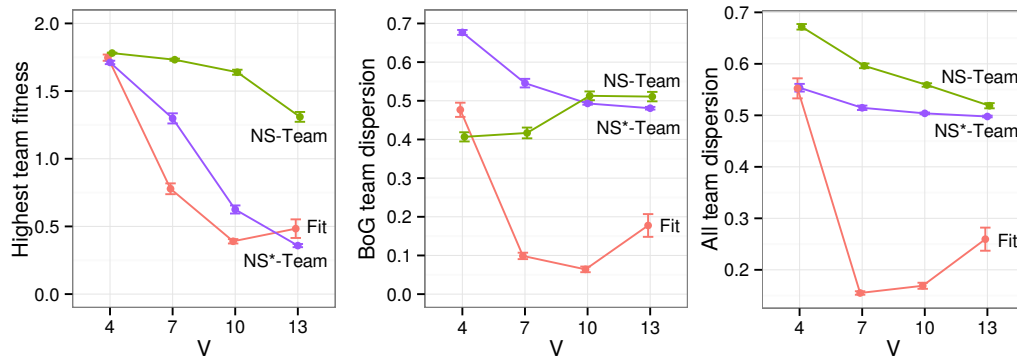
Figure 12: Comparison of pure novelty search (*NS*-Team*) and *NS-Team*, where there is a multiobjectivisation with the novelty and team fitness objectives. Left: Highest team fitness scores achieved in each evolutionary run, with each approach and in each task setup. Middle: Behavioural dispersion of the best-of-generation teams. Right: Behavioural dispersion of all the evolved teams.

In Figure 12, we compare *NS-Team* with *NS*-Team*, and present fitness-driven coevolution as baseline. We use different task difficulty levels (V), and the number of predators is always three. The results show that the performance of pure novelty search (*NS*-Team*) is significantly inferior to the multiobjectivisation of novelty and team fitness (*NS-Team*) across all task setups ($p < 0.001$, Mann-Whitney). Nonetheless, it is noteworthy that the performance of *NS*-Team* was never significantly inferior to fitness-driven coevolution, and actually managed to achieve a significantly higher performance in the *V7* and *V10* task setups ($p < 0.001$). As novelty search encourages the exploration of behaviour regions that have not been visited so far, the coverage of the behaviour space is greater, and it is thus more likely to discover solutions in behaviour regions associated with high fitness scores. Our results thus suggest that novelty-driven coevolution can achieve high-quality cooperative solutions without explicitly looking for them in the first place, which is consistent with previous non-coevolutionary novelty search studies (Gomes et al., 2013; Lehman and Stanley, 2011a).

The results in Figure 12 (middle) show that pure novelty search is effective in avoiding convergence to stable states across all setups, and can find a good diversity of team behaviours (Figure 12, right). The lack of a team fitness objective, however, makes the behavioural exploration rather ineffective: in the more demanding task setups, pure novelty search fails to reach the high-quality regions of the collaboration space.

6 Experiments with the Multi-rover and Herding Tasks

We evaluate the proposed approaches in two additional robotics tasks, the multi-rover task and the herding task, to assess the general applicability of novelty-driven cooperative coevolution. These tasks require more complex controllers than the predator-prey task (Section 5), as the agents have a higher number of sensors and effectors. To deal with this higher complexity, we use the NEAT algorithm (Stanley and Miikkulainen, 2002) to evolve the neural controllers for the agents.
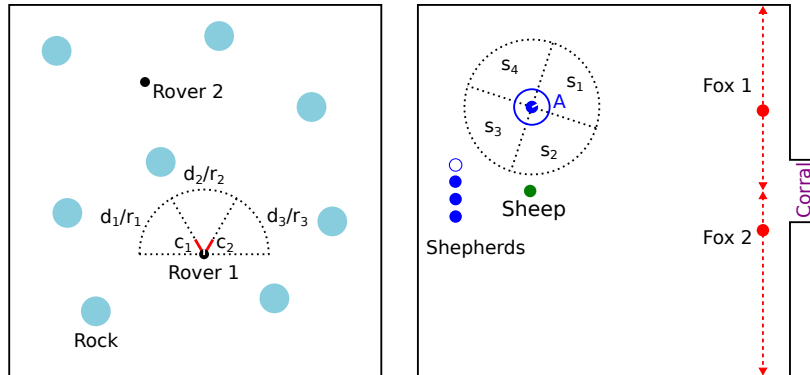
Figure 13: Left: an example of the initial conditions in the multi-rover task. Both rovers have the same sensor and effector setup, although the setup is only shown for *Rover 1*. Right: initial conditions in the herding task. The shepherds start in a linear formation. In the figure on the right we have moved the top-most shepherd to show the sensor setup. The two foxes are placed randomly along the respective line segment.

