Swarm Intell (2013) 7:115–144
DOI 10.1007/s11721-013-0081-z

Evolution of swarm robotics systems with novelty search

Jorge Gomes · Paulo Urbano · Anders Lyhne Christensen

Received: 31 October 2012 / Accepted: 5 April 2013 / Published online: 14 May 2013

Abstract Novelty search is a recent artificial evolution technique that challenges traditional evolutionary approaches. In novelty search, solutions are rewarded based on their novelty, rather than their quality with respect to a predefined objective. The lack of a predefined objective precludes premature convergence caused by a deceptive fitness function. In this paper, we apply novelty search combined with NEAT to the evolution of neural controllers for homogeneous swarms of robots. Our empirical study is conducted in simulation, and we use a common swarm robotics task—aggregation, and a more challenging task—sharing of an energy recharging station. Our results show that novelty search is unaffected by deception, is notably effective in bootstrapping evolution, can find solutions with lower complexity than fitness-based evolution, and can find a broad diversity of solutions for the same task. Even in non-deceptive setups, novelty search achieves solution qualities similar to those obtained in traditional fitness-based evolution. Our study also encompasses variants of novelty search that work in concert with fitness-based evolution to combine the exploratory character of novelty search with the exploitatory character of objective-based evolution. We show that these variants can further improve the performance of novelty search. Overall, our study shows that novelty search is a promising alternative for the evolution of controllers for robotic swarms.

Electronic supplementary material The online version of this article (doi:10.1007/s11721-013-0081-z) contains supplementary material, which is available to authorised users.

J. Gomes (✉)
LabMAg, FCUL & Instituto de Telecomunicações, Lisbon, Portugal
e-mail: jgomes@di.fc.ul.pt

P. Urbano
LabMAg, FCUL, Lisbon, Portugal
e-mail: pub@di.fc.ul.pt

A.L. Christensen
Instituto Universitário de Lisboa (ISCTE-IUL) & Instituto de Telecomunicações, Lisbon, Portugal
e-mail: anders.christensen@iscte.pt

Keywords Evolutionary robotics · Neuroevolution · Swarm robotics · Novelty search · NEAT · Behavioural diversity · Deception

1 Introduction

Motivations for the use of evolutionary techniques to design control systems for robots are numerous (Harvey et al. 1993; Nelson et al. 2009). In the swarm robotics domain in particular, the complexity stemming from the intricate dynamics required to produce self-organised behaviour complicates the hand-design of control systems. Artificial evolution, on the contrary, has been shown capable of exploiting the intricate dynamics and synthesise self-organised behaviours (see for example Trianni 2008; Sperati et al. 2008). However, evolutionary robotics techniques have only been proven effective in relatively simple tasks (Sprong 2011; Doncieux et al. 2011; Brambilla et al. 2013). The lack of studies that report successful evolution of behaviours for complex tasks can be ascribed to the difficulties in configuring the evolutionary process such that adequate solutions are synthesised within a reasonable amount of time (Doncieux et al. 2011).

The most common approach in artificial evolution, and evolutionary robotics in particular, is to guide the evolutionary process towards a fixed objective (Nelson et al. 2009) (henceforth referred to as *fitness-based evolution*). The experimenter defines a fitness function that estimates the quality of candidate solutions with respect to a given task, and this fitness function is used to score the individuals in the population. While search based on the task objective may intuitively seem reasonable, it is associated with a number of issues of which *deception* is one of the most prominent (Whitley 1991; Jones and Forrest 1995). Deception is a challenging issue in evolutionary computation, which occurs when the fitness function misguides the evolutionary process (Lehman and Stanley 2011a), potentially causing evolution to converge to local optima. As the complexity of a task or a system increases, it becomes more difficult to craft an appropriate fitness function, and fitness-based evolution becomes more vulnerable to deception (Zaera et al. 1996).

Novelty search (Lehman and Stanley 2011a) is a distinctive evolutionary approach which rewards solutions based solely on their behavioural novelty. In fitness-based evolution, the objective is typically static, and the evaluations of individuals are independent of one another. In novelty search, on the other hand, individuals are evaluated by a dynamic measure that scores candidate solutions based on how different they are from solutions evaluated so far, with respect to their behaviour. Due to the absence of a static objective, novelty search is unaffected by premature convergence. Mouret and Doncieux (2009) showed that the novelty-based paradigm can also be effective in bootstrapping evolution. Novelty search has proven capable of finding a broad diversity of solutions to a given problem (Lehman and Stanley 2011b) and solutions with lower neural network complexity than fitness-based evolution (Lehman and Stanley 2011a). Novelty search has been successfully applied to many domains, including non-collective evolutionary robotics (for examples, see Krcah 2010; Lehman and Stanley 2011a; Mouret 2011).

In this paper, we propose and study the application of novelty search to the evolution of neural controllers for swarm robotics systems. Our motivation is the high level of complexity associated with swarm robotics, which stems from the intricate dynamics between many interacting units. The high level of complexity has the tendency to generate deceptive fitness landscapes (Whitley 1991), and novelty search has been shown to be unaffected by deception (Lehman and Stanley 2011a). To evaluate novelty search in the swarm robotics domain, we conduct several experiments. We use two different tasks in our study: (i) an aggregation task, and (ii) a resource sharing task. The former is a task commonly used in the field of evolutionary swarm robotics. The latter is a more challenging task in which the swarm must coordinate to ensure that each member has periodic and exclusive access to a charging station. In all experiments, we establish comparisons between novelty search and traditional fitness-based evolution. One of the key components in novelty search is the *novelty measure* that quantifies the novelty of each solution. It is based on a *behaviour characterisation* (usually a real-valued vector) that corresponds to an approximate representation of the individual's actual behaviour. The behaviour characterisation is typically domain-dependent and task-dependent. We study different approaches to the definition of behaviour characterisations for swarm robotics tasks. Our characterisations capture the macroscopic swarm-level behaviour and thus are independent of the swarm size. In the aggregation task, we evaluate characterisations composed of behavioural features sampled at regular intervals. In the resource sharing task, we go on to show that simple characterisations that summarise an entire simulation are, in fact, sufficient for novelty search to find good solutions.

One issue that can arise in novelty search is that a significant part of the effort may be spent exploring novel but unfruitful regions of the behaviour space (Lehman and Stanley 2010a; Cuccu and Gomez 2011). A number of methods that combine the exploratory nature of novelty search with the exploitatory nature of fitness-based evolution have been proposed to address this issue (Lehman and Stanley 2010a; Cuccu and Gomez 2011; Mouret 2011; Gomes et al. 2012). We explore the potential of two such methods, namely *progressive minimal criteria novelty search* (Gomes et al. 2012) and *linear scalarisation* of novelty and fitness objectives (Cuccu and Gomez 2011).

The paper is organised as follows: in Sect. 2, we discuss related work, present the current challenges in evolutionary robotics, and introduce the novelty search algorithm. In Sect. 3, we use novelty search to evolve aggregation behaviours. We experiment with three behaviour characterisations, and study how each one affects the behavioural diversity and the performance of novelty search. In Sect. 4, we experiment with a more challenging resource sharing task. We find that some behaviour characterisations open the search space too much and thereby reduce the effectiveness of novelty search. In Sect. 5, we show how the problem of vast behaviour spaces can be mitigated by combining novelty search with fitness-based evolution. We conclude in Sect. 6 with a summary of the contributions of the paper and with a discussion of ongoing work.

2 Related work

In this section, we first discuss swarm robotics and evolutionary robotics, and the main challenges associated with these fields. We then present novelty search and how it can overcome some of these challenges. We go on to review recently proposed variants of novelty search. We conclude the section with a description of NEAT, the neuroevolution method used in our experiments.

2.1 Swarm robotics

The field of swarm robotics, as well as the more general field of swarm intelligence (Bonabeau et al. 1999), take inspiration from the observation of social insects. In a swarm intelligence system, be it natural such as an ant colony, or artificial such as a large-scale decentralised multirobot system, relatively simple units rely on self-organisation to display collectively intelligent behaviour. As such, swarm robotics is an auspicious approach to the decentralised coordination of large numbers of robots (Şahin 2005). An extensive survey of the modelling of swarm robotics systems and the problems that have been addressed can be found in Brambilla et al. (2013), Bayindir and Şahin (2007). Self-organisation in multirobot systems has, however, proven difficult to design by hand. Manually designing the control for the individual units of a swarm requires the decomposition of the macroscopic swarm behaviour into microscopic behavioural rules (Trianni 2008). Such decomposition includes discovering the relevant interactions between the individual robots, and between the robots and the environment, which will ultimately lead to the emergence of global self-organised behaviour. Unfortunately, there is no general method for decomposing a desired global behaviour into the rules that govern each individual. System designers therefore typically take inspiration from biological swarm systems or rely on manual trial and error.

2.2 Evolutionary robotics

Evolutionary robotics is a field concerned with the application of evolutionary computation to the synthesis of robotic systems. Evolutionary robotics is an alternative for the design of control for swarm robotics systems, because the application of evolutionary computation eliminates the need for manual decomposition of the desired macroscopic behaviour. Artificial evolution essentially performs an iterative trial and error process in which candidate solutions are evaluated according to their swarm-level behaviour. Macroscopic performance evaluation is thus used to guide the evolutionary process towards the objective. Several swarm robotics tasks have been solved with evolutionary approaches, such as coordinated motion (Baldassarre et al. 2007), foraging (Liu et al. 2007), aggregation (Trianni et al. 2003), hole avoidance (Trianni et al. 2006), aerial vehicles communication (Hauert et al. 2009), categorisation (Ampatzis et al. 2008), group transport (Gross and Dorigo 2008), and social learning (Pini and Tuci 2008).

Traditional evolutionary approaches are, however, prone to suffer from a number of issues (Doncieux et al. 2011). Deception (Whitley 1991; Jones and Forrest 1995) is a challenging issue in evolutionary computation, because it can cause the evolutionary process to converge prematurely to local optima. Deception occurs when the fitness function creates a *deceiving* fitness gradient. This typically happens when the fitness function fails to adequately reward the intermediated steps that are needed to achieve the global optimum. A related issue that can arise when applying evolutionary computation to complex tasks is the bootstrap problem (Gomez and Mikkulainen 1997; Mouret and Doncieux 2009). This problem occurs when the task is too demanding to exert significant selective pressure on the population during the early stages of evolution, as all of the individuals perform equally poorly. As a consequence, there is no fitness gradient and the evolutionary process starts to drift in an uninteresting region of the solution space.

One way to circumvent deception is through the use of techniques that maintain genotypic diversity in the population, such as fitness sharing (Goldberg and Richardson 1987), promotion of diversity based on the fitness score of the solutions (Hu et al. 2005; Hutter and Legg 2006), intermingling individuals of different genetic ages (Hornby 2006; Castelli et al. 2011), and minimisation of the age of the genotypes (Schmidt and Lipson 2011). However, the problem may ultimately be in the fitness function itself, not in the particular search algorithm. If the fitness function is actively misguiding the search, the evolutionary process may still fail, regardless of the amount of genotypic diversity present in the population.

A distinct approach to address the issue of deception is through the use of coevolution. In competitive coevolution, individual fitness is evaluated through competition with other individuals in the population, rather than through an absolute fitness measure. Ideally, this creates an *arms race* that leads to increasingly better solutions. This approach has been applied with success in some domains (e.g. Chellapilla and Fogel 1999), but it is also associated with a number of issues that stem from the potentially counterproductive dynamics between the multiple co-evolving species, such as convergence to mediocre stable states (Watson and Pollack 2001). Other techniques to overcome deception and to bootstrap evolution rely on the decomposition of the objective into multiple sub-goals that each are easier to attain. These techniques include incremental evolution (Gomez and Mikkulainen 1997), fitness shaping (Uchibe et al. 2002), and multiobjectivisation (Deb 2001; Knowles et al. 2001). A common drawback of these approaches is that task decomposition may not always be possible, and when it is, a significant amount of a priori knowledge about the task is required to devise appropriate sub-tasks.

