DOI: http://dx.doi.org/10.7551/978-0-262-33027-5-ch059

Cooperative Coevolution of Morphologically Heterogeneous Robots

Jorge Gomes[1,2,3] and Pedro Mariano[3] and Anders Lyhne Christensen[1,2,4]

[1]BioMachines Lab, Lisbon, Portugal
[2]Instituto de Telecomunicações, Lisbon, Portugal
[3]Faculdade de Ciências, Universidade de Lisboa, BioISI, Portugal
[4]Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
jgomes@di.fc.ul.pt, plmariano@fc.ul.pt, anders.christensen@iscte.pt

Abstract

Morphologically heterogeneous multirobot teams have shown significant potential in many applications. While cooperative coevolutionary algorithms can be used for synthesising controllers for heterogeneous multirobot systems, they have been almost exclusively applied to morphologically homogeneous systems. In this paper, we investigate if and how cooperative coevolutionary algorithms can be used to evolve behavioural control for a morphologically heterogeneous multirobot system. Our experiments rely on a simulated task, where a ground robot with a simple sensor-actuator configuration must cooperate tightly with a more complex aerial robot to find and collect items in the environment. We first show how differences in the number and complexity of skills each robot has to learn can impair the effectiveness of cooperative coevolution. We then show how coevolution's effectiveness can be improved using incremental evolution or novelty-driven coevolution. Despite its limitations, we show that coevolution is a viable approach for synthesising control for morphologically heterogeneous systems.

Introduction

Cooperative coevolution (CCEA) has been advocated as a valuable approach for the evolution of heterogeneous multiagent systems (Potter et al., 2001). In the classic CCEA architecture (Potter and Jong, 2000), each agent evolves in an isolated population, and the individuals are evaluated by forming collaborations with individuals from the other populations. One key advantage of CCEAs is that since populations are isolated, it is possible for different populations to evolve radically different agents, with genomes of different lengths, and even to use different evolutionary algorithms. This vast heterogeneity has, however, rarely been exploited. Most previous works focus on the evolution of controllers for behaviourally heterogeneous, but morphologically homogeneous, multiagent systems (see for instance Potter et al., 2001; Yong and Miikkulainen, 2009; Nitschke et al., 2012). This means that all agents in the system have a similar complexity, similar sensor-effector capabilities, and use the same genotype representation.

Morphologically heterogeneous multirobot systems have shown significant potential in a number of applications (Howard et al., 2006; Dorigo et al., 2013; Duan and Liu, 2010). The cooperation between morphologically heterogeneous robots can, for instance, augment the capabilities of the group, allowing the achievement of tasks that are beyond the reach of a single type of robot. The behavioural control for morphologically heterogeneous systems is typically designed manually. This process can, however, be challenging, as behavioural control must integrate the capabilities of different robot types, in a way that the efficiency of the group becomes greater than if the different robot types worked independently without cooperation (Dorigo et al., 2013).

In this paper, we study if and how cooperative coevolution can be used to evolve effective controllers for agents with radically different capabilities in a task that requires tight cooperation. Cooperative coevolution is traditionally associated with a number of challenges (Wiegand, 2003) that stem from the intricate dynamics of coevolving two or more populations, where the evaluation of the individuals in one population depends on the individuals of the other populations. A key element in the evolution of cooperative behaviours is *synchronised learning* (Uchibe et al., 1998): populations should exhibit a mutual development of skills, in order to avoid loss of fitness gradients and convergence to mediocre stable states. We will study how the presence of radically different agents affects the mutual development of skills.

Our experiments are based on a simulated item collection task, where a ground robot with very limited capabilities, must cooperate with an aerial robot with a significantly more complex sensor-effector configuration. We explore different task variants to study how the differences between the agents, regarding the skills that must be evolved, affect the performance of cooperative coevolution. We then try to improve the effectiveness of coevolution with two techniques found in previous works: incremental evolution (Doncieux and Mouret, 2014) and novelty-driven cooperative coevolution (Gomes et al., 2014). We compare the advantages and drawbacks of each technique, and study how they can contribute to the effective coevolution of behaviours for multirobot systems with morphologically heterogeneous robots.

Jorge Gomes, Pedro Mariano, Anders Lyhne Christensen (2015) Cooperative Coevolution of Morphologically Heterogeneous Robots. Proceedings of the European Conference on Artificial Life 2015, pp. 312-319

Related Work

Morphologically Heterogeneous Systems

Heterogeneous multirobot systems are characterised by the morphological and/or behavioural diversity of their constituent robots. Behavioural heterogeneity is commonly employed to allow behaviour specialisation within the group (Nitschke et al., 2012). In morphologically heterogeneous systems, on the other hand, agents have varied actuation and sensing capabilities, and collaborate to take advantage of the collective set of capabilities (Dorigo et al., 2013).