6.1 Multi-Rover Task Setup

The multi-rover task requires a team of vehicles (rovers) to find and collect features of interest (rocks) in the environment. More than one rover is needed to collect one rock, and each rover must use a different actuator. Each rover can only use one actuator at a time. Therefore, behaviour specialisation and cooperation in the team of rovers is required to solve the task. Our version of the task is similar to the multi-rover task presented by Nitschke et al. (2009), but in our setup, we use only two rovers and one type of rock. The task is challenging because the rovers must find each other in the environment, then they also need to find the rocks, and complement each other to successfully collect them. The experimental parameters are listed in Appendix A.

The task environment is depicted in Figure 13 (left). Eight rocks are placed randomly inside an arena bounded by walls. The two rovers start in random locations and with random orientations. Each rover has the following sensors: (i) two short-range sensors (c_1, c_2) to detect collisions, (ii) three sensors for the detection of rocks ($r_1 - r_3$), (iii) three sensors for the detection of the other rover ($d_1 - d_3$), and (iv) one sensor that returns the type of the actuator currently used by the nearby rover. Two outputs control the linear speed and turning angle of the rover, and two other outputs control the two actuators. When an actuator is activated, the rover remains still. To collect a rock, the two rovers need to be simultaneously over the rock. Then, one rover needs to activate its *type 1* actuator, while the other rover activates its *type 2* actuator. The rock disappears from the environment when it is collected.

The fitness function F_r corresponds to the number of rocks collected during the simulation trial. The team-level behaviour characterisation $\beta_r(t)$ is composed of four features: (i) mean distance between the rovers; (ii) mean movement speed, averaged over the two rovers; (iii) mean distance of each rover to the nearest rock, averaged over the two rovers; and (iv) number of rocks collected. The individual-level characterisation $\beta_r(a)$ is composed of the following features: (i) mean distance between a and the nearest rock; (ii) mean movement of a; (iii) for how long the *type 1* actuator was activated by a; and (iv) for how long the *type 2* actuator was activated by a. All means are taken over the simulation time, and all features are normalised to the range [0,1]. Each

team of individuals is evaluated in ten independent simulation trials.

6.2 Herding Task Setup

In the herding task (Potter et al., 2001), a group of shepherds must drive one or more sheeps into a corral. Additionally, foxes can also be present, which try to capture the sheep, and must be kept away by the shepherds. In our task setup, there are four shepherds, one sheep, and two foxes. As shown by Potter et al. (2001), the presence of foxes increases the number of skills required to solve the task, and as such behavioural specialisation within the shepherds group might be required to solve the task. Only the controllers for the shepherds are evolved. The shepherds are physically homogeneous.

The initial conditions of the herding task are depicted in Figure 13 (right). Each fox is placed randomly at the right side of the arena. The shepherds and the sheep have fixed initial positions. Each shepherd has the following sensors: (i) four sensors that return the distance of the nearest shepherd ($s_1 - s_4$); and (ii) eight sensors that return the distance and relative orientation of the sheep, the two foxes, and the centre of the corral. The two outputs control respectively the linear speed and turning angle of the shepherd. The experimental parameters are listed in Appendix A.

When a shepherd approaches the sheep or one of the foxes (distance inferior to the action range A), the sheep/fox moves away from that shepherd. The sheep is otherwise passive. The behaviour of the foxes is preprogrammed: each fox tries to intercept the sheep by estimating its future position and by heading in that direction. A trial ends when the sheep enters the corral or is captured by a fox.

The fitness function rewards the shepherds for getting the sheep closer to the corral, and in case the sheep is successfully corralled, for the amount of time it took:

$$F_h = \begin{cases} 2 - \tau/T & \text{if sheep is corralled} \\ max(0, 1 - d_f/d_i) & \text{otherwise} \end{cases}, \tag{5}$$

where τ is the number of time steps elapsed, T is the maximum trial length, d_f is the final distance of the sheep to the corral, and d_i is the initial distance.

The team-level behaviour characterisation $\beta_h(t)$ describes the effects of the shepherds on the sheep: (i) final distance of sheep to the corral; (ii) mean distance of sheep to the border of the arena; (iii) mean distance between the sheep and the foxes; and (iv) trial length. The agent-level characterisation $\beta_h(a)$ describes the role of shepherd a: mean distance of a to (i) the sheep, (ii) the corral, (iii) the first fox, and (iv) the second fox. All means are taken over the simulation time, and all features are normalised to the range [0,1]. Each team is evaluated in ten independent simulation trials.