2.3 Novelty search

While the methods discussed above for mitigating deception might help the evolutionary process to avoid getting stuck in local optima, they leave the underlying problem untreated, namely that the fitness function itself might be misdirecting the search. With this issue in mind, a new evolutionary approach was recently proposed—novelty search. In novelty search, the evolutionary process is based on the promotion of phenotypic (i.e., behavioural) diversity and innovation, contrasting with more common techniques that strive to maintain genotypic diversity. Lehman and Stanley (2011a) used two deceptive robotics tasks to show how novelty search was able to find good solutions faster and more consistently than fitness-based evolution. Even though the objective was not directly pursued in any of their experiments, a solution was found more consistently through the exploration of the behaviour space. Successful applications of novelty search include the evolution of adaptive neural networks (Soltoggio and Jones 2009); genetic programming (Lehman and Stanley 2010b); evolution strategies (Cuccu et al. 2011); body–brain coevolution (Krcah 2010); biped robot control (Lehman and Stanley 2011a); and robot navigation in deceptive mazes (Lehman and Stanley 2008; Mouret 2011).

Implementing novelty search requires little change to any evolutionary algorithm aside from replacing the fitness function with a domain-dependent novelty metric. This metric quantifies how different an individual is from the other, previously evaluated individuals with respect to behaviour. Previously seen behaviours are stored in an archive. The archive is initially empty, and new behaviours are added to it if they are significantly different from the ones already there, i.e., if their novelty score is above some threshold.

A novelty metric characterises how far an individual is from other individuals in the behaviour space. This metric depends on the sparseness at a given point in the behaviour space. A simple measure of sparseness at a point is the average distance to the k-nearest neighbours of that point, where k is a fixed parameter empirically determined. The sparseness ρ at point x is given by

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

where μ_i is the ith-nearest neighbour of x with respect to the distance metric bdist. Note that the computational cost of the nearest-neighbours calculation increases linearly with the size of the population and the size of the archive. However, it is possible to limit the size of the archive (Lehman and Stanley 2011a) and to use data structures such as KD-trees to reduce this cost. The function bdist is a measure of behavioural difference between two individuals in the search space. Candidates from sparse regions of the behaviour space thus tend to receive higher novelty scores, which results in an evolutionary process that strives to uniformly explore the behaviour space. Note that the novelty metric promotes behavioural diversity within the population at all times and therefore helps to avoid convergence to a single solution, which is common in fitness-based evolution.

The behaviour of each individual is typically characterised by a vector of real numbers. The behavioural distance bdist is then given by the distance between the corresponding characterisation vectors (the Euclidean distance is commonly used). The experimenter should design the behaviour characterisation so that it captures behaviour aspects that are considered relevant to the problem or task. For example, in a maze navigation task (Lehman and Stanley 2011a) the behaviour characterisation was the trajectory of the robot through the maze. The design of the characterisation has direct implications on the effectiveness of novelty search. An excessively detailed characterisation can open the search space too much, and might cause the evolution to focus on regions of the behaviour space that are irrelevant for the task for which a solution is sought. On the other hand, an incomplete or inadequate characterisation can originate counterproductive *conflation*. Conflation occurs because the mapping between observable behaviours and behaviour characterisations is typically not injective. As such, notably different behaviours can have similar behaviour characterisations, which can potentially hinder the evolution of novel solutions (Kistemaker and Whiteson 2011).

Designing the behaviour characterisation often requires knowledge about which behaviour features are relevant for solving a task. However, unlike fitness shaping techniques, it is not necessary to understand exactly how these behaviour features affect fitness, or in which order the features must be evolved. Novelty search does not require a fitness gradient to guide evolution, which makes the approach applicable to some classes of problems that are difficult to solve using traditional fitness-based evolution (Kistemaker and Whiteson 2011).

2.4 Novelty search variants

Several extensions have been proposed to overcome the limitation of novelty search with respect to guiding evolution towards good solutions in vast behaviour spaces. These extensions are based on the combination of the exploratory character of novelty search with the exploitatory character of fitness-based evolution.

Lehman and Stanley (2010a) proposed *minimal criteria novelty search* (MCNS), an extension of novelty search where individuals must meet some domain-dependent minimal criteria to be selected for reproduction. In Lehman and Stanley (2010a), the authors applied MCNS in two maze navigation tasks and demonstrated that MCNS evolved solutions more consistently than both novelty search and fitness-based evolution. Similar results were reported in Kirkpatrick (2012), using competitive coevolution and a different task. However, MCNS suffers from a number of drawbacks (Lehman and Stanley 2010a). First, the choice of minimal criteria in a particular domain requires careful consideration and domain knowledge, since it adds significant restrictions to the search space. Constraining the search space too much can hinder the evolution of some types of solutions. Second, if no individuals are found that meet the minimal criteria, search is effectively random. In situations where an initial randomly generated population is unlikely to contain such individuals, it may therefore be necessary to seed MCNS with a genome specifically evolved to meet the criteria. Finally, if the minimal criteria are too stringent, it might be difficult to mutate apt individuals without violating the criteria, and thus many evaluations may be wasted.

In recent work (Gomes et al. 2012), we proposed an extension named *progressive minimal criteria novelty search* (PMCNS), which overcomes the drawbacks of MCNS. In PMCNS, the respective benefits of novelty search and fitness-based evolution are combined by letting novelty search freely explore new regions of the behaviour space, as long as solutions meet a progressively stricter fitness criterion. PMCNS was found to outperform several other evolutionary algorithms, and to evolve higher scoring individuals while still maintaining behavioural diversity.

Cuccu and Gomez (2011) proposed an alternative approach for combining novelty and fitness, where the score of each individual is based on a linear scalarisation of the novelty score and the fitness score (henceforth referred to as *linear scalarisation*). They applied the approach to a deceptive box-pushing task, and found that linear scalarisation outperformed both novelty search and fitness-based evolution. Mouret (2011) proposed novelty-based multiobjectivisation, which is a Pareto-based multiobjective evolutionary algorithm. A novelty objective is added to the task objective in a multiobjective optimisation. The technique was applied to a deceptive maze navigation problem. Compared with pure novelty search, the use of multiobjectivisation only led to marginally better results. Other techniques for sustaining behavioural diversity in evolutionary robotics are reviewed in Mouret and Doncieux (2012).

2.5 NEAT

In our experiments, the controllers of the robots are time recurrent neural networks evolved by NEAT (short for NeuroEvolution of Augmenting Topologies) (Stanley and Miikkulainen 2002). NEAT is a widely used neuroevolution approach, and one of the most successful approaches developed to date. NEAT simultaneously optimises the weighting parameters and the structure of artificial neural networks. It begins the evolution with a population of small, simple networks and complexifies the network topology into diverse species over generations. This leads to the evolution of increasingly sophisticated behaviour. A key feature in NEAT is its distinctive approach to maintain a diversity of growing structures simultaneously. Unique historical markings are assigned to each new structural component. During crossover, genes with the same historical markings are aligned, producing valid offspring efficiently, without the need of complex topological comparisons. NEAT uses speciation and fitness sharing to protect new structural innovations. This reduces competition between networks with distinct topologies, providing time for the weights of new structures to be optimised. Networks are assigned to species based on the extent to which they share historical markings. Complexification is thus supported by both historical markings and speciation, allowing NEAT to establish high-level features early in evolution and then elaborate on them as the evolutionary process progresses. In effect, NEAT searches for a compact, appropriate network topology by incrementally complexifying existing structures.

It is important to note that both novelty search and NEAT strive to maintain diversity, but at different levels: whereas NEAT maintains *genotypic* diversity, novelty search maintains *phenotypic* diversity. Novelty search and NEAT thus complement one another and their combined use has a number of advantages. In particular, the complexification mechanism of NEAT can introduce order in the exploration done by novelty search, with less complex behaviours being explored before progressing to more complex ones (Lehman and Stanley 2011a).

3 Evolution of aggregation behaviours with novelty search

In this section, we apply novelty search to the problem of swarm aggregation—a commonly studied task in swarm robotics. In the aggregation task, a dispersed robot swarm must form a single cluster. We conduct three sets of experiments with novelty search, each with a distinct behaviour characterisation. We compare the performance of novelty search to the performance of traditional fitness-based evolution. We also include results from evolutionary runs with NEAT in which random fitness scores are assigned to individuals. Random evolution serves as a baseline for performance comparisons.

Aggregation is a task of fundamental importance in many biological systems. It is the basis for the emergence of various forms of cooperation, and can be considered a prerequisite for the accomplishment of many collective tasks (Trianni et al. 2003). Several works describe the evolution of aggregation behaviours for swarms of robots. Commonly, the parameters of neural networks with fixed topologies are optimised by fitness-based evolutionary algorithms. Baldassarre et al. (2003) successfully evolved controllers for a swarm of robots to aggregate and move towards a light source in a clustered formation. Three classes of behaviours were evolved, but each evolutionary run always converged to only one of the classes. Trianni et al. (2003) studied the evolution of a swarm of simple robots to perform aggregation in a square arena. Two different behaviours were evolved: *static clustering* which leads to the formation of compact and stable clusters, and *dynamic clustering* which leads to loose but moving clusters. Bahgeçi and Şahin (2005) used a similar experimental setup as Trianni et al. (2003), and studied how some of the parameters of the evolutionary algorithm affected the performance and the scalability of the evolved behaviours.

In the previous studies discussed above, the robots used directional sound sensing. Directional sensing allowed the robots to follow gradients towards groups of other robots emitting sound signals. The use of sound and directional microphones makes the aggregation task sufficiently easy for controllers based on reactive neural networks without any hidden neurons to solve the task (Trianni et al. 2003; Baldassarre et al. 2003; Bahgeçi and Şahin 2005). In our work, the aggregation task is more challenging: the robots do not use sound, the range of the sensors is significantly lower than in previous studies, and the arena is larger. These modifications increase the difficulty of the task and may require radically different strategies for aggregation (Soysal et al. 2007), since robots can only sense one another when they are close.

3.1 Experimental setup

Our experimental framework is based on the Simbad 3d Robot Simulator (Hugues and Bredeche 2006) for the robotic simulations, and on NEAT4J[1] for the implementation of NEAT. Simbad 3d simulates kinematics and implements simple collision handling. The environment is a 3 m by 3 m square arena bounded by walls. The swarm is homogeneous and composed of seven robots. The robots are modelled based on the e-puck educational robot (Mondada et al. 2009), but do not strictly follow its specification. Each robot is circular with a diameter of 8 cm and can move at speeds of up to 12 cm/s. Regarding sensors, each robot is equipped with eight IR sensors evenly distributed around its chassis for the detection of obstacles (walls or other robots) within a range of 10 cm, and eight sensors dedicated to the detection of other robots within a range of 25 cm. Both types of sensors return the distance of the object that is being sensed, or the maximum value if nothing is sensed. An

[1] NeuroEvolution for Augmenting Topologies for Java—http://neat4j.sourceforge.net.

additional sensor (count sensor) returns the percentage of nearby robots (within a radius of 25 cm), relative to the desired cluster size (the total swarm size in our experiments). The simulated sensors are not based on any specific hardware. Nevertheless, the obstacle sensors could be implemented with active IR sensors, while the robot sensors and the robot count sensor could be implemented with short-range communication (Correll and Martinoli 2007; Gutiérrez et al. 2008). The inputs of the neural network controller are the normalised readings from the sensors mentioned above. The controller has three outputs: one to control the speed of each motor, and one dedicated to completely stopping the robot if its activation is above 0.5.

We evaluate each controller 10 times. In each simulation, we vary the initial position and orientation of each robot. The initial positions are randomised in such a way that the robots are placed at least 50 cm from one another, which ensures that the robots are always reasonably well distributed at the beginning of the simulation. Each simulation lasts for 2500 simulation steps, which corresponds to 250 s of simulated time. The best individual of each generation was post-evaluated in 100 simulations, in order to obtain a more accurate fitness estimate.