The Swarmanoid project (Dorigo et al., 2013) studied morphologically heterogeneous robotic swarms. Some of the tasks that were approached include: a gap crossing task, where ground robots receive instructions from aerial robots (Mathews et al., 2010); an indoor navigation task, where aerial robots aid ground robots in navigating through cluttered environments (Ducatelle et al., 2011); and a search-and-retrieval task in a 3-D environment, where aerial, ground, and climbing robots cooperate (Dorigo et al., 2013). Other works outside the Swarmanoid project have also shown the potential of cooperation between ground and aerial robots, especially in detection, search, and rescue tasks (Duan and Liu, 2010). Aerial robots can be used to assist ground robots, providing valuable information related to the environment (Lacroix and Le Besnerais, 2011).

Morphologically heterogeneous systems also encompass systems composed of robots of a similar nature (e.g., ground robots only). Heterogeneity can be used to reduce the cost of the group, by assigning different sensor/actuator capabilities to different robots. The robots then cooperate to take advantage of each other's capabilities. In (Parker et al., 2004), capable leader robots assist sensor-limited robots in navigating indoor environments. Howard et al. (2006) extend this approach to a task where few complex robots cooperate with a large number of inexpensive robots to map the environment and establish a sensor network. Grabowski et al. (2000) also study a mapping and exploration task, using two types of ground robots equipped with complementary sensors. Candea et al. (2001) study the coordination of different robot types in the context of the RoboCup competition.

Cooperative Coevolution

In the studies on multirobot systems discussed above, distributed control was achieved by manually designing the behavioural rules of the individual robots. Previous works have shown that this can be a challenging task, since the decomposition of the desired global behaviour into individual behavioural rules is often complex and inconspicuous (Dorigo et al., 2004). This challenge is exacerbated in heterogeneous systems (Dorigo et al., 2013), as behavioural control must be able to integrate the different abilities of different robot types to work in synergy towards the achievement of a common goal. One possible solution for this problem is the use of evolutionary algorithms to synthesise robot controllers (Dorigo et al., 2004; Uchibe et al., 1998; Potter et al., 2001). Besides automating the controller design, evolutionary algorithms have the potential to discover optimal solutions for the problem, and to discover diverse, unexpected solutions.

Cooperative coevolutionary algorithms are a natural fit for the evolution of heterogeneous multiagent systems (Potter et al., 2001). The classic cooperative coevolution architecture (Potter and Jong, 2000) operates with a system comprising two or more populations. Each agent is typically assigned to a separate population. The individuals of a population are evaluated by forming teams with individuals from the other populations. The fitness gradient is therefore relative: it is strictly a function of the individuals' contribution within the context of the other populations.

One advantage of CCEAs is that, due to the separation of populations, an arbitrary level of heterogeneity can be accommodated within the system. Cooperative coevolution, however, is typically applied only to morphologically homogeneous systems, focusing only on behavioural specialisation (Potter et al., 2001; Yong and Miikkulainen, 2009; Gomes et al., 2014; Nitschke et al., 2009, 2012). There have only been few reports of successful evolution of morphologically heterogeneous systems, and in the reported studies, agents had only minor morphological differences, for instance: a keepaway soccer task where agents have different moving and passing speeds (Gomes et al., 2014); a foraging task where agents have different movement speed and sensing ranges (Yang et al., 2012); and a predator-prey task where the predators have slightly different linear and turning speeds (Blumenthal and Parker, 2004).

Overcoming Coevolution Challenges

In a CCEA, the search space of each population is constrained by the individuals in the other populations. The search space is thus constantly changing, and the fitness of an individual can vary significantly depending on with which collaborators it is evaluated. This dynamic can cause two known pathologies: convergence to mediocre stable states (also known as relative *overgeneralisation*) and loss of fitness gradient (Wiegand, 2003). Convergence to mediocre stable states occurs when populations are unable to further improve their individuals, given the current set of collaborators drawn from the other populations. Previous works have shown that CCEAs tend to gravitate towards equilibrium states, not necessarily optimal solutions, which can impair their effectiveness (Panait, 2010). Loss of fitness gradient is more common in competitive coevolution, but can also appear in CCEAs: it occurs when a population reaches a state such that the other populations lose the fitness diversity necessary for meaningful progress (Wiegand, 2003).