6.3 Evolutionary Setup

NEAT (Stanley and Miikkulainen, 2002) is a widely used neuroevolution algorithm, and one of the most successful approaches in the evolutionary robotics domain. NEAT simultaneously optimises the connection weights and the topology of the neural network. It employs speciation and fitness sharing to maintain high genotypic diversity in the population, and to protect topological innovations. It should be noted, however, that in the evolutionary robotics domain, the genotypic diversity is largely unrelated to behavioural diversity. Very similar behaviours can be originated by totally different genotypes (neural networks), while similar neural networks can originate different behaviours (see for instance Mouret and Doncieux, 2012). Increasing the genotypic diversity therefore does not necessarily cause a greater exploration of the behaviour space (Gomes et al., 2015c).
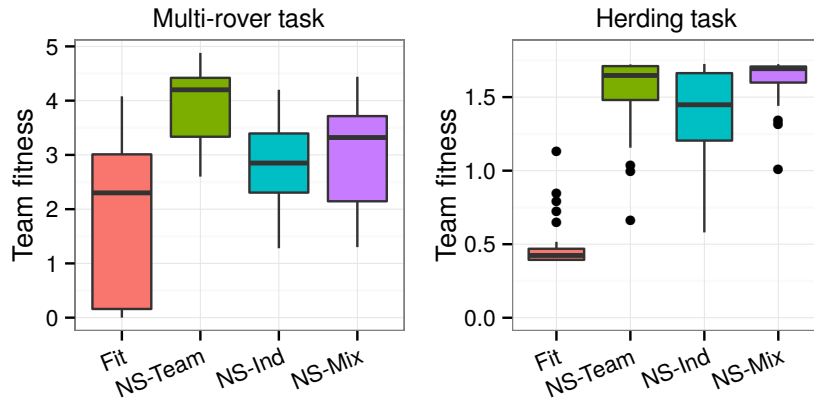
Figure 14: Highest team fitness scores achieved with each method and task. Each treatment was repeated in 30 independent evolutionary runs. The whiskers represent the highest and lowest value within 1.5 IQR.

The parameters of the NEAT algorithm were the same for both tasks, and are listed in Appendix A. Novelty-driven cooperative coevolution was implemented over NEAT as described in Section 3, and with the same parameter values as the predator-prey experiments (Section 5.2). In order to implement the NSGA-II algorithm in NEAT, the individuals were scored according to their Pareto front and crowding distance, respecting the original NSGA-II ranking, and the selection and speciation processes relied on these scores.

6.4 Results

Figure 14 summarises the highest team fitness scores achieved in each evolutionary run, for each method and each task. Overall, the results obtained in both tasks are consistent with the results obtained with the predator-prey task, presented in Section 5. In the herding task, *Fit* displays a very poor performance, and the best solutions consistently failed to drive the sheep towards the corral. In the multi-rover task, the performance of *Fit* displayed a very high variability: some runs achieved good solutions, where a reasonable number of rocks is collected, while others failed completely, with not a single rock collected. In both tasks, *NS-Team* significantly outperforms *Fit* (Mann-Whitney test, adjusted $p < 0.001$). In the herding task, *NS-Team* consistently evolved solutions where the sheep was corralled (fitness over 1), and in the multi-rover task it consistently evolved solutions where at least three rocks were collected.

The relative performance of the novelty variants is also similar to the previous results. Novelty with team-level characterisations (*NS-Team*) displayed the highest performance in the multi-rover task ($p < 0.01$), and a similar performance to *NS-Mix* in the herding task ($p = 0.35$). Novelty based on individual-level characterisations (*NS-Ind*) was always significantly inferior to *NS-Team* (adjusted $p < 0.05$). The herding task was the only one where *NS-Mix* was able to match the performance of *NS-Team*. One possible reason for this result is that the herding task requires division of labour, rather than tight cooperation between the agents: each agent can perform its subtask independently, without relying on other agents, for example chasing one specific fox, or attempting to corral the sheep.