3.2 Configuration of the evolutionary algorithms

Fitness-based evolution and random evolution use the default NEAT implementation provided by the NEAT4J library. In random evolution, random fitness scores are assigned to each individual. Novelty search was implemented over NEAT, following the description and parameters in (Lehman and Stanley 2011a). We used a k value of 15 nearest neighbours and individuals are stochastically added to the archive with a probability of 2 %, as suggested in Lehman and Stanley (2010b). The parameters for NEAT were the same in all experiments: recurrent links are allowed, crossover rate—25 %, mutation rate—10 %, population size—200, and each evolutionary process was conducted for 250 generations. The remaining parameters were assigned their default value according to the NEAT4J implementation.

The fitness function is based on the average distance to the centre of mass (also used in Trianni et al. 2003). The fitness F_a of a simulation with T time steps and N robots is defined as

$$F_a = \sum_{i=1}^{N} \frac{1 - \mathrm{dist}(\mathbf{R}_T, \mathbf{r}_{i,T})}{N}, \tag{2}$$

where $\mathbf{R}_T$ is the centre of mass at the end of the simulation, and $\mathbf{r}_{i,T}$ is the position of robot i at the same instant. The distance values are normalised to [0, 1]. The fitness scores obtained in each of the 10 simulations are combined to a single value using the harmonic mean, as advocated in Bahgeçi and Şahin (2005).

The behaviour characterisations we use in novelty search are based on spatial inter-robot relationships, measured at regular intervals of 5 s throughout the simulation. We devised three characterisations:

$\mathbf{b}_{\mathrm{cm}}$: The average distance to centre of mass of the swarm is sampled throughout the simulation. Considering a simulation with N robots and τ temporal samples, the behaviour characterisation $\mathbf{b}_{\mathrm{cm}}$ is given by

$$\mathbf{b}_{\mathrm{cm}} = \frac{1}{N}\left[\sum_{i=1}^{N} \mathrm{dist}(\mathbf{R}_1, \mathbf{r}_{i,1}), \ldots, \sum_{i=1}^{N} \mathrm{dist}(\mathbf{R}_\tau, \mathbf{r}_{i,\tau})\right]. \tag{3}$$

$\mathbf{b}_{\mathrm{cl}}$: The number of robot clusters is sampled at regular intervals throughout the simulation, inspired by the metric used in Bahgeçi and Şahin (2005). Two robots belong to the same cluster if the distance between them is less than their sensor range (25 cm). The behaviour characterisation $\mathbf{b}_{\mathrm{cl}}$ is given by

$$\mathbf{b}_{\mathrm{cl}} = \frac{1}{N}\big[\mathrm{clustersCount}(1), \dots, \mathrm{clustersCount}(\tau)\big]. \tag{4}$$

$\mathbf{b}_{\mathrm{cmcl}}$: The two characterisations $\mathbf{b}_{\mathrm{cm}}$ and $\mathbf{b}_{\mathrm{cl}}$ are concatenated to form a single characterisation. Both $\mathbf{b}_{\mathrm{cm}}$ and $\mathbf{b}_{\mathrm{cl}}$ have the same length and each element of the characterisation vectors ranges from 0 to 1. Thus both components of the behaviour characterisation approximately have the same contribution to the novelty metric. The new characterisation $\mathbf{b}_{\mathrm{cmcl}}$ is given by

$$\mathbf{b}_{\mathrm{cmcl}} = (\mathbf{b}_{\mathrm{cm}}, \mathbf{b}_{\mathrm{cl}}). \tag{5}$$

We computed the spatial inter-robot relationships at every 5 s and the simulation lasted for 250 s. This resulted in behaviour characterisation vectors of length 50 for $\mathbf{b}_{\mathrm{cm}}$ and $\mathbf{b}_{\mathrm{cl}}$, and vectors of length 100 for $\mathbf{b}_{\mathrm{cmcl}}$. As 10 simulations are conducted to evaluate each controller, the corresponding final behaviour characterisation vector is the element-wise mean of the vectors obtained in all 10 simulations. In order to establish a basis for comparison, all the controllers evolved by novelty search and by random evolution were also scored by the fitness function F_a. It is important to note that the fitness scores did not have any influence on the evolutionary process in the novelty search experiments.

3.3 Performance comparison

The fitness trajectories for novelty search, fitness-based evolution, and random evolution, are depicted in Fig. 1 (left). The data points plotted are the averages of the highest fitness score found so far from the start of the evolutionary run and until the current generation.[2] Note that since novelty search does not follow a fitness gradient, the best solutions are not necessarily found in the last generation. As such, it is necessary to save the interesting solutions (for instance, the best one found so far) throughout the evolutionary process. We continued the evolutionary process beyond the 150th generation for all experiments, but there was no significant change in the fitness scores after that point. Although the lowest possible fitness score is ≈ 0.05, the highest fitness of an initial random population is on average ≈ 0.55. The relatively high fitness of an initial random population is explained by the fact that stochastically moving robots tend be significantly closer to one another than in the worst case scenario where robots are located in opposite corners of the arena.

Figure 1 (right) shows box-plots of the highest fitness score for each method in 30 evolutionary runs. The results show that novelty search could consistently find relatively high scoring solutions, with all the behaviour characterisations. There was no significant differences between the highest fitness scores achieved with fitness-based evolution and with novelty search with $\mathbf{b}_{\mathrm{cmcl}}$ (Mann–Whitney U test, p-value < 0.05). With $\mathbf{b}_{\mathrm{cl}}$, the highest fitness scores were significantly lower than the highest scores found by novelty search with $\mathbf{b}_{\mathrm{cmcl}}$

[2] The values that are plotted are fitness scores measured by the fitness function F_a, and not the actual scores used for selection in novelty search or in random evolution. Note that even random evolution has an ascending fitness trajectory because the data points plotted are the highest score achieved so far in the evolutionary process. The trajectory for random evolution can thus increase from time to time when an individual, by chance, scores higher than any previously evaluated individual.

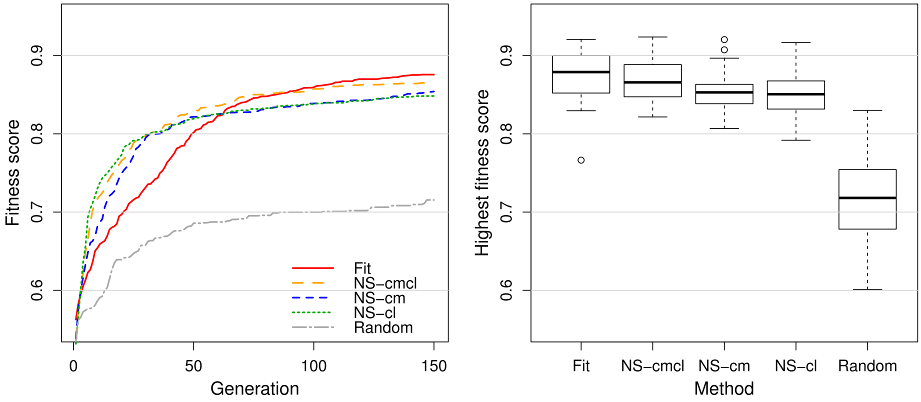

Fig. 1 Highest fitness scores achieved in the aggregation task with fitness-based evolution (*Fit*), random evolution (*Random*), and novelty search with the three behaviour characterisations (*NS-cmcl*, *NS-cm*, *NS-cl*). *Left*: average fitness value of the highest scoring individual found so far at each generation. The values are averaged over 30 independent evolutionary runs for each method. *Right*: box-plots of the highest fitness score found in each evolutionary run, for each method. The whiskers extend to the lowest and the highest data point within 1.5 times the interquartile range. Outliers are indicated by *circles*

and fitness-based evolution. The capacity of novelty search to bootstrap the evolutionary process should be noted. From around generation 5 to generation 40, all novelty search variants achieve fitness scores significantly higher than fitness-based evolution (p-value < 0.01).

Previous studies have shown that novelty search can perform better than fitness-based evolution in deceptive tasks, but fails to match the performance of fitness-based evolution when the task is non-deceptive (Lehman and Stanley 2011a; Mouret 2011). Our results reveal that in the aggregation task, the fitness-function is not deceptive, as fitness-based evolution typically converges to the most effective strategy, achieving high fitness scores (except in a single run, see Fig. 1). Still, novelty search managed to achieve fitness scores that are comparable to the scores achieved in fitness-based evolution.

3.4 Behavioural diversity

The analysis of the explored behaviour space in novelty search and in fitness-based evolution allows for a better understanding of the evolutionary dynamics. Since each behaviour characterisation vector has either 50 or 100 dimensions, we applied a dimensionality reduction method to facilitate visualisation of the explored regions of the behaviour space. We used a Kohonen self-organising map (Kohonen 1990). Kohonen maps are neural networks trained using unsupervised learning to produce a two-dimensional discretisation of the input space of the training samples, while preserving the topological relations. We trained a Kohonen map with the behaviour vectors found both by novelty search and by fitness-based evolution, and then mapped each individual to the region whose vector is more similar to the individual's behaviour vector.

3.4.1 Centre of mass behaviour characterisation

The Kohonen maps corresponding to the explored behaviour space in fitness-based evolution and novelty search with $\mathbf{b}_{\mathrm{cm}}$ can be seen in Fig. 2. The results show that fitness-based evolution focussed the search on only a subset of the behaviour regions. In contrast, novelty

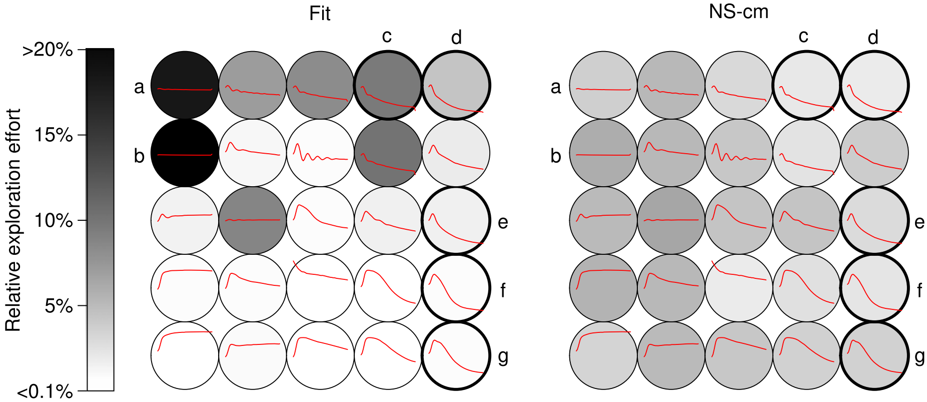

Fig. 2 Kohonen maps representing the explored behaviour space in fitness-based evolution (Fit) and in novelty search with $\mathbf{b}_{\text{cm}}$ (NS-cm). Each *circle* is a neuron corresponding to the vector depicted by the embedded plot (the average distance to the centre of mass over time). Each behaviour vector is mapped to the neuron with the most similar vector. The *darker* the *background* of a neuron is, the more behaviours were mapped to it. The regions associated with higher fitness scores are indicated with a *bold circle* (regions *c* to *g*)

Table 1 The number of individuals mapped to each behaviour region *a*–*g* (see Fig. 2), relative to the total number of mapped individuals

Behaviour region	Fit exploration (%)	NS-cm exploration (%)
a	18.17	3.80
b	22.18	6.39
c	10.27	1.82
d	4.57	1.54
e	1.23	2.75
f	0.15	1.90
g	0.14	3.50

search explored the behaviour space much more uniformly. If we consider the behaviour regions associated with higher fitness scores (regions *c* to *g*), important differences become apparent: fitness-based evolution avoids behaviour regions where the average distance to the centre of mass rises beyond the initial value, such as the regions *f* and *g* (see Table 1). Instead, the evolutionary process is much more focussed on behaviours that lead to a monotonic decrease in the average distance to the centre of mass. This bias is introduced by the fitness function, as it favours a low average distance to centre of mass. Evolving solutions where the average distance rises beyond the initial value might require going against the fitness gradient. As such, these types of solutions are avoided in the fitness-based evolutionary process. Novelty search, on the other hand, is not subject to a static evolutionary pressure, and can therefore explore and discover a wider range of solutions to the task.