We hypothesise that when populations have to evolve substantially different skills, with different complexity, a lack of synchronised learning is more likely to occur, causing convergence to stable states and/or loss of fitness diversity. In

Jorge Gomes, Pedro Mariano, Anders Lyhne Christensen (2015) Cooperative Coevolution of Morphologically Heterogeneous Robots. Proceedings of the European Conference on Artificial Life 2015, pp. 312-319

this paper, we study this effect and how related issues can be mitigated. To this end, we evaluate incremental evolution (Gomez and Miikkulainen, 1997) and novelty search (Lehman and Stanley, 2011; Gomes et al., 2014) as means to improve coevolution's effectiveness.

Incremental Evolution In an incremental evolution scheme, the goal-task is decomposed in simpler tasks for which solutions are easier to find (Gomez and Miikkulainen, 1997). A series of evolutionary stages are defined by the experimenter, and evolution moves from one stage to the next when the population reaches a sufficient performance level. Incremental evolution can be accomplished by defining a series of environments with increasing complexity (*environmental complexification*), or by defining a series of sub-objectives (*staged evolution*) (Doncieux and Mouret, 2014). Incremental evolution has been used together with cooperative coevolution in a number of previous works (Nitschke et al., 2012; Yong and Miikkulainen, 2009; Uchibe and Asada, 2006). In our work, we evaluate incremental evolution, staged evolution in particular, as a way to encourage synchronised learning among the populations.

Novelty Search Novelty search is a widely recognised approach for overcoming premature convergence (Lehman and Stanley, 2011; Gomes et al., 2015). In recent work, Gomes et al. (2014) proposed a novelty-based method for avoiding convergence to equilibrium states in cooperative coevolution. The proposed technique (*NS-T*) relies on team-level behaviour characterisations, and rewards behaviourally novel collaborations in addition to high-fitness ones, as typically done in CCEAs. The team-level characterisations capture what the team as a whole achieves, without discriminating what each agent does for the team. It is shown that by rewarding individuals that cause novel collaborations, an evolutionary pressure towards novel equilibrium states is created. As there is a more effective exploration of the solution space, *NS-T* can reach collaborations associated with higher fitness scores more often than a traditional CCEA, and can evolve a diverse set of solutions for a given task in a single evolutionary run (Gomes et al., 2014).

Cooperative Item Collection Task

In the simulated task we use in this study, an aerial robot must assist a ground robot in collecting items randomly dispersed in an unbounded environment, see Figure 1. The ground robot has significantly fewer sensory capabilities than the aerial robot (detailed below). There is no direct communication between the two robots: they can only sense the relative position of each other when in close proximity. To accomplish the task, the aerial robot must learn to find the items and to guide the ground robot towards them. Complementary, the ground robot should follow the aerial robot and collect the items. The two robots must cooperate so that they do not lose track of one another.

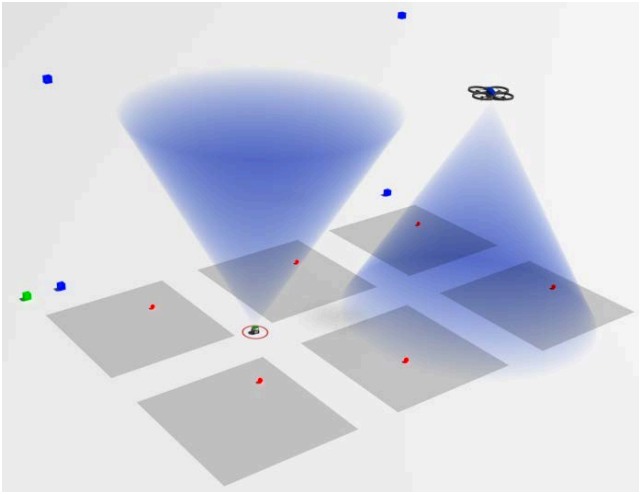

Figure 1: Cooperative item collection task. The red spheres are the items to be collected. One item is placed in each of the grey zones. The green cube indicates the starting position of the ground robot, and the blue cubes indicate the possible starting positions of the aerial robot. The blue cones depict the viewing range of the robots. The red circle around the ground robot depicts the range of its item sensor.

Robots Setup

Each robot is independently controlled by a neural network. The normalised sensor values are fed to the neural network, and the outputs control the actuators of the robot. We describe the sensor-actuator configuration of each robot below.

Ground Robot The ground robot has the ability to collect (remove) items from the environment. To collect an item, the robot simply has to pass over it. The ground robot is equipped with eight binary sensors:

- Four binary sensors to detect items within a range of 10 cm (depicted in Figure 1). Each sensor reads whether an item is present or not in the respective quadrant.
- Four binary sensors to detect the presence of the aerial robot. These sensors model an upwards-facing camera with a view angle of 60°. Each sensor indicates whether the aerial robot is present in the respective quadrant of the viewing cone. If the altitude of the aerial robot is over 250 cm, the ground robot's sensors are unable to detect it.