The analysis of the best-of-generation (BoG) teams (see Figure 15) reveals that *Fit*
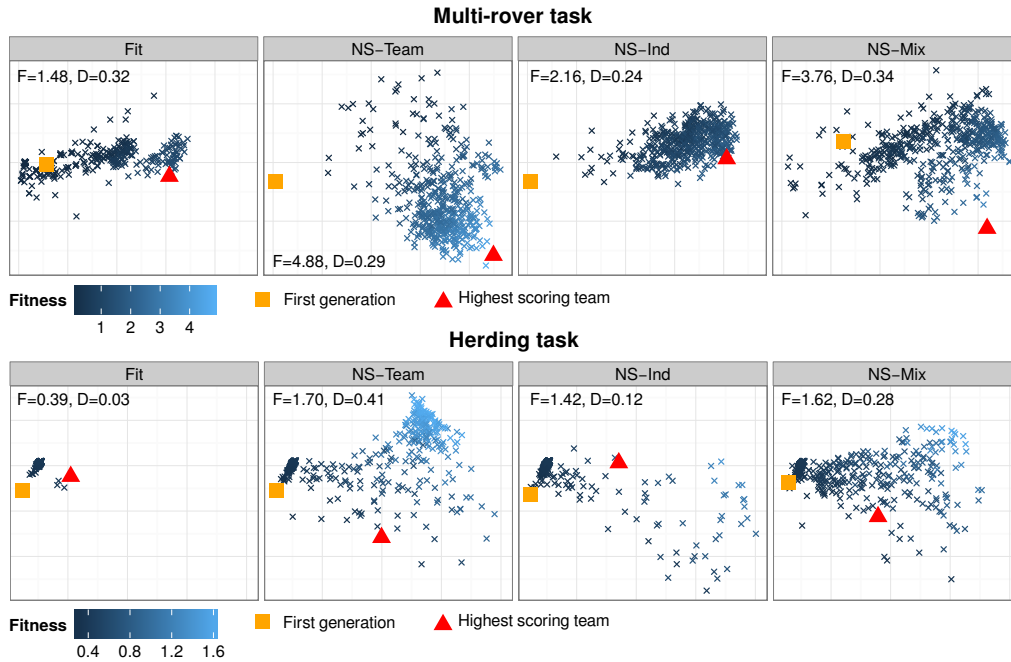
Figure 15: Behaviour of the best-of-generation teams in representative evolutionary runs. The behaviour space was reduced to a two-dimensional space with Sammon mapping.

fails in the herding task because it strongly converges to a very narrow region in the team behaviour space. In the multi-rover task, the problem of premature convergence is not so severe, as evidenced by the relatively high levels of BoG team dispersion, see Figure 16. The performance of *Fit* was, however, significantly inferior to the other methods that obtained similar values of BoG team dispersion. The results in Figure 15 (top) suggest an explanation for this phenomena: although *Fit* achieves a fair amount of behavioural exploration, this exploration is focused on a region that is distant from high-quality solutions (bottom-right corner of the space).

In both tasks, *NS-Team* and *NS-Mix* display the highest levels of team behaviour dispersion, considering all teams ($p < 0.001$). *NS-Ind* has relatively high levels of individual behaviour dispersion in both tasks, but they neither translate into a higher diversity of team behaviours, nor the achievement of higher quality solutions.

7 Discussion

Premature convergence to stable states

Our results with a simple genetic algorithm and the predator-prey task first showed that fitness-based coevolution (*Fit*) often fails as the task becomes more complex. The populations often converge to suboptimal equilibria, and therefore fail to achieve effective solutions for the task. In Section 6, we tried fitness-based coevolution with a more elaborate neuroevolution algorithm — NEAT, that sustains high genetic diversity in the populations. We experimented with two additional tasks: multi-rover and herding. Even with higher genetic diversity in the populations, fitness-based coevolution often converged prematurely in these tasks. As previous works have shown (Wiegand,
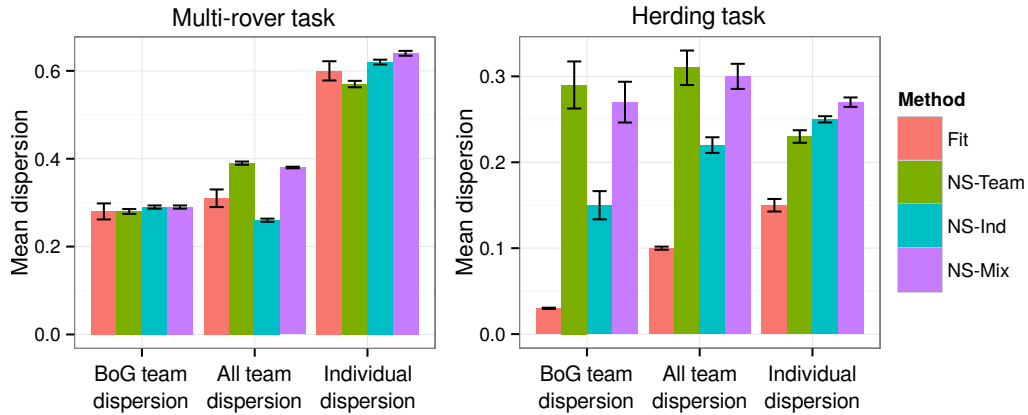
Figure 16: Mean dispersion of the best-of-generation individuals, team behaviour exploration, and individual behaviour exploration (see Section 4) for each evolutionary setup. The respective standard error bars are shown.