The analysis of the behaviour patterns (see plots inside the neurons in Fig. 2) reveals an interesting point: the regions explored by novelty search are characterised by vectors that differ much less than uniformly sampled vectors of length 50 would. This happens because the elements of a given behaviour vector are inherently correlated. Robots cannot, for instance, travel instantly to any location in the environment, and the difference between consecutive samples of the average distance to the centre of mass is therefore limited by the speed of the robots. As such, the reachable behaviour regions constitute only an (often

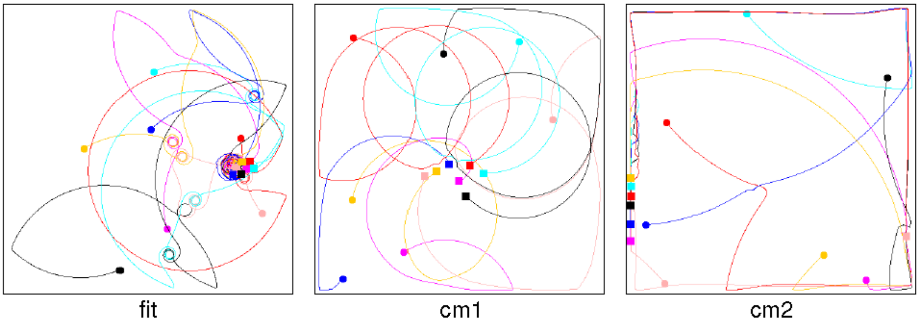

Fig. 3 The typical best solution evolved by fitness-based evolution (fit), and two examples of solutions found by novelty search with $\mathbf{b}_{\mathrm{cm}}$ (cm1 and cm2). Each *line* represents the trajectory of a single robot throughout the simulation. The *circles* depict the initial positions of the robots and the *squares* depict their final positions. Videos of the behaviours are available as online supplemental material

small) subset of the total behaviour space. This explains why novelty search can consistently find successful solutions to the task, even in a high-dimensional behaviour space.

The results also show that fitness-based evolution spends a considerable amount of time in behaviour regions where there are no aggregation dynamics at all (regions *a* and *b*). These regions correspond to the initial best solutions, where robots do not move or move randomly in the arena, in order to maintain an average distance to the centre of mass that is at least as low as the average distance at the start of the simulation. This class of behaviours constitutes a local maximum. Novelty search does not spend as much time in such behaviour regions (see Table 1), as the search moves towards regions with novel behaviours. Since the initial average distance to the centre of mass is situated in the middle of the spectrum, the novel behaviours can be aggregation behaviours as well as dispersion behaviours. Typically, each novelty search evolutionary run explores both types of behaviour simultaneously, maintaining a healthy diversity in the population. This phenomenon can offer an explanation for the relatively good performance of novelty search in the earlier stages of evolution: novelty search does not get stuck in the early local maximum, and as such can explore and discover better solutions faster.

To confirm the behavioural diversity evolved by each method, we resorted to the visual inspection of the best solutions. In fitness-based evolution, the highest scoring individuals tended to follow the same behaviour pattern (Fig. 3, fit):

fit: The robots explore the environment in large circles, and form static clusters when they encounter one another. If a cluster is small, the robots abandon it after a while and recommence the circular exploration.

The fit behaviour pattern was also often found by novelty search. However, novelty search commonly evolved a different type of solutions in which walls are exploited to achieve aggregation. Examples of behaviour patterns that use the walls are described below and depicted in Fig. 3 (cm1 and cm2).

cm1: The robots go towards the walls while avoiding any other robots encountered. When they reach a wall, they follow the wall for a while and then depart with a certain circular trajectory that causes them to pass through the centre of the arena. Eventually, the robots end up forming a loose cluster close to the centre.

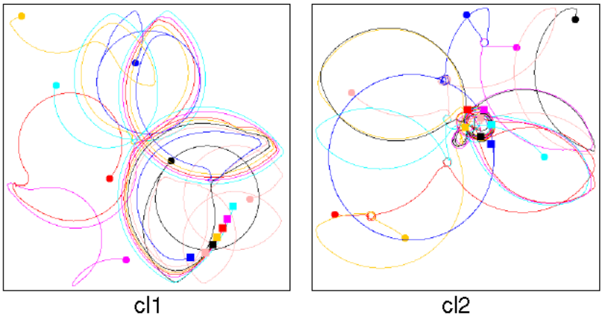

Fig. 4 Examples of distinctive behaviours evolved by novelty search with $\mathbf{b}_{\mathrm{cl}}$. Each *line* represents the trajectory of a single robot throughout the simulation. The *circles* depict the initial positions of the robots and the *squares* depict their final positions. Videos of the behaviours are available as online supplemental material

cm2: The robots move in straight lines until they encounter a wall, and then, depending on the approach angle, they either remain static for a while or start to follow the wall. When two or more robots encounter one another, they stop and form a cluster near the wall.

Visual inspection of the behaviours confirms that fitness-based evolution did not explore some classes of solutions, and in particular, solutions where the robots navigate near the walls. If all the robots initially move towards one of the walls surrounding the arena, they will often end up far apart. Consequently, the centre of mass of the robots will often be close to the centre of the arena, far from the robots, which explains the initial high average distance to the centre of mass (regions *e, f, g* in Fig. 2). Learning to navigate near the walls potentially requires the evolution of many solutions with very low fitness scores. Avoiding navigation close to walls, on the other hand, results in higher fitness scores, because robots will, by chance, be closer to one another, and therefore to the centre of mass. Given the opportunistic nature of fitness-based evolution, the stepping stone of first navigating along walls to later achieve aggregation, is thus unlikely to be found. In fact, we have been unable to find reports in the literature on the evolution of aggregation behaviours that exploit walls. Such behaviours have, however, proven successful in biological systems, such as in self-organised aggregation of cockroaches (Jeanson et al. 2005).

3.4.2 Number of clusters behaviour characterisation

The difference between the fitness scores of the solutions achieved with $\mathbf{b}_{\mathrm{cm}}$ and $\mathbf{b}_{\mathrm{cl}}$ was not significant. Nevertheless, we found that the different characterisations affected the evolved behaviours. Many of the best solutions evolved with $\mathbf{b}_{\mathrm{cl}}$ were similar to the solutions evolved by fitness-based evolution, namely the formation and disbandment of small clusters. However, new behaviour patterns were also found. The distinctive behaviours evolved by novelty search with $\mathbf{b}_{\mathrm{cl}}$ were focussed on the exploitation of inter-robot relations and of flocking in particular. For example, the following two behaviour patterns were identified (see Fig. 4):

cl1: The robots navigate in circles, and when two robots meet, one starts to follow the other, which leads to flocking with circular trajectories. Eventually, a single moving file is formed.

cl2: Similar to cl1, but when a robot cluster reaches a reasonable size, the cluster becomes static.

Overall, novelty search with $\mathbf{b}_{\mathrm{cl}}$ focussed on different classes of behaviours than novelty search with $\mathbf{b}_{\mathrm{cm}}$. One of the main reasons for the difference in the evolved behaviours is conflation. Conflation occurs when individuals with distinct observable behaviours have

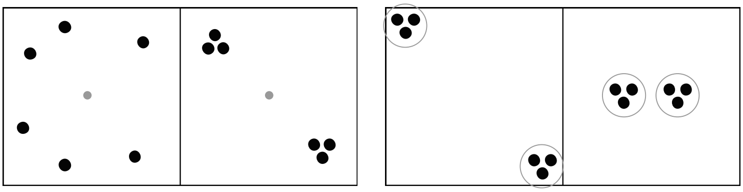

Fig. 5 An illustration of conflation for the centre of mass behaviour characterisation $\mathbf{b}_{\mathrm{cm}}$ (*left*) and for the number of clusters behaviour characterisation $\mathbf{b}_{\mathrm{cl}}$ (*right*). In both cases, the differences would not be reflected in the respective behaviour characterisations

very similar behaviour characterisation vectors (Lehman and Stanley 2011a). The consequence is that an individual with a distinct observable behaviour might not be considered novel by the novelty measure, and may thus disappear from the population. Conflation can represent both an advantage in terms of efficiency because it reduces the size of the search space, and a disadvantage when it inhibits the discovery of successful solutions or important stepping stones.

Two examples of behaviours that can be conflated are shown in Fig. 5. For the $\mathbf{b}_{\mathrm{cm}}$ characterisation, the degree of clustering of the robots is irrelevant; while for the $\mathbf{b}_{\mathrm{cl}}$ characterisation, the distance between robots (and clusters) is irrelevant. The impact of conflation can be seen in the evolved behaviours: with the $\mathbf{b}_{\mathrm{cm}}$ characterisation, there were more behaviours that exploited walls, because navigating near them has a great impact on the novelty measure; while with $\mathbf{b}_{\mathrm{cl}}$, the behaviours focussed on the interactions between the robots and clusters, including following one another and disbanding clusters.

3.4.3 Combined behaviour characterisation

Novelty search with the composed behaviour characterisation $\mathbf{b}_{\mathrm{cmcl}}$ achieved on average marginally higher fitness scores than $\mathbf{b}_{\mathrm{cm}}$ and $\mathbf{b}_{\mathrm{cl}}$ (Mann–Whitney U test, $p\text{-value} < 0.05$), see Fig. 1 (right). The composed behaviour characterisation considers both the average distance to centre of mass and the number of clusters formed. As such, different behaviours are less likely to be conflated, which can potentially lead to better solutions.

It should, however, be noted that $\mathbf{b}_{\mathrm{cm}}$ and $\mathbf{b}_{\mathrm{cl}}$ are closely related with one another. It is not possible, for instance, to have a low average distance to the centre of mass and at the same time a large number of clusters. A large number of clusters implies that the swarm is scattered, and as such, the robots will have a high average distance to the centre of mass. Combining $\mathbf{b}_{\mathrm{cm}}$ and $\mathbf{b}_{\mathrm{cl}}$ reduces conflation to some extent, but since the two characterisations are related to one another, no additional effort is needed to explore the larger behaviour space. This explains why the behaviour space with more dimensions ($\mathbf{b}_{\mathrm{cmcl}}$) did not have a negative impact on the effectiveness of novelty search in this case.

3.5 Neural network complexity

Previous work has shown that an advantage of novelty search, when used together with NEAT, is its ability to evolve solutions with lower genomic complexity than fitness-based evolution (Lehman and Stanley 2011a). To determine if such advantage holds in the aggregation task, we analysed the complexity (sum of the number of neurons and number of connections) of the solutions found by fitness-based evolution and by novelty search. The

Table 2 Comparison of the least complex solutions evolved by fitness-based evolution (Fit) and novelty search with $\mathbf{b}_{\text{cmcl}}$ (NS). The *Complexity* columns lists the network complexity (sum of the number of neurons and the number of connections) of the least complex individual with a fitness score above a certain *Fitness level*. The *Generation* column lists the average generation in which the least complex individual was generated. The values are averages of 30 evolutionary runs for each method

Fitness level	Generation		Complexity	
	Fit	NS	Fit	NS
0.60	9	5	74.20	71.33
0.65	18	7	75.77	71.67
0.70	28	14	77.10	72.40
0.75	42	22	79.10	74.13
0.80	55	41	80.90	77.33
0.85	76	63	83.17	81.70

comparison is established by analysing the average complexity (across the multiple evolutionary runs) of the least complex solution with a fitness score above a certain threshold. The results are shown in Table 2. We only show the results for the $\mathbf{b}_{\text{cmcl}}$ characterisation because the results for the $\mathbf{b}_{\text{cm}}$ and $\mathbf{b}_{\text{cl}}$ characterisations are similar.

The results show that on average, for the same fitness levels, novelty search finds individuals with significantly less complex neural networks (Mann–Whitney U test, p-value $<$ 0.05). The difference is especially pronounced at the lower fitness levels. Note that the networks in the initial populations (without any hidden neurons) have a complexity of 71 (20 neurons and 51 links). The difference in the network complexity can be ascribed to the convergent nature of fitness-based evolution. If the best controllers in the earlier stages of evolution have more complex neural networks, fitness-based evolution starts to converge to such complex structures. As novelty search does not converge, and has tendency to explore simple solutions before moving on to more complex ones (Lehman and Stanley 2011a), it is capable of finding solutions with a lower network complexity.