The two outputs of the neural controller control respectively the linear speed (within [0,3] cm/step) and the turning angle (within $[-\pi/5,\pi/5]$ rad/step) of the robot.

Aerial Robot The aerial robot is equipped with a downwards-facing camera with a view angle of 60°, and with a maximum working altitude of 250 cm. The robot is also equipped with sensors for locating itself in the environment. It has a total of 15 sensors:

- Six real-valued sensors to detect items. Each sensor returns the distance to the closest item within the respective horizontal section.
- Six real-valued sensors to detect the ground robot. Each sensor returns the distance to the ground robot if it is present in the respective horizontal section, or the maximum value if it is not.
- Current altitude.
- Distance and relative angle to the centre of the arena. These sensors are used to allow the aerial robot to localise itself in the unbounded environment.

The four outputs of the neural controller dictate the movement of the robot, relative to the current robot's heading: (i) thrust in the left-right axis, (ii) in the forward-backwards axis, (iii) in the up-down axis, and (iv) rotation around its centre. The maximum speed of the robot is 20 cm/step in any direction (seven times faster than the ground robot), and the maximum rotation speed is $\pi/10$ rad/step. There is no altitude limit. When the robot has an altitude of zero (grounded), it can only move in the upwards direction.

Task Variants

We use a number of task variants in which different skills must be learnt by the aerial robot before it can assist the ground robot in solving the task. This approach allows us to gain insight on how differences in the learning speed of the agents affect the performance of cooperative coevolution.

In all task variants, six items are spread over an area of 550x350 cm^2, with each item placed randomly inside a zone of 150x150 cm^2, see Figure 1. The environment is unbounded, and the robots are thus free to roam away from the items and from each other. A simulation ends when all items are collected, or when 1000 time steps have elapsed. Each candidate solution (pair of ground robot and aerial robot controllers) is evaluated in five independent simulations. The ground robot always starts in the upper left corner of the arena, while the initial conditions of the aerial robot depend on the task variant, see Figure 1. The task variants are described below:

***Fix-Tog* – Fixed altitude, start together** The aerial robot has no control over its altitude, but remains at the ideal sensing altitude (250 cm) throughout the whole simulation. The aerial and ground robots start together in the upper-left corner of the arena.

***Fix-Sep* – Fixed altitude, start separate** The aerial robot maintains the maximum sensing altitude. The two robots start in opposite sides of the arena, from where they cannot sense each other. This means that the aerial robot must first learn to find the ground robot.

***Var-Tog* – Variable altitude, start together** The aerial robot starts on the ground, and can freely move up and down. The aerial robot must thus learn to take-off, and control its altitude in order to optimise the sensing range. The two robots start together.

***Var-Sep* – Variable altitude, start separate** The aerial robot starts on the ground, and the two robots start in opposite sides of the arena. The aerial robot must thus initially learn to control its altitude and find the ground robot.

Methods

Base Cooperative Coevolutionary Algorithm

All the evaluated methods are implemented over the same coevolution architecture. One population evolves the controller of the ground robot, while the other population evolves the controller for the aerial robot. Every generation, each population is evaluated in turn. To evaluate an individual from one population, a team is formed with one representative from the other population. The representative from a population is the individual that obtained the highest fitness score in the previous generation, or a random one in the first generation. Only the individual currently being evaluated receives the fitness score obtained by the team. The fitness score of a team, F_i, corresponds to the number of items that were successfully collected during the simulation trial.

The neural network individuals of each population are evolved by NEAT (Stanley and Miikkulainen, 2002), a state-of-the-art neuroevolution algorithm that evolves both the weights and topology of the networks, and has been extensively used in evolutionary robotics. We use the NEAT4J[1] implementation and most of the default parameter values: each population has a size of 150 individuals, the mutation probability is 25%, crossover probability is 20%, the probability of adding a connection is 5%, the probability of adding a neuron is 3%, and the target number of species is 5.