2004; Panait et al., 2006b), premature convergence is not necessarily caused by lack of genetic diversity, but by a strong attraction to stable states: populations can become over-adapted to one another. The issue is not related to the evolutionary algorithm itself, but to the way the population individuals are rewarded in a coevolutionary algorithm.

As suggested in previous works (Panait, 2010), we increased the number of collaborations with which an individual is evaluated, to increase the likelihood of convergence to (near-)optimal solutions. This strategy, however, only worked in the two-population setup. In the three-population setups, increasing the number of random collaborations failed to improve the performance of fitness-based coevolution. Our results showed that increasing the number of collaborations does not help the coevolutionary algorithm to escape stable states.

Novelty-driven coevolution

To overcome convergence to suboptimal equilibria, we proposed to add a novelty score in the evaluation of the individuals of each population. We assessed three cooperative coevolutionary algorithms based on novelty search, each with a different way of computing the novelty scores: (i) novelty based on team-level behaviour characterisations (*NS-Team*), (ii) based on agent-level characterisations (*NS-Ind*), and (iii) a combination of the two (*NS-Mix*). In all methods, we used a multiobjective algorithm, NSGA-II, to combine the novelty and team fitness objectives. In the case of *NS-Mix*, three objectives were used: individual novelty, team novelty, and team fitness.

Our results clearly revealed that the most effective way of introducing novelty search in CCEAs is *NS-Team*. The relative performance of the novelty-based methods was consistent across all the considered task setups: *NS-Team > NS-Mix > NS-Ind*. The algorithms based on individual-level evaluations (*NS-Ind* and *NS-Mix*) could evolve more diverse agent behaviours, but typically this did not translate to more diverse or effective team solutions. Our results suggest that encouraging novelty of agent behaviours can actually be harmful for the adaptation of the populations to one another. *NS-Ind* was always the lowest performing novelty-based method.

When compared to fitness-driven coevolution, *NS-Team* evolved significantly bet-

ter solutions for almost all task setups. The more challenging the task setup was, the greater the performance difference between *NS-Team* and *Fit*, as *NS-Team* successfully managed to avoid convergence to stable states. *NS-Team* could also discover a greater diversity of team behaviours in a single evolutionary run. In the predator-prey task, we showed that *NS-Team* evolved a diverse set of solutions for the task, whereas *Fit* tended to focus on a single class of solutions.

Besides the three tasks used in this paper, recent works have also confirmed the advantages of *NS-Team* using two different simulated multirobot tasks: a keepaway soccer task (Gomes et al., 2014b), and a cooperative item collection task (Gomes et al., 2015a). In future work, we will evaluate the solutions evolved by *NS-Team* in real multirobot systems, to validate that the diversity and quality of evolved behaviours is transferable to real systems.

Scalability with the number of agents

In the predator-prey task, we evaluated *NS-Team* in task setups varying from two to seven agents, with each agent evolving in a separate population. *NS-Team* scaled well with the number of agents, evolving good solutions for all team sizes. For the same task setup, increasing the number of predators never harmed the performance of *NS-Team*. Our analysis revealed that *NS-Team* can take advantage of most of the available agents to solve the task, even when a lower number of agents is actually enough, which suggests that *NS-Team* can evolve cooperation for a relatively large number of agents.

In future work, we will evaluate the proposed approach with larger multiagent systems. One concern is that with relatively large teams, one particular agent might not have a significant impact in the behaviour of the team as a whole, thus resulting in less accurate fitness and novelty gradients. In recent work (Gomes et al., 2015b), we proposed an algorithm, *Hyb-CCEA*, for the evolution of partially heterogeneous teams, that is, heterogeneous teams with homogeneous sub-teams. In ongoing work, we are assessing how *Hyb-CCEA* can be combined with novelty-driven coevolution. We hypothesise that by having multiple agents share the same controller, it might be easier for evolution to modify the behaviour of the team as a whole.