4 Evolution of resource sharing behaviours with novelty search

In this section, we study the application of novelty search to a more complex task in which a swarm of robots share a single resource. The swarm must coordinate in order to allow each member periodical access to a single battery charging station. In the task, the robots should first find the charging station, and then effectively share the station to ensure the survival of all the robots in the swarm. The charging station can only hold one robot at the time.

The problem of autonomous charging and resource conflict management is widely studied in the literature. Cao et al. (1997) identify resource conflicts as one of the fundamental challenges in the design of cooperative behaviours in multirobot systems. Resource conflicts arise when a single indivisible resource (in our task, the charging station) is requested by multiple robots at the same time. The problem of sharing an energy charging station in particular is addressed in Muñoz Meléndez et al. (2002), Michaud and Robichaud (2002), Kernbach and Kernbach (2011).

4.1 Experimental setup

We use a swarm composed of five homogeneous robots. The robots are identical to the ones used in the aggregation experiments (see Sect. 3.1), except that they are equipped with

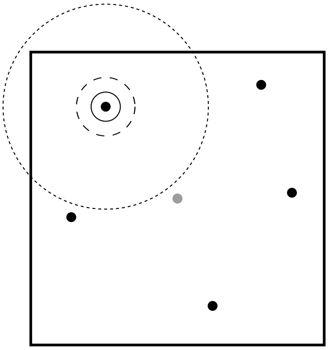

Fig. 6 The resource sharing task experimental setup. The *grey circle* in the *centre* is the charging station. The *black filled circles* are the robots (starting positions vary in each simulation). The *solid circle* around the *top left* robot represents the range of the obstacles sensor, the *dashed circle* represents the range of the robot sensor, and the *fine dashed circle* represents the range of the charging station sensor

additional sensors. Each robot has (i) eight IR sensors evenly distributed around its chassis for the detection of obstacles (walls or other robots) up to a range of 10 cm; (ii) eight sensors dedicated to the detection of other robots up to a range of 25 cm; (iii) a ring of eight sensors for the detection of the charging station up to a range of 1 m; (iv) a binary sensor that indicates whether or not the robot is currently being recharged; and (v) a proprioceptive sensor that reads the current energy level of the robot. The sensors (i), (ii) and (iii) return the distance of the object that is being sensed. The experimental setup and the sensor ranges are depicted in Fig. 6. The environment is a 3 m by 3 m square arena bounded by walls.

Each robot starts with full energy (1000 units), and the energy consumption increases linearly with the speed of the motors: a robot spends five units per second when motors are off, and 10 units of energy per second when both motors operate at their maximum speed. The charging station has the same diameter as a robot, and to recharge, the robots must remain static inside the charging station. The charging station is located in the centre of the arena, and charges a robot at a rate of 100 units of energy per second. Similarly to the aggregation experiments, each controller is evaluated in 10 simulations with random starting positions for the robots. Each simulation lasts for 2500 simulation steps. The highest scoring individual of each generation was post-evaluated in 100 simulations.

4.2 Configuration of the evolutionary algorithms

We used the same parameter values for novelty search and the NEAT algorithm as in the aggregation experiments (see Sect. 3.2). However, we continued each evolutionary run until the 400th generation, since the resource sharing task proved to be more challenging than the aggregation task.

The fitness function F_s used to evaluate the controllers is a linear combination of the number of robots alive at the end of the simulation (henceforth referred to as *survivors*) and the average energy of the robots throughout the entire simulation:

$$F_s = 0.9 \cdot \frac{|a_T|}{N} + 0.1 \cdot \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{e_{i,t}}{T N e_{\max}}, \tag{6}$$

where $|a_T|$ is the number of survivors, T is the length of the simulation, N is the number of robots in the swarm, $e_{i,t}$ is the energy of the robot i at time t, and $e_{\max}$ is the energy capacity of a robot. The second term of F_s concerning the average energy was included to differentiate solutions where the same number of robots survive. Without the second term, there would be no fitness gradient, since it is unlikely that the initial population contains solutions where at least one robot survives until the end.[3]

We experimented with behaviour characterisations of a different nature in the resource sharing experiments, compared to the characterisations used in the aggregation experiments. While in the aggregation task, we used spatial relationships between the robots sampled every 5 s during the simulation, in the resource sharing experiments, we use only quantities that characterise a simulation as a whole. We evaluated two behaviour characterisations:

$\mathbf{b}_{\text{simple}}$: The first characterisation is closely related to the fitness function, and it is composed of two values (normalised to the interval [0, 1]): (i) the number of robots that survive till the end of the simulation; and (ii) the average energy of all alive robots throughout the simulation. $\mathbf{b}_{\text{simple}}$ is given by

$$\mathbf{b}_{\text{simple}} = \left(\frac{|a_T|}{N}, \sum_{t=1}^{A} \sum_{i \in a_t} \frac{e_{i,t}}{A \cdot |a_t| \cdot e_{\max}} \right), \tag{7}$$

where A is the number of time steps in which there was at least one robot alive and a_t is the set of robots alive at time t.

$\mathbf{b}_{\text{extra}}$: The second behaviour characterisation is an extension of $\mathbf{b}_{\text{simple}}$. Two more features, which are not directly related to the fitness function, were added to the characterisation: (i) the average speed of the alive robots throughout the simulation; and (ii) the average distance of the alive robots to the charging station. The movement of a robot in a given instant is determined by the average wheel speed at that instant. The two additional features are also normalised to the interval [0, 1]. $\mathbf{b}_{\text{extra}}$ is defined as

$$\mathbf{b}_{\text{extra}} = \left(\mathbf{b}_{\text{simple}}, \sum_{t=1}^{A} \sum_{i \in a_t} \frac{s_{i,t}}{A \cdot |a_t| \cdot s_{\max}}, \sum_{t=1}^{A} \sum_{i \in a_t} \frac{d_{i,t}}{A \cdot |a_t| \cdot d_{\max}} \right), \tag{8}$$

where $s_{i,t}$ and $d_{i,t}$ are the speed of the robot i and its distance to the charging station, respectively, at time t. $s_{\max}$ is the maximum speed of a robot, and $d_{\max}$ is half the length of the diagonal of the arena.

4.3 Performance comparison

Figure 7 depicts the highest fitness scores achieved by novelty search, fitness-based evolution, and random evolution. Random evolution only achieves very low fitness scores (<0.2). In fitness-based evolution, the distribution of the highest fitness scores achieved is characteristically wide. Fitness-based evolution achieved close to the maximum fitness score in 10/30 of the evolutionary runs, but failed to evolve any viable solution (where at least one robot consistently survives) in 10/30 of the runs. Bootstrapping proved difficult in fitness-based evolution, as most runs got stuck in low regions of the fitness landscape for a large number of generations.

Novelty search, on the other hand, did not get stuck in local maxima. The result indicates that novelty search was unaffected by deception, and was capable of bootstrapping the evolutionary process. Novelty search with $\mathbf{b}_{\text{simple}}$ consistently achieved high fitness scores,

[3] We empirically determined the probability of randomly generating a solution where at least one robot survives until the end (in any of the 10 trials) to be approximately 1 %.

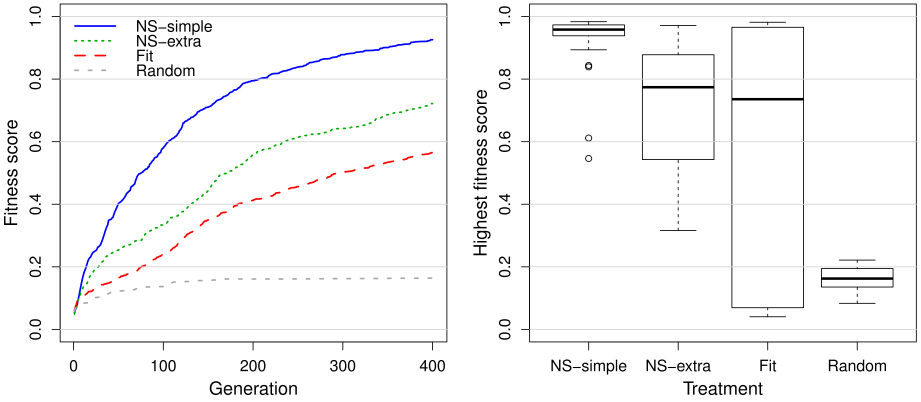

Fig. 7 Highest fitness scores achieved in the resource sharing task with novelty search with $\mathbf{b}_{\text{simple}}$ and $\mathbf{b}_{\text{extra}}$ (*NS-simple*, *NS-extra*), fitness-based evolution (*Fit*), and random evolution (*Random*). *Left*: average fitness value of the highest scoring individual found so far at each generation. The values are averaged over 30 independent evolutionary runs for each method. *Right*: box-plots of the highest fitness score found in each evolutionary run, for each method. The whiskers extend to the lowest and the highest data point within 1.5 times the interquartile range. Outliers are indicated by *circles*. The maximum fitness score in practice is about 0.98, which corresponds to all robots surviving until the end of the experiment, while maintaining high levels of energy

with 4–5 robots surviving in the best solutions. The fitness scores achieved with $\mathbf{b}_{\text{simple}}$ are significantly higher than those achieved by fitness-based evolution (Mann–Whitney U test, p-value < 0.01). With $\mathbf{b}_{\text{extra}}$, novelty search failed to match the performance of $\mathbf{b}_{\text{simple}}$, and the fitness scores achieved are significantly lower (p-value < 0.01). Within 400 generations, *NS-simple* could consistently achieve fitness scores close to the maximum value. However, it should be noted that *NS-extra* and *Fit* do not appear stable at that point, and if more generations were allowed, they might still improve. The differences between the behaviour characterisations and the evolved behaviours are discussed below.

4.4 Behavioural diversity

Through visual inspection of the solutions that achieved the highest fitness scores (above 0.95), we found that all evolutionary methods produced solutions that display similar behaviours. In the successful behaviours, the robots always start by searching for the charging station. Depending on the solution, the robots move in straight lines, in large circles, or in spirals, until the station has been located. The second part of the successful behaviours concerns the coordination of access to the charging station. We observed three different coordination behaviours (see Fig. 8):

Charge and go away: If a robot in the charging station detects another robot approaching, it leaves the station to let the approaching robot recharge. The leaving robot performs what resembles a random walk in the arena and eventually returns to the station to recharge.

Charge and surround: If a robot in the charging station detects another robot approaching, it leaves the station to let the approaching robot recharge. The leaving robot begins to circle the charging station, and approaches the station again when its energy level is below a solution-specific threshold.

Charge and wait: Once a robot is in the charging station, it continues to occupy the station until its energy level is above a solution-specific threshold. When the robot leaves, it moves

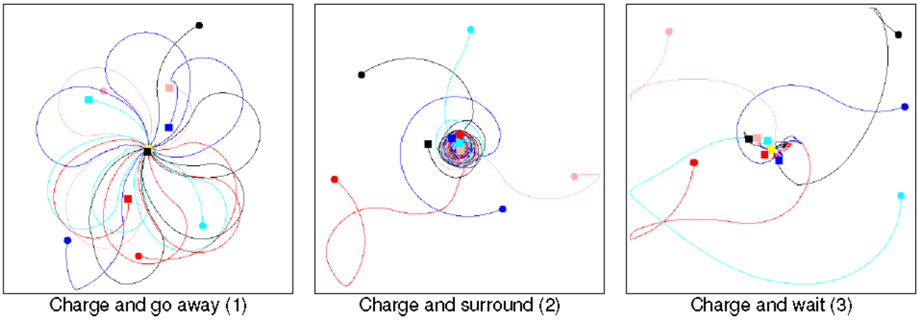

Fig. 8 The patterns of behaviour corresponding to the highest scoring solutions found by novelty search and fitness-based evolution in the resource sharing task. Each *line* represents the trajectory of the robot throughout the simulation. The *circles* indicate initial positions and *squares* indicate final positions. Videos of the behaviours are available as online supplemental material

only a short distance away from the station. Then, the robot remains almost static until its energy level is below a solution-specific threshold, at which point the robot tries to recharge again.