Incremental Evolution

We define a series of sub-goals that must be accomplished before reaching the ultimate objective of collecting items (F_i). Incremental evolution was configured specifically to bridge the gap between the number and complexity of skills each robot has to evolve, by encouraging the development of skills in the aerial robot and the cooperation between the two robots. We consider the following sub-goals:

1. Minimise the difference between the aerial robot's altitude (a_t) and the near-maximum sensing altitude (A, 240 cm) over the simulation trial (T time steps). The goal is achieved when 20% of the current individuals achieve a score of at least 0.9.

$$F_a = 1 - min\left(1, \sum_{t\in[1,T]} \frac{|a_t - A|}{T \cdot A}\right) \quad (1)$$

[1] `http://neat4j.sourceforge.net`

Jorge Gomes, Pedro Mariano, Anders Lyhne Christensen (2015) Cooperative Coevolution of Morphologically Heterogeneous Robots. Proceedings of the European Conference on Artificial Life 2015, pp. 312-319

2. Maximise the time robots spend within the sensing range of one another (t_w). The goal is achieved when 20% of the current individuals achieve a score of at least 0.7.

$$F_w = t_w/T \quad (2)$$

The configuration of incremental evolution depends on the task variant. Incremental evolution was not used in the *Fix-Tog* variant. For the *Fix-Sep* variant, we considered two stages: $F_w \rightarrow F_i$. For the *Var-Tog* and *Var-Sep* variants, we used three stages: $F_a \rightarrow F_w \rightarrow F_i$.

Non-cooperative Incremental Evolution

This approach is similar to the incremental evolution described above, with one key difference: the ground robot only starts evolving when the last stage (F_i, collecting items) is reached. During the previous stages, only the aerial robot evolves, while the ground robot remains still in its initial position. The rationale of this approach is to develop essential skills in the aerial robot before starting coevolution.

Novelty-driven Coevolution

Novelty-driven coevolution was implemented as described in (Gomes et al., 2014), using the *NS-T* technique, which computes the individual's novelty scores based on the behavioural novelty displayed by the team in which the individual participated. The team behaviour characterisation, used to compute the novelty of each team, is a vector of four real values normalised to [0,1]: (i) number of items caught; (ii) time robots spent within the sensing range of one another; (iii) average distance of one robot to the other; and (iv) average distance of each robot to the closest item. The novelty score of each individual is combined with its fitness score through a linear scalarisation that gives the same weight to the novelty score and to the fitness score.

As suggested in (Gomes et al., 2015), we use a value of k=15 (nearest neighbours) for the novelty score computation, and the archive is randomly composed: every generation, four random individuals are added to the archive.

Results

Base Cooperative Coevolutionary Algorithm

We begin by studying the performance of the base CCEA in the different task variants. The highest fitness scores achieved throughout evolution are depicted in Figure 2. Besides the four task variants, we also present a baseline where the aerial robot is not present — the ground robot alone evolves to collect the items. Each evolutionary treatment was repeated in 30 independent evolutionary runs.

The results show that there are clear performance differences in the four variants. The CCEA can consistently find good solutions for the *Fix-Tog* variant. In the other task variants, where the aerial robot needs to develop certain skills before being able to cooperate, the CCEA's performance is significantly affected ($p < 0.001$, Mann-Whitney test). Coevolution displayed the lowest performance in the *Var-Sep* variant ($p < 0.001$), where the aerial robot had to evolve the most complex behaviour. It is, however, possible to achieve solutions of similar quality for all task variants, as evidenced by the results in Figure 2 (right). This suggests that the large differences in performance are not necessarily explained by task difficulty, but rather by the ineffectiveness of the evolutionary process in reaching good solutions for some variants.

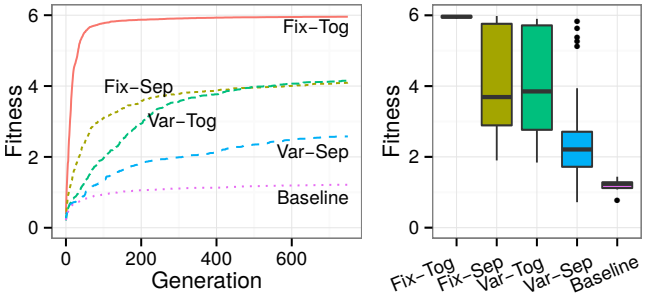

Figure 2: Fitness scores achieved by the basic *CCEA* in each of the task variants. Left: average highest fitness scores achieved at each generation. Right: boxplots of the highest scores achieved in each evolutionary run.

Table 1: N: number of successful / failed runs. Time within: average fraction of time the robots in the highest scoring solutions spent within the sensing range of each other.