Parameter sensitivity and generalisation

One of the main concerns when using novelty-based techniques is the necessity of providing a behaviour similarity measure (Kistemaker and Whiteson, 2011). For each of the considered tasks, we chose a small number of behavioural traits that intuitively described the behaviour of the agents in the context of the task objective. The chosen behavioural traits were based on *systematically derived behaviour characterisations* (Gomes et al., 2014c): they were always directly observable in the task, and did not require complex calculations and/or fine tuning. Although the definition of behavioural measures did not pose a problem in our tasks, in future work we will experiment with task-independent (generic) behaviour distance measures (Doncieux and Mouret, 2010; Gomes and Christensen, 2013), and other techniques that make novelty less sensitive to the choice of the distance measure (Doncieux and Mouret, 2013).

In our experiments, we used both team-level (in *NS-Team*) and individual-level characterisations (in *NS-Ind* and *NS-Mix*). The definition of individual-level characterisations can be more delicate, as it can be hard to describe the behaviour of a single agent in a way that it is mostly independent from the other cooperating agents. Our results, however, showed that the use of individual-level characterisations never brought significant advantages over *NS-Team*, which uses only team-level characterisations.

Another significant algorithmic parameter is the combination of novelty and fit-

ness scores. In most of the experiments presented in this paper, the novelty objectives were always combined with a team fitness objective via multiobjective optimisation (Mouret, 2011; Gomes et al., 2015c; Lehman et al., 2013), which does not rely on domain-specific parameters. One possible direction of future work would be to investigate techniques for combining novelty and team fitness specifically tailored for coevolutionary algorithms.

Our tasks were based on two different neuroevolution algorithms: a simple genetic algorithm with direct encoding and no crossover, and NEAT (Stanley and Miikkulainen, 2002), a neuroevolution algorithm with topology evolution, crossover, and fitness sharing. Novelty-driven coevolution performed well with both algorithms, and the relative performance of the methods was consistent, which suggests that the proposed methods are independent of the underlying evolutionary algorithm. As novelty-driven coevolution is essentially just a different approach for scoring the individuals, in future work we will assess how novelty-driven coevolution can be integrated in other more elaborate coevolutionary algorithms, such as CONE (Nitschke et al., 2009) or Hyb-CCEA (Gomes et al., 2015b).

8 Conclusion

In this paper, we addressed the problem of premature convergence in cooperative co-evolutionary algorithms (CCEAs), a well-known problem that compromises the use of CCEAs as optimisation tools. We showed that rewarding individuals that cause novel team behaviours (*NS-Team*) is a promising approach to avoid convergence to suboptimal equilibria. *NS-Team* consistently outperformed traditional fitness-driven coevolution across multiple task setups, achieving higher team fitness scores and a wider diversity of effective solutions. The proposed approach only requires one collaboration to evaluate each individual, which contrasts with previous approaches that relied on using a large number of collaborations to overcome premature convergence. Therefore, *NS-Team* can be used in problem domains where the evaluations are costly, such as in embodied multiagent systems, and scales well with the number of populations in the coevolutionary algorithm. *NS-Team* has been successfully used in a total of five different multirobot tasks so far, which confirms the general applicability of the proposed approach. To the best of our knowledge, our approach is the first that overcomes the problem of convergence to suboptimal equilibria in the domain of embodied multiagent systems, and in a cooperative coevolutionary algorithm with more than two populations.

Acknowledgements

A Experimental Parameters

All the evolutionary techniques described in this paper were implemented over ECJ.[5] The simulated tasks were implemented over MASON (Luke et al., 2005).[6] The source code can be found here: `https://github.com/jorgemcgomes/mase/releases/tag/ECJ_novelty_driven_coevolution`. The parameters of the novelty search algorithm were the same for all experiments. All experiments with predator-prey task used the same genetic algorithm parameters, and the experiments with the multi-rover and herding task used the same NEAT parameters. The parameters are listed in Table 2.

Table 2: Parameters used in the experiments.