4.4.1 *Fitness-based evolution*

As mentioned above, fitness-based evolution did not consistently evolve solutions with high fitness scores (see Fig. 7). In fact, of the 30 runs conducted, 10 runs never evolved solutions with a fitness score much higher than the initial randomly generated population. We analysed the controllers evolved in the runs that only achieved low fitness scores and identified two behaviour patterns:

- The robots move slowly in small circles. When one of the robots detects the charging station, it moves towards the station, and occupies it till the end of the simulation. As a result, the rest of the swarm dies.
- All the robots remain almost static from the beginning of the simulation until they run out of energy.

Both behaviours represent local maxima in the fitness landscape. In the first case, evolution converges to solutions where only one robot survives at the expense of the rest of the swarm. In the second case, the evolutionary process starts to converge to controllers that reduce the wheel speed to conserve energy. Conserving energy causes the robots to survive longer, thus slightly increasing the fitness score of the controller. However, reducing the wheel speed also decreases the chance that a robot will find the charging station. Once an evolutionary process starts to converge to a local maximum based on energy conservation, it can take many generations to escape from that maximum, or evolution may not escape at all.

4.4.2 *Simple behaviour characterisation*

The highest scoring solutions evolved by novelty search follow the same behaviour patterns as the highest scoring solutions evolved by fitness-based evolution (see Fig. 8). However, an analysis of the explored behaviour space reveals that novelty search could in fact evolve a greater diversity of solutions (see Fig. 9). The greater diversity comes from variations of the

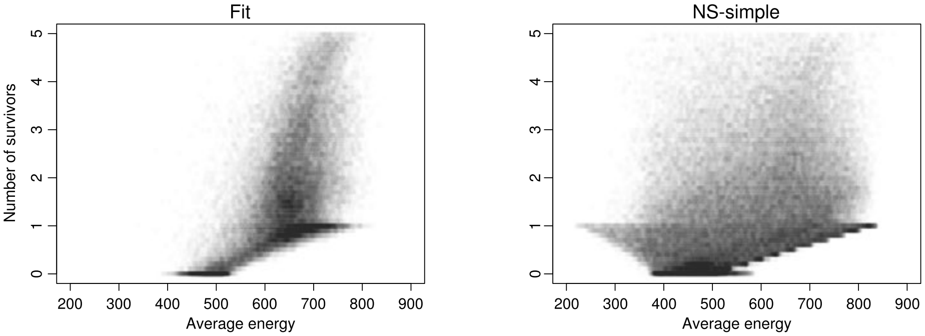

Fig. 9 Behaviour space exploration with fitness-based evolution (*Fit*) and novelty search with $\mathbf{b}_{\mathrm{simple}}$ (*NS-simple*), in all evolutionary runs. The x-axis is the average energy level of the robots still alive, the y-axis is the number of survivors. Each individual is mapped according to its characterisation. *Darker zones* indicate that there were more individuals evolved with the behaviour of that zone

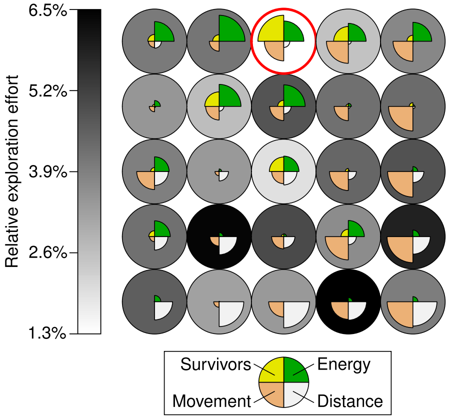

Fig. 10 Kohonen map representing the explored behaviour space in novelty search with $\mathbf{b}_{\mathrm{extra}}$. Each *circle* represents a behaviour pattern, depicted by the four slices of different colour. Each *slice* represents one component of the behaviour characterisation vector—the bigger the slice, the higher the value of that component. The *darker* the *background* of a *circle*, the more individuals were evolved with the corresponding behaviour (Color figure online)

same behaviour patterns. For instance, if the coordination behaviours use different energy thresholds to trigger entering and leaving the charging station, it will result in different average levels of energy. In such cases, the behaviour patterns may appear similar through visual inspection, but they have distinct behaviour characterisations.

4.4.3 Extra behaviour characterisation

We adapted the visualisation technique based on Kohonen maps to the four-dimensional behaviour space created by the $\mathbf{b}_{\mathrm{extra}}$ characterisation to analyse the degree of exploration of different behaviour regions. The results in Fig. 10 show how novelty search with the $\mathbf{b}_{\mathrm{extra}}$ characterisation explored the behaviour space—a notable variety of combinations of average energy, movement, and distance to the charging station. However, the behaviour dimension related to the number of surviving robots was the least explored dimension of the behaviour space. In almost all the explored regions, the number of surviving robots was either zero or one. Only a single behaviour region included behaviours in which the whole swarm often survives (highlighted in Fig. 10), and that region was one of the least explored.

The relatively low performance of novelty search is caused by the significantly larger behaviour space and the lack of correlation between the behaviour features. Novelty search can freely explore dimensions such as average movement and distance to charging station without any robots surviving. Furthermore, both of these dimensions are intuitively easier to explore than the dimension concerning the number of surviving robots. As such, the opportunistic nature of evolution causes the search to focus less on the surviving robots dimension, often inhibiting the evolution from finding successful solutions to the task. This is in contrast with the correlated features used in the $\mathbf{b}_{\text{cmcl}}$ characterisation from the aggregation experiments, in which novelty search performed well despite the enlarged behaviour space (see Sect. 3.4.3).

The two-dimensional $\mathbf{b}_{\text{simple}}$ characterisation restricts novelty search to explore the two dimensions directly related to the fitness function. Similar solutions in which robots move at different speeds are, for instance, conflated when the $\mathbf{b}_{\text{simple}}$ characterisation is used, whereas they are considered different (and potentially novel) when the $\mathbf{b}_{\text{extra}}$ characterisation is used. As a consequence, novelty search with the $\mathbf{b}_{\text{simple}}$ characterisation focussed exploration on the dimensions directly related to the fitness function and had significantly more success in finding solutions with high fitness scores. On the other hand, conflation can be prejudicial to the diversity of solutions that are evolved. For instance, by analysing the best solutions evolved with each characterisation, we verified that the behaviour pattern *Charge and wait* was less common with $\mathbf{b}_{\text{simple}}$ when compared to $\mathbf{b}_{\text{extra}}$. This behaviour is intuitively one of the most interesting solutions, since it takes advantage of the variable energy spending to extend the lifetime of the robots. However, it was conflated with $\mathbf{b}_{\text{simple}}$, since this characterisation does not consider the speed of the robots.

Our results suggest that caution must be exercised when behaviour characterisations are defined. Dimensions that novelty search may opportunistically explore at the cost of other dimensions that are more crucial for the task should not be included. Alternatively, a number of methods have been proposed that aim to guide exploration in novelty search towards high fitness solutions. In Sect. 5, we study the application of two such methods to the resource sharing task.

4.5 Neural network complexity

The experiments with the aggregation task showed that novelty search found successful solutions with less complex neural networks than fitness-based evolution. To determine if the same is true for the resource sharing task, we analysed the complexity of the solutions evolved for this task. Table 3 shows a comparison between the complexity of the solutions found by novelty search with $\mathbf{b}_{\text{simple}}$ and fitness-based evolution. The results show that there was a considerable difference between the complexity of the solutions evolved by the two methods. At fitness levels above 0.2, the simpler solutions evolved by novelty search were of significantly lower complexity than the solutions evolved by fitness-based evolution (Mann–Whitney U test, p-value < 0.05). Note that the complexity of the networks in the initial populations is 107 (29 neurons and 78 links).

5 Combining novelty and fitness

The results of the resource sharing experiments showed that although novelty search finds a broad diversity of behaviours, regions with high-fitness behaviours may never be explored. This issue has also been reported in other studies (Lehman and Stanley 2010a; Cuccu and Gomez 2011), and a common solution is to combine novelty search with fitness-based evolution (see Sect. 2.4). The combination is promising, because while novelty search

Table 3 Comparison of the least complex solutions evolved by fitness-based evolution (Fit) and novelty search with $\mathbf{b}_{\text{simple}}$ (NS). The *Complexity* columns lists the network complexity (sum of number of neurons and number of connections) of the least complex individual with a fitness score above a certain *Fitness level*. The *Generation* column lists the average generation in which the least complex individual was generated. The values are averages of 30 evolutionary runs for each method

Fitness level	Generation		Complexity	
	Fit	NS	Fit	NS
0.20	65	21	114.90	110.50
0.40	150	69	127.30	118.40
0.60	192	135	134.20	125.90
0.80	213	192	136.90	132.40
0.90	268	239	145.70	133.90

promotes exploration of the behaviour space, fitness-based evolution exploits existing high fitness solutions. In this section, we study the application of *progressive minimal criteria novelty search* (PMCNS) (Gomes et al. 2012) and *linear scalarisation* (Cuccu and Gomez 2011) to the evolution of solutions for the resource sharing task. We compare the results obtained with PMCNS and with linear scalarisation to the results obtained with pure novelty search and with fitness-based evolution.

5.1 Progressive minimal criteria novelty search

PMCNS (Gomes et al. 2012) is an extension of *minimal criteria novelty search* (MCNS) (Lehman and Stanley 2010a). In MCNS, the exploration of the behaviour space is restricted by domain-dependent minimal criteria that evolved individuals must meet in order to be selected for reproduction. The objective of PMCNS is to take advantage of the restrictions on behaviour space exploration provided by MCNS, but without having to define fixed, domain-dependent criteria a priori. In PMCNS, a dynamic fitness threshold is used as the minimal criterion: individuals with a fitness score above the threshold meet the criterion. The fitness threshold is progressively increased during the evolutionary process. The idea behind the increasing fitness criterion is to progressively restrict the search space to regions of the behaviour space with higher fitness scores, and thereby avoid that novelty search spends much or all effort on regions of the behaviour space with novel, but low fitness behaviours.

The minimal criterion starts at the theoretical minimum of the fitness score (typically zero), so all controllers initially meet the criterion. In each generation g, the new criterion mc_g is based on the fitness score v_g of the Pth percentile of the individuals in the current population, i.e., the fitness score below which P percent of the individuals fall. The parameter P controls the exigency of the minimal criterion (0—all individuals meet the criterion, 1—only the individual with the highest fitness meets the criterion). Only increases in the minimal criterion are allowed, and in order to smoothen the increase, the minimal criterion from the previous generation is used to determine the criterion for the current generation (see Eq. (9)). The score used for selection of each individual i in the population is then calculated according to Eq. (10):

$$mc_g = mc_{g-1} + \max\bigl(0, (v_g - mc_{g-1}) \cdot S\bigr), \tag{9}$$

$$\text{score}(i) = \begin{cases} nov_i & \text{if } fit_i \geq mc_g, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The variables nov_i and fit_i are the novelty and fitness score of the individual i, respectively. The smoothening parameter S controls the speed of adaptation of the minimal criterion.

5.2 Linear scalarisation of novelty and fitness scores

Cuccu and Gomez (2011) proposed a linear scalarisation of novelty and fitness score, as an approach to sustain diversity while improving the performance of traditional fitness-based evolution. Each individual i is evaluated to obtain both fitness score, fit_i, and novelty score, nov_i, which after being normalised (Eq. (11)) are combined according to Eq. (12):

$$\overline{fit}_i = \frac{fit_i - fit_{\min}}{fit_{\max} - fit_{\min}}, \qquad \overline{nov}_i = \frac{nov_i - nov_{\min}}{nov_{\max} - nov_{\min}}, \tag{11}$$

$$\mathrm{score}(i) = (1-\rho)\cdot\overline{fit}_i + \rho\cdot\overline{nov}_i. \tag{12}$$

The parameter ρ controls the relative weight of fitness and novelty, and must be specified by the experimenter (usually through trial and error). $fit_{\min}$ and $nov_{\min}$ are, respectively, the lowest fitness and novelty scores in the current population, and $fit_{\max}$ and $nov_{\max}$ are the corresponding highest scores.