	Successful runs		Failed runs	
Variant	N	Time within	N	Time within
Fix-Tog	30	96.5%	0	NA
Fix-Sep	12	70.6%	10	19.2%
Var-Tog	13	75.3%	11	14.4%
Var-Sep	5	45.1%	24	6.8%

To understand the reasons behind evolutionary failure, we looked at the highest scoring individuals evolved in each run. The evolutionary runs were divided into two sets: the successful runs, in which the highest fitness score was above 5; and the failed runs, in which the highest fitness score was below 3. We focused on the amount of time the robots spent within the sensing range of each other, which is directly related to the degree of cooperation between them, since the aerial robot cannot assist the ground robot if it is permanently outside its sensing range. Table 1 shows that in the successful runs, the robots stay within the sensing range of each other most of the time, across all task variants. In the failed runs, however, the scenario is different: the degree of cooperation between the robots is significantly lower ($p < 0.001$). Why do some of the evolutionary runs fail to evolve cooperation? We investigate two possible causes:

Loss of fitness gradients: Having agents that need to learn substantially different sets of skills can result in loss of fit-

ness gradient in the populations (Wiegand, 2003). If one agent does not possess the skills necessary to make any impact in the performance of the team, fitness diversity might be lost. Consequently, the individuals cannot be adequately ranked, and evolution starts to drift. We investigated the loss of gradients by measuring the relative standard deviation (RSD) of fitness scores in each population, at every generation. A value close to zero means that fitness diversity is almost absent, indicating loss of fitness gradients. Higher values, on the other hand, indicate a rich fitness diversity.

Convergence to mediocre stable states: If the two populations fail to sustain a mutual development of skills, they can converge to a mediocre stable state (Panait, 2010): the individuals of one population can become over-adapted to the poor behaviours found in the other population. To analyse convergence to stable states, we resorted to a measure of team behaviour exploration, as done in previous works (Gomes et al., 2014). Team behaviour exploration is given by the mean behavioural difference between every two individuals evolved in a given evolutionary run, thus representing the dispersion of the individuals over the team behaviour space. Low values indicate that the individuals converged to a narrow region of the behaviour space, while higher values suggest that the behaviour space was reasonably explored.

The dispersion of fitness scores, shown in Table 2, suggest that the loss of fitness gradients is *not* the main cause for evolutionary failure. The average dispersion (RSD) is relatively high in both successful and failed runs, across all task variants and in both populations, meaning that fitness diversity is maintained. Regarding the behaviour space exploration, Table 3 shows that there is significantly less exploration in the failed runs than in the successful runs ($p < 0.001$). The relatively low degree of behaviour space exploration in the failed runs suggest that the main cause of evolutionary failure was convergence to mediocre stable states.

Table 2: Average dispersion of fitness scores inside each population, at every generation. Dispersion is given by the relative standard deviation (RSD).

	Successful runs		Failed runs	
Variant	Ground	Aerial	Ground	Aerial
Fix-Tog	0.23	0.47	NA	NA
Fix-Sep	1.03	0.52	1.02	0.42
Var-Tog	0.47	0.70	0.96	0.52
Var-Sep	1.06	0.49	1.15	0.55

Table 3: Team behaviour exploration, given by the mean behavioural difference between every two individuals.

Variant	Successful runs	Failed runs
Fix-Tog	$0.396_{\pm.07}$	NA
Fix-Sep	$0.600_{\pm.06}$	$0.305_{\pm.04}$
Var-Tog	$0.619_{\pm.07}$	$0.318_{\pm.07}$
Var-Sep	$0.526_{\pm.07}$	$0.272_{\pm.05}$

Improving Cooperative Coevolution

We evaluated three methods (described in the *Methods* section) to try to improve coevolution's effectiveness. The quality of solutions achieved with each method, for the three task variants where the base CCEA failed, are depicted in Figure 3. We also analysed the exploration of the behaviour space, using the measure described in the previous section. The results are shown in Table 4. Every evolutionary treatment was repeated in 30 independent evolutionary runs.

Incremental evolution (*Inc*) Incremental evolution was on average the highest performing approach, and significantly improved over the basic CCEA in all task variants ($p < 0.05$). The results in Figure 3 (left) show that incremental evolution tends to reach high fitness scores in fewer generations than the other methods. Incremental evolution initially rewards the robots for staying within sensing range of one another. As the robots are essentially forced to cooperate before reaching the final stage, evolution is less likely to get stuck in a mediocre stable state where the robots do not cooperate when collecting the items. Nevertheless, incremental evolution still fails in many evolutionary runs. The effectiveness of incremental evolution depends on the defined sub-goals, and as such it is possible that a substantially different configuration could yield better results.

Incremental evolution is, however, associated with well known limitations (Doncieux and Mouret, 2014), as a great deal of domain knowledge is required to design effective evolutionary stages. We knew beforehand that maintaining an altitude close to 250 cm for the aerial robot and having the robots close to one another were desirable properties. If, however, we had little knowledge about the solution of the task, it would have been hard to shape the evolutionary process, with a risk of biasing it towards suboptimal solutions. Besides knowing which subgoals the evolutionary process must achieve, the experimenter must also specify what the order of the subgoals should be, and how to determine if a subgoal has been accomplished.