Parameter	Value	Parameter	Value	Parameter	Value
Novelty search					
Novelty nearest-k	15	Add archive prob.	2.5%	Max. archive size	2000
Genetic Algorithm					
Population size	150	Elite size	5	Tournament size	5
Mutation type	Gaussian	Gene mutation prob.	0.05	Crossover prob.	0.5
Crossover type	one point				
NEAT					
Population size	150	Target species count	5	Crossover prob.	0.2
Recurrency allowed	true	Mutation prob.	0.25	Prob. add link	0.05
Prob. add node	0.03	Prob. mutate bias	0.3		
Predator-prey task					
Arena side length	100 u	Prey placement area	100 u^2	Max. trial length	300 s
Prey speed	1 u/s	Pred. linear speed	1 u/s	Pred. turn speed	45°/s
Num. predators	2–7	Prey vision range	4–13 u		
Multi-rover task					
Arena side length	150 u	Max. trial length	1000 s	Max. linear speed	1 u/s
Max. rotation speed	23°/s	Sensor range	25 u	Min. actuator activation time	25 s
Rock diameter	12 u	Agent diameter	4 u		
Herding task					
Arena side length	150 u	Max. trial length	500 s	Sheep speed	1 u/s
Fox speed	1 u/s	Shepherd linear speed	1 u/s	Shep. turn speed	23°/s
Action range	5 u	Shep. sensor range	25 u		

References

Cuccu, G. and Gomez, F. J. (2011). When novelty is not enough. In *European Conference on the Applications of Evolutionary Computation (EvoApplications)*, volume 6624 of *LNCS*, pages 234–243. Springer-Verlag.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons.

Doncieux, S. and Mouret, J.-B. (2010). Behavioral diversity measures for evolutionary robotics. In *Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE Press.

Doncieux, S. and Mouret, J.-B. (2013). Behavioral diversity with multiple behavioral distances. In *Congress on Evolutionary Computation (CEC)*, pages 1427–1434. IEEE Press.

---

[5]`http://cs.gmu.edu/~eclab/projects/ecj/`
[6]`http://cs.gmu.edu/~eclab/projects/mason/`

Doncieux, S. and Mouret, J.-B. (2014). Beyond black-box optimization: a review of selective pressures for evolutionary robotics. *Evolutionary Intelligence*, 7(2):71–93.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Gomes, J. and Christensen, A. L. (2013). Generic behaviour similarity measures for evolutionary swarm robotics. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 199–206. ACM Press.

Gomes, J., Mariano, P., and Christensen, A. (2014a). Novelty search in competitive coevolution. In *Parallel Problem Solving from Nature PPSN XIII*, volume 8672 of *LNCS*, pages 233–242. Springer.

Gomes, J., Mariano, P., and Christensen, A. L. (2014b). Avoiding convergence in cooperative coevolution with novelty search. In *International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pages 1149–1156. IFAAMAS.

Gomes, J., Mariano, P., and Christensen, A. L. (2014c). Systematic derivation of behaviour characterisations in evolutionary robotics. In *International Conference on the Synthesis and Simulation of Living Systems (ALife)*, pages 202–209. MIT Press.

Gomes, J., Mariano, P., and Christensen, A. L. (2015a). Cooperative coevolution of morphologically heterogeneous robots. In *European Conference on Artificial Life*, pages 312–319. MIT Press.

Gomes, J., Mariano, P., and Christensen, A. L. (2015b). Cooperative coevolution of partially heterogeneous multiagent systems. In *International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pages 297–305. IFAAMAS.

Gomes, J., Mariano, P., and Christensen, A. L. (2015c). Devising effective novelty search algorithms: A comprehensive empirical study. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 943–950. ACM Press.

Gomes, J., Urbano, P., and Christensen, A. (2013). Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7(2–3):115–144.

Gomes, J., Urbano, P., and Christensen, A. L. (2014d). Pmcns: Using a progressively stricter fitness criterion to guide novelty search. *International Journal of Natural Computing Research*, 4:1–19.

Gomez, F. and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 5(3–4):317–342.

Jansen, T. and Wiegand, R. P. (2004). The cooperative coevolutionary (1+1) EA. *Evolutionary Computation*, 12(4):405–434.

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Neural-Network Models of Cognition Biobehavioral Foundations*, volume 121 of *Advances in Psychology*, pages 471–495. North-Holland.

Kistemaker, S. and Whiteson, S. (2011). Critical factors in the performance of novelty search. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 965–972. ACM Press.

Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on Artificial Life (Alife XI)*. MIT Press.

Lehman, J. and Stanley, K. O. (2010). Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 103–110. ACM Press.

Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.

Lehman, J. and Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 211–218. ACM Press.

Lehman, J., Stanley, K. O., and Miikkulainen, R. (2013). Effective diversity maintenance in deceptive domains. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 215–222. ACM Press.

Liapis, A., Yannakakis, G. N., and Togelius, J. (2015). Constrained novelty search: A study on game content generation. *Evolutionary computation*, 23(1):101–129.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., and Balan, G. (2005). Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527.