5.3 Experimental setup

We used the experimental setup of the resource sharing task (see Sect. 4.1) for this set of experiments. The fitness function and the behaviour characterisations are also the same as we used in previous experiments (see Sect. 4.2).

For PMCNS, we chose a percentile value of $P = 0.50$ (values of 0.25, 0.50 and 0.75 were tested), which corresponds to the median value of the fitness scores in the population, and a smoothening parameter of $S = 0.25$, which corresponds to relatively slow increases in the minimal criterion value. For linear scalarisation, the parameter ρ was set to $\rho = 0.75$ (values of 0.25, 0.50 and 0.75 were tested), which means that the score of each individual is composed of 75 % of the novelty score and 25 % of the fitness score. A discussion of the parameter values for linear scalarisation and PMCNS can be found in Cuccu and Gomez (2011) and in Gomes et al. (2012), respectively.

5.4 Results

We ran experiments with both behaviour characterisations $\mathbf{b}_{\text{simple}}$ and $\mathbf{b}_{\text{extra}}$ to evaluate the performance of PMCNS and linear scalarisation in behaviour spaces of different dimensionality. The resulting fitness trajectories are shown in Fig. 11.

The results show that both PMCNS and linear scalarisation are more effective than pure novelty search when using $\mathbf{b}_{\text{extra}}$ (Mann–Whitney U test, p-value < 0.01), and achieve similar fitness scores when using $\mathbf{b}_{\text{simple}}$. With the $\mathbf{b}_{\text{extra}}$ characterisation, pure novelty search fails to reach high fitness scores, as evolution tends to focus the exploration on behaviour dimensions that are not directly relevant for solving the task. The results suggest that PMCNS and linear scalarisation can overcome this issue, and that the inclusion of a fitness component in the selection criteria helps to guide the evolutionary process towards solutions with high fitness. It should also be noted that PMCNS and linear scalarisation do not appear to be affected by the deceptiveness of the fitness function since they could consistently achieve high fitness scores.

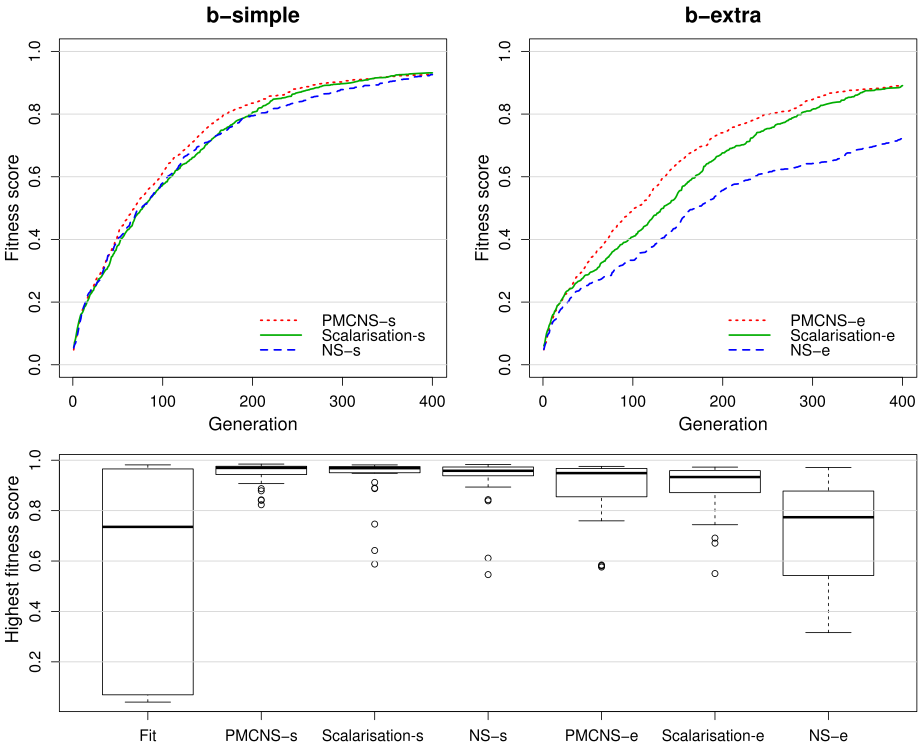

Fig. 11 Highest fitness scores obtained in the resource sharing task with fitness-based evolution (*Fit*), and novelty-based evolutionary techniques with $\mathbf{b}_{\text{simple}}$ (*PMCNS-s*, *Scalarisation-s*, *NS-s*) and $\mathbf{b}_{\text{extra}}$ (*PMCNS-e*, *Scalarisation-e*, *NS-e*). *Top*: average fitness value of the highest scoring individual found so far at each generation. The values are averaged over 30 independent evolutionary runs for each method. *Bottom*: box-plots of the highest fitness score found in each evolutionary run, for each method. The whiskers extend to the lowest and the highest data point within 1.5 times the interquartile range. Outliers are indicated by *circles*

Figures 12 and 13 depict the behaviour space exploration with $\mathbf{b}_{\text{simple}}$ and $\mathbf{b}_{\text{extra}}$, respectively. An analysis of the behaviour space exploration shows that PMCNS and linear scalarisation have a greater focus on regions associated with high fitness scores compared to pure novelty search. The coverage of the behaviour space was not negatively affected in PMCNS and linear scalarisation. In fact, the two methods found the same broad range of behaviours as pure novelty search. The difference between pure novelty search and novelty search combined with fitness-based search is the amount of exploration done in each behaviour region: PMCNS and linear scalarisation focussed less on the low fitness behaviours (few or no robots surviving till the end of the simulation) and more on the high-fitness behaviours (high number of surviving robots). As a result, the PMCNS and linear scalarisation could achieve solutions with significantly higher fitness scores than the best solutions evolved with pure novelty search.

6 Conclusions

We studied the application of novelty search to the evolution of controllers for swarms of robots. The study was based on two distinct swarm robotics tasks: (i) an aggregation task,

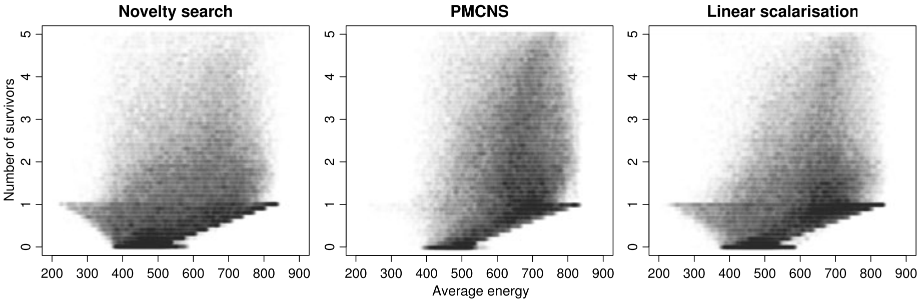

Fig. 12 Behaviour space exploration for each variant of novelty search, with the $\mathbf{b}_{\text{simple}}$ characterisation, in all evolutionary runs. The x-axis is the average energy level of the robots still alive, the y-axis is the number of survivors. Each individual is mapped according to its behaviour. *Darker zones* indicate that there were more individuals evolved with the behaviour of that zone

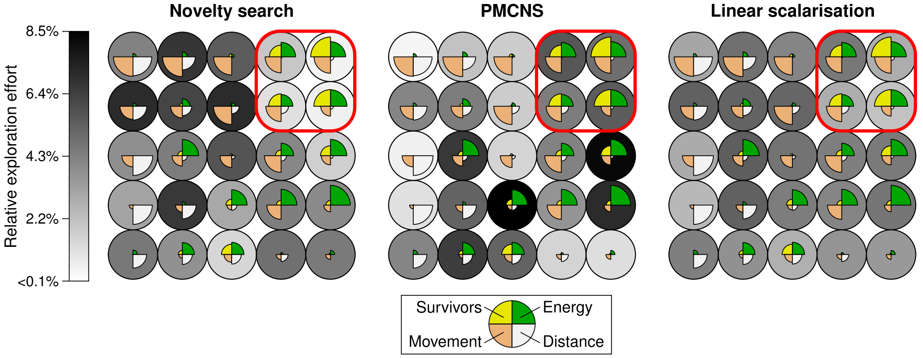

Fig. 13 Kohonen maps representing the explored behaviour space with each variant of novelty search, with the $\mathbf{b}_{\text{extra}}$ characterisation. Each *circle* represents a behaviour pattern, depicted by the four slices of different colour. Each *slice* represents one component of the behaviour characterisation—the bigger the slice, the greater the value of that component. The *darker* the *background* of a *circle* is, the more individuals were evolved with the corresponding behaviour. The behaviour patterns with higher fitness scores are indicated in the *upper right corner* of each map (Color figure online)

and (ii) a resource sharing task. The aggregation task was non-deceptive, as fitness-based evolution consistently managed to find high fitness solutions. Nevertheless, novelty search could achieve a similar performance in terms of fitness scores. We also showed how novelty search found several alternative and successful solutions to the task. Our analysis was based on Kohonen self-organising maps that allowed for the visualisation of the degree of exploration conducted in different regions of the behaviour space.

The resource sharing task was a deceptive setup in which fitness-based evolution often got stuck in local maxima. Novelty search was unaffected by deception and displayed a significantly better performance than fitness-based evolution. In both tasks, novelty search was distinctively able to bootstrap the evolutionary process, it could consistently find behaviours with high fitness scores early in the evolutionary process, and it was able to find successful solutions with lower neural network complexity than the solutions evolved by fitness-based evolution.

To the best of our knowledge, our study is the first in which novelty search has been applied to evolutionary swarm robotics. Since behaviour characterisations are domain-dependent and a fundamental component in novelty search, we studied two different approaches to the design of characterisations: one based on the spatial inter-robot relationships sampled at regular intervals, and one based on two to four quantities that summarise the swarm behaviour throughout an entire experiment. None of the characterisations depends on the swarm size and they are thus scalable. In our experiments, we combined different behaviour characterisations and found that such combinations were only effective when the dimensions in the characterisation were directly related to the task. The opportunistic nature of artificial evolution will cause the search to first focus on the behaviour dimensions that are easier to explore. If such dimensions are not related to the task, the search will spend considerable effort in unfruitful regions of the behaviour space, reducing the effectiveness of novelty search.

We discovered that novelty search may not always find high-fitness solutions, especially when the behaviour space has dimensions that are unrelated or only weakly related to the objective. To overcome this issue, we studied two variants of novelty search, PMCNS and linear scalarisation of novelty and fitness scores, in which novelty search operates in concert with fitness-based evolution. Our experimental results showed that PMCNS and linear scalarisation are effective in guiding the exploration towards behavioural regions of higher fitness, without compromising the capacity of novelty search to find a broad diversity of solutions. The diversity of evolved solutions was neither negatively affected in PMCNS nor in linear scalarisation, when compared to pure novelty search.

Overall, our study shows that novelty search can be successfully applied to swarm robotics systems. Novelty search has several advantages over fitness-based evolution. One of the prominent advantages is that novelty search often produces a broad diversity of successful behaviours based on self-organisation. In the swarm robotics domain, diversity and self-organisation are particularly important because of the difficulties of designing such behaviours by hand. Moreover, novelty search is unaffected by deception, less prone to bootstrapping issues, and can evolve solutions with less complex neural networks. Novelty search is therefore a promising alternative for the artificial evolution of controllers for swarm robotics systems.

6.1 Future work

Our results showed that care must be taken in the definition of macroscopic behaviour characterisations, especially when combining distinct behaviour features. It is necessary to ensure that (i) each component of the characterisation is relevant to the task, and (ii) that the components do not differ too much in terms of how easy or hard it is for novelty search to explore the corresponding dimensions of the behaviour space. It may be difficult to guarantee that macroscopic behaviour characterisations always meet these criteria, and in ongoing work we are therefore studying how individual weights can be assigned to components and modified during evolution, either manually or automatically. Such weights could ensure that particularly important dimensions are thoroughly explored.