Non-cooperative incremental evolution (*NInc*) Non-cooperative incremental evolution displayed a relatively poor performance across all variants, and it was always significantly inferior to incremental evolution ($p < 0.05$), despite the similarities between these two methods. The rationale of this approach was to bootstrap the aerial robot before starting cooperative coevolution. With this approach, however, the aerial robot evolves a behaviour that is over-adapted to a static ground robot. When cooperative coevolution starts, and the ground robot starts moving, the aerial robot is not prepared to deal with it, and its previously learned behaviours tend to fail. This result supports previous works (Dorigo et al., 2013) that argue that when developing cooperative systems, the cooperation between the agents must be taken into account from the very beginning.

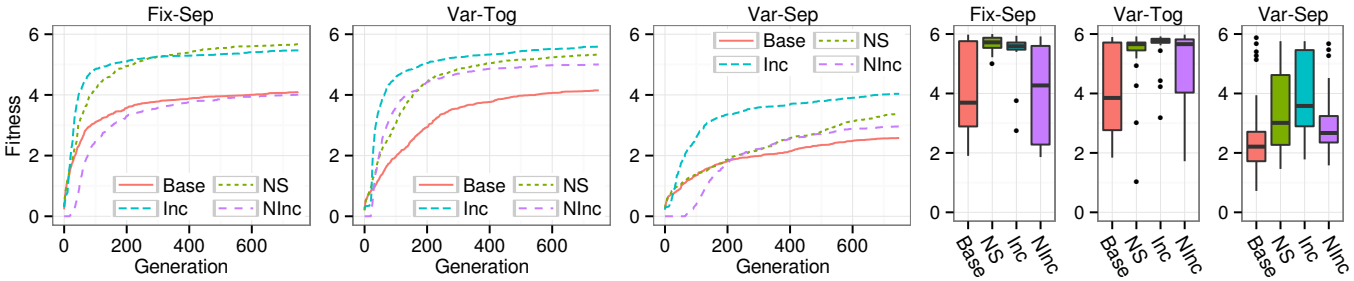

Figure 3: Fitness scores achieved in each task variant, with the base CCEA (*Base*), incremental evolution (*Inc*), non-cooperative incremental evolution (*NInc*), and novelty-driven coevolution (*NS*). Fitness corresponds to the number of items collected (F_i). Left: average highest fitness scores achieved at each generation. Right: boxplots of the highest scores achieved in each run.

Table 4: Average team behaviour exploration for the different evolutionary treatments and task variants.

Var.	Base	NS	Inc	NInc
Fix-Tog	0.40±.07	0.71±.03	NA	NA
Fix-Sep	0.44±.14	0.68±.05	0.54±.08	0.43±.14
Var-Tog	0.46±.15	0.68±.07	0.50±.07	0.47±.11
Var-Sep	0.31±.11	0.49±.07	0.43±.13	0.35±.09

Novelty-driven coevolution (*NS*) Novelty-driven coevolution avoids convergence to mediocre stable states in a completely different way than incremental evolution: it rewards the exploration of the behaviour space, without introducing biases towards specific behaviours. The results in Table 4 highlight this difference: novelty-driven coevolution exhibited a significantly higher degree of behaviour exploration than all other approaches in all task variants ($p < 0.05$).

Regarding the fitness scores achieved, *NS* significantly outperformed the basic CCEA in all variants ($p < 0.05$). The performance of *NS* is, however, inferior to incremental evolution in *Var-Tog* and *Var-Sep*. Novelty-driven coevolution was not always effective, as evidenced by the number of failed runs in the *Var-Tog* and *Var-Sep* variants (see Figure 3, right). In this item collection task, the team behaviour space can only be adequately explored if the two robots cooperate. If, for instance, the aerial robot does nothing at all, or simply flies away, the diversity of team behaviours that can be achieved is significantly compromised. Novelty-driven coevolution can get trapped exploring only a small region of the team behaviour space, and thus fails to discover high-quality solutions.

Conclusion

This work addressed the challenge of coevolving behaviours for cooperative multirobot systems where the robots have significant morphological differences. Our experiments relied on a task where a highly capable aerial robot must assist a relatively simple ground robot in collecting items. We used multiple task variants in which we varied the number of skills the aerial robot had to develop before being able to cooperate with the ground robot.

In the simplest task variant, where the aerial robot can immediately start cooperating with the ground robot, coevolution consistently found (near-)optimal solutions. Despite the disparity between the sensor-effector capabilities of the robots, and thus different complexity of their controllers, coevolution was able to sustain a mutual development of skills. In the task variants where the aerial robot had to develop additional skills before being able to cooperate, however, coevolution frequently failed. Our results showed that coevolution often converges to stable states where there is little or no cooperation between the robots.