Mouret, J.-B. (2011). Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics*, volume 341 of *Studies in Computation Intelligence*, pages 139–154. Springer.

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

Naredo, E. and Trujillo, L. (2013). Searching for novel clustering programs. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 1093–1100. ACM Press.

Naredo, E., Trujillo, L., and Martínez, Y. (2013). Searching for novel classifiers. In *Genetic Programming (EuroGP)*, volume 7831 of *LNCS*, pages 145–156. Springer.

Nitschke, G. S., Eiben, A. E., and Schut, M. C. (2012a). Evolving team behaviors with specialization. *Genetic Programming and Evolvable Machines*, 13(4):493–536.

Nitschke, G. S., Schut, M. C., and Eiben, A. E. (2009). Collective neuro-evolution for evolving specialized sensor resolutions in a multi-rover task. *Evolutionary Intelligence*, 3(1):13–29.

Nitschke, G. S., Schut, M. C., and Eiben, A. E. (2012b). Evolving behavioral specialization in robot teams to solve a collective construction task. *Swarm and Evolutionary Computation*, 2:25–38.

Nolfi, S. (2012). Co-evolving predator and prey robots. *Adaptive Behavior*, 20(1):10–15.

Panait, L. (2010). Theoretical convergence guarantees for cooperative coevolutionary algorithms. *Evolutionary Computation*, 18(4):581–615.

Panait, L. and Luke, S. (2005a). Cooperative multi-agent learning: The state of the art. *Autonomous Agents & Multi-Agent Systems*, 11(3):387–434.

Panait, L. and Luke, S. (2005b). Time-dependent collaboration schemes for cooperative coevolutionary algorithms. In *AAAI Fall Symposium on Coevolutionary and Coadaptive Systems*. AAAI Press.

Panait, L., Luke, S., and Harrison, J. F. (2006a). Archive-based cooperative coevolutionary algorithms. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 345–352. ACM Press.

Panait, L., Luke, S., and Wiegand, R. P. (2006b). Biasing coevolutionary search for optimal multi-agent behaviors. *IEEE Transactions on Evolutionary Computation*, 10(6):629–645.

Panait, L., Wiegand, R. P., and Luke, S. (2004). A visual demonstration of convergence properties of cooperative coevolution. In *Parallel Problem Solving from Nature (PPSN)*, volume 3242 of *LNCS*, pages 892–901. Springer-Verlag.

Popovici, E., Bucci, A., Wiegand, R. P., and De Jong, E. D. (2012). Coevolutionary principles. In *Handbook of Natural Computing*, pages 987–1033. Springer.

Popovici, E. and De Jong, K. (2005). A dynamical systems analysis of collaboration methods in cooperative co-evolution. In *AAAI Fall Symposium on Coevolutionary and Coadaptive Systems*.

Popovici, E. and De Jong, K. (2006). The dynamics of the best individuals in co-evolution. *Natural Computing*, 5(3):229–255.

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

Potter, M. A., Meeden, L. A., and Schultz, A. C. (2001). Heterogeneity in the coevolved behaviors of mobile robots: The emergence of specialists. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1337–1343. Morgan Kaufmann.

Rawal, A., Rajagopalan, P., and Miikkulainen, R. (2010). Constructing competitive and cooperative agent behavior using coevolution. In *IEEE Conference on Computational Intelligence and Games (CIG)*, pages 107–114. IEEE Press.

Risi, S., Hughes, C. E., and Stanley, K. O. (2010). Evolving plastic neural networks with novelty search. *Adaptive Behavior*, 18(6):470–491.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409.

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

Whitley, L. D. (1991). Fundamental principles of deception in genetic search. In *Foundations of Genetic Algorithms*, pages 221–241. Morgan Kaufmann.

Wiegand, R. P. (2004). *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University.

Wiegand, R. P., Liles, W. C., and De Jong, K. A. (2001). An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 1235–1245. ACM Press.

Wiegand, R. P., Liles, W. C., and De Jong, K. A. (2002). Analyzing cooperative coevolution with evolutionary game theory. In *Congress on Evolutionary Computation (CEC)*, volume 2, pages 1600–1605. IEEE Press.

Wiegand, R. P. and Potter, M. A. (2006). Robustness in cooperative coevolution. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 369–376. ACM Press.

Yong, C. H. and Miikkulainen, R. (2009). Coevolution of role-based cooperation in multiagent systems. *IEEE Transactions on Autonomous Mental Development*, 1(3):170–186.