While pure novelty search performed reasonably well in our tasks, our results show that the inclusion of a fitness component can further increase the performance of novelty search. As such, in future work we are going to elaborate on strategies for combining novelty and fitness. For instance, we are studying if the use of dynamic weights in linear scalarisation can bring significant advantages. Such approach could allow the search to focus more on exploitation or more on exploration at different stages of the evolutionary process.

An alternative that we are currently studying is *generic novelty measures* for swarm robotics. Doncieux and Mouret (2010) proposed a behaviour characterisation based on the mapping between the sensor input and the actuator output. We are studying how such characterisations could be defined in the collective robotics domain, and if they represent a viable alternative to domain-dependent novelty measures. Such generic measures are typically used in combination with fitness-based evolution, since the behaviour spaces created by generic measures tend to be vast and only weakly related to the objective.

Supplemental material Videos of the behaviours are available as online supplemental material and at the following URL: http://home.iscte-iul.pt/~alcen/si2013. The source code of the software used in the experiments can be found at: https://github.com/jorgemcgomes/evosimbad.

Acknowledgements This research has been supported by Fundação para a Ciência e a Tecnologia (FCT) grants PTDC/EEACRO/104658/2008, PEst-OE/EEI/LA0008/2011, and SFRH/BD/89095/2012.

References

Ampatzis, C., Tuci, E., Trianni, V., & Dorigo, M. (2008). Evolution of signaling in a multi-robot system: categorization and communication. *Adaptive Behavior*, *16*(1), 5–26.

Bahgeçi, E., & Şahin, E. (2005). Evolving aggregation behaviors for swarm robotic systems: a systematic case study. In *Swarm intelligence symposium* (pp. 333–340). New York: IEEE Press.

Baldassarre, G., Nolfi, S., & Parisi, D. (2003). Evolving mobile robots able to display collective behaviors. *Artificial Life*, *9*(3), 255–268.

Baldassarre, G., Trianni, V., Bonani, M., Mondada, F., Dorigo, M., & Nolfi, S. (2007). Self-organized coordinated motion in groups of physically connected robots. *IEEE Transactions on Systems, Man and Cybernetics*, *37*(1), 224–239.

Bayindir, L., & Şahin, E. (2007). A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering and Computer Sciences*, *15*(2), 115–147.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. New York: Oxford University Press.

Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, *7*(1), 1–41.

Cao, Y. U., Fukunaga, A. S., & Kahng, A. B. (1997). Cooperative mobile robotics: antecedents and directions. *Autonomous Robots*, *4*(1), 7–27.

Castelli, M., Manzoni, L., & Vanneschi, L. (2011). A method to reuse old populations in genetic algorithms. In *LNCS: Vol. 7026. Portuguese conference on artificial intelligence (EPIA)* (pp. 138–152). Berlin: Springer.

Chellapilla, K., & Fogel, D. B. (1999). Evolving neural networks to play checkers without relying on expert knowledge. *IEEE Transactions on Neural Networks*, *10*(6), 1382–1391.

Correll, N., & Martinoli, A. (2007). Modeling self-organized aggregation in a swarm of miniature robots. In *IEEE international conference on robotics and automation (ICRA)* (pp. 379–384). New York: IEEE Press.

Cuccu, G., & Gomez, F. J. (2011). When novelty is not enough. In *LNCS: Vol. 6624. European conference on the applications of evolutionary computation (EvoApplications)* (pp. 234–243). Berlin: Springer.

Cuccu, G., Gomez, F. J., & Glasmachers, T. (2011). Novelty-based restarts for evolution strategies. In *IEEE congress on evolutionary computation (IEEE CEC)* (pp. 158–163). New York: IEEE Press.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Hoboken: Wiley.

Doncieux, S., & Mouret, J.-B. (2010). Behavioral diversity measures for evolutionary robotics. In *IEEE congress on evolutionary computation (IEEE CEC)* (pp. 1–8). New York: IEEE Press.

Doncieux, S., Mouret, J.-B., Bredeche, N., & Padois, V. (2011). Evolutionary robotics: exploring new horizons. In *Studies in computational intelligence: Vol. 341. New horizons in evolutionary robotics* (pp. 3–25). Berlin: Springer.

Goldberg, D. E., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: second international conference on genetic algorithms* (pp. 41–49). Mahwah: Erlbaum.

Gomes, J., Urbano, P., & Christensen, A. L. (2012). Progressive minimal criteria novelty search. In *LNAI: Vol. 7637. Ibero-American conference on artificial intelligence (IBERAMIA)* (pp. 281–290). Berlin: Springer.

Gomez, F., & Mikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptative Behaviour*, *5*(3–4), 317–342.

Gross, R., & Dorigo, M. (2008). Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling. *Adaptive Behavior*, *16*(5), 285–305.

Gutiérrez, A., Campo, A., Dorigo, M., Amor, D., Magdalena, L., & Monasterio-Huelin, F. (2008). An open localization and local communication embodied sensor. *Sensors*, *8*(11), 7545–7563.

Harvey, I., Husbands, P., & Cliff, D. (1993). Issues in evolutionary robotics. In *International conference on simulation of adaptive behavior (SAB)* (pp. 364–373). Cambridge: MIT Press.

Hauert, S., Zufferey, J.-C., & Floreano, D. (2009). Evolved swarming without positioning information: an application in aerial communication relay. *Autonomous Robots*, *26*(1), 21–32.

Hornby, G. (2006). ALPS: the age-layered population structure for reducing the problem of premature convergence. In *Genetic and evolutionary computation conference (GECCO)* (pp. 815–822). New York: ACM.

Hu, J., Goodman, E. D., Seo, K., Fan, Z., & Rosenberg, R. (2005). The hierarchical fair competition (HFC) framework for sustainable evolutionary algorithms. *Evolutionary Computation*, *13*(2), 241–277.

Hugues, L., & Bredeche, N. (2006). Simbad: an autonomous robot simulation package for education and research. In *LNCS: Vol. 4095. International conference on simulation of adaptive behavior (SAB)* (pp. 831–842). Berlin: Springer.

Hutter, M., & Legg, S. (2006). Fitness uniform optimization. *IEEE Transactions on Evolutionary Computation*, *10*(5), 568–589.

Jeanson, R., Rivault, C., Deneubourg, J.-L., Blanco, S., Fournier, R., Jost, C., & Theraulaz, G. (2005). Self-organized aggregation in cockroaches. *Animal Behaviour*, *69*(1), 169–180.

Jones, T., & Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *International conference on genetic algorithms (ICGA)* (pp. 184–192). San Mateo: Morgan Kaufmann.

Kernbach, S., & Kernbach, O. (2011). Collective energy homeostasis in a large-scale microrobotic swarm. *Robotics and Autonomous Systems*, *59*(12), 1090–1101.

Kirkpatrick, D. A. (2012) *Novelty search in competitive coevolution using normalized compression distance*. Master thesis, College of Engineering, Florida Institute of Technology.

Kistemaker, S., & Whiteson, S. (2011). Critical factors in the performance of novelty search. In *Genetic and evolutionary computation conference (GECCO)* (pp. 965–972). New York: ACM.

Knowles, J., Watson, R., & Corne, D. (2001). Reducing local optima in single-objective problems by multi-objectivization. In *LNCS: Vol. 1993. Evolutionary multi-criterion optimization* (pp. 269–283). Berlin: Springer.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480.

Krcah, P. (2010). Solving deceptive tasks in robot body-brain co-evolution by searching for behavioral novelty. In *International conference on intelligent systems design and applications (ISDA)* (pp. 284–289). New York: IEEE Press.

Lehman, J., & Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *International conference on the synthesis and simulation of living systems (ALIFE)* (pp. 329–336). Cambridge: MIT Press.

Lehman, J., & Stanley, K. O. (2010a). Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Genetic and evolutionary computation conference (GECCO)* (pp. 103–110). New York: ACM.

Lehman, J., & Stanley, K. O. (2010b). Efficiently evolving programs through the search for novelty. In *Genetic and evolutionary computation conference (GECCO)* (pp. 837–844). New York: ACM.

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

Lehman, J., & Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *Genetic and evolutionary computation conference (GECCO)* (pp. 211–218). New York: ACM.

Liu, W., Winfield, A. F. T., & Sa, J. (2007). Modelling swarm robotic systems: a case study in collective foraging. In *Towards autonomous robotic systems (TAROS)* (pp. 25–32).

Michaud, F., & Robichaud, E. (2002). Sharing charging stations for long-term activity of autonomous robots. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (Vol. 3, pp. 2746–2751). New York: IEEE Press.

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

Mouret, J.-B. (2011). Novelty-based multiobjectivization. In *Studies in computational intelligence: Vol. 341. New horizons in evolutionary robotics* (pp. 139–154). Berlin: Springer.

Mouret, J.-B., & Doncieux, S. (2009). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *IEEE congress on evolutionary computation (IEEE CEC)* (pp. 1161–1168). New York: IEEE Press.

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

Muñoz Meléndez, A., Sempé, F., & Drogoul, A. (2002). Sharing a charging station without explicit communication in collective robotics. In *International conference on simulation of adaptive behavior (SAB)* (pp. 383–384). Cambridge: MIT Press.

Nelson, A. L., Barlow, G. J., & Doitsidis, L. (2009). Fitness functions in evolutionary robotics: a survey and analysis. *Robotics and Autonomous Systems*, *57*(4), 345–370.

Pini, G., & Tuci, E. (2008). On the design of neuro-controllers for individual and social learning behaviour in autonomous robots: an evolutionary approach. *Connection Science*, *20*(2–3), 211–230.

Şahin, E. (2005). Swarm robotics: from sources of inspiration to domains of application. In *LNCS: Vol. 3342. International workshop on swarm robotics* (pp. 10–20). Berlin: Springer.

Schmidt, M., & Lipson, H. (2011). Age-fitness Pareto optimization. In *Genetic and evolutionary computation: Vol. 8. Genetic programming theory and practice VIII* (pp. 129–146). Berlin: Springer.

Soltoggio, A., & Jones, B. (2009). Novelty of behaviour as a basis for the neuro-evolution of operant reward learning. In *Genetic and evolutionary computation conference (GECCO)* (pp. 169–176). New York: ACM.

Soysal, O., Bahgeçi, E., & Şahin, E. (2007). Aggregation in swarm robotic systems: evolution and probabilistic control. *Turkish Journal of Electrical Engineering and Computer Sciences*, *15*(2), 199–225.

Sperati, V., Trianni, V., & Nolfi, S. (2008). Evolving coordinated group behaviours through maximisation of mean mutual information. *Swarm Intelligence*, *2*(2–4), 73–95.

Sprong, C. (2011). *Common tasks in evolutionary robotics, an overview*. Technical report, Faculty of Sciences, University of Amsterdam, Netherlands. URL http://www.few.vu.nl/nl/Images/werkstuk-sprong_tcm38-217791.pdf.

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural network through augmenting topologies. *Evolutionary Computation*, *10*(2), 99–127.

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

Trianni, V., Gross, R., Labella, T. H., Şahin, E., & Dorigo, M. (2003). Evolving aggregation behaviors in a swarm of robots. In *LNCS: Vol. 2801. European conference on artificial life (ECAL)* (pp. 865–874). Berlin: Springer.

Trianni, V., Nolfi, S., & Dorigo, M. (2006). Cooperative hole avoidance in a swarm-bot. *Robotics and Autonomous Systems*, *54*(2), 97–103.

Uchibe, E., Yanase, M., & Asada, M. (2002). Behavior generation for a mobile robot based on the adaptive fitness function. *Robotics and Autonomous Systems*, *40*(2–3), 69–77.

Watson, R. A., & Pollack, J. B. (2001). Coevolutionary dynamics in a minimal substrate. In *Genetic and evolutionary computation conference (GECCO)* (pp. 702–709). San Mateo: Morgan