We tried to improve coevolution's effectiveness using techniques described in previous works: incremental evolution; a non-cooperative version of incremental evolution, bootstrapping the more complex agent; and novelty-driven coevolution. Incremental evolution and novelty-driven coevolution significantly outperformed the basic CCEA, but they were still unable to consistently solve all task variants. Some evolutionary runs failed to evolve cooperation, especially when the aerial robot had to learn multiple skills before being able to cooperate. Novelty search was slightly less effective than incremental evolution, but it still managed to reach high-quality solutions, and relied less on the experimenter's knowledge. Our results also revealed that bootstrapping the more complex agent before the coevolutionary process is not effective, as it does not take in consideration the influence the agents have on the behaviours of one another.

Our experiments suggest that in order to coevolve agents with very different capabilities and sets of skills, the experimenter might need to modify the coevolutionary process to avoid mediocre stable states, and to encourage the evolution of the necessary skills for effective cooperation. This can be accomplished by incorporating domain knowledge into the process (incremental evolution) or by adopting a more open-ended evolutionary approach (novelty search). Despite this limitation, we showed that cooperative coevolution can yield good solutions, and can be considered a viable approach to evolve behaviours for morphologically heterogeneous mul-

tirobot systems. In future work, we are investigating if these results generalise to other tasks and robot types, and how cooperative coevolution might perform when different populations use different evolutionary algorithms.

Acknowledgements This research is supported by Fundação para a Ciência e Tecnologia (FCT), with grant SFRH/BD/89095/2012 and projects UID/EEA/50008/2013, UID/Multi/04046/2013, and EXPL/EEI-AUT/0329/2013.

References

Blumenthal, H. J. and Parker, G. B. (2004). Co-evolving team capture strategies for dissimilar robots. In *AAAI Artificial Multi-agent Learning Symposium*, volume 2. AAAI Press.

Candea, C., Hu, H., Iocchi, L., Nardi, D., and Piaggio, M. (2001). Coordination in multi-agent RoboCup teams. *Robotics and Autonomous Systems*, 36(2):67–86.

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.

Dorigo, M., Floreano, D., Gambardella, L., Mondada, F., et al. (2013). Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4):60–71.

Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T. H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., et al. (2004). Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2–3):223–245.

Duan, H. B. and Liu, S. Q. (2010). Unmanned air/ground vehicles heterogeneous cooperative techniques: Current status and prospects. *Science China Technological Sciences*, 53(5):1349–1355.

Ducatelle, F., Di Caro, G., Pinciroli, C., and Gambardella, L. (2011). Self-organized cooperation between robotic swarms. *Swarm Intelligence*, 5(2):73–96.

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

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

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

Grabowski, R., Navarro-Serment, L. E., Paredis, C. J., and Khosla, P. K. (2000). Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8(3):293–308.

Howard, A., Parker, L. E., and Sukhatme, G. S. (2006). Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *International Journal of Robotics Research*, 25(5-6):431–447.

Lacroix, S. and Le Besnerais, G. (2011). Issues in cooperative air/ground robotic systems. In *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 421–432. Springer.

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

Mathews, N., Christensen, A. L., O'Grady, R., and Dorigo, M. (2010). Cooperation in a heterogeneous robot swarm through spatially targeted communication. In *Swarm Intelligence*, volume 6234 of *LNCS*, pages 400–407. Springer.

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. (2012). Evolving behavioral specialization in robot teams to solve a collective construction task. *Swarm and Evolutionary Computation*, 2:25–38.

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

Parker, L., Kannan, B., Tang, F., and Bailey, M. (2004). Tightly-coupled navigation assistance in heterogeneous multi-robot teams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1016–1022. IEEE Press.

Potter, M. A. and Jong, K. A. D. (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.

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

Uchibe, E. and Asada, M. (2006). Incremental coevolution with competitive and cooperative tasks in a multirobot environment. *Proceedings of the IEEE*, 94(7):1412–1424.

Uchibe, E., Nakamura, M., and Asada, M. (1998). Co-evolution for cooperative behavior acquisition in a multiple mobile robot environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 425–430. IEEE Press.

Wiegand, R. P. (2003). *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University.

Yang, J., Liu, Y., Wu, Z., and Yao, M. (2012). The evolution of cooperative behaviours in physically heterogeneous multi-robot systems. *Int. Journal of Advanced Robotic Systems*, 9(253).

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.

Jorge Gomes, Pedro Mariano, Anders Lyhne Christensen (2015) Cooperative Coevolution of Morphologically Heterogeneous Robots. Proceedings of the European Conference on Artificial Life 2015, pp. 312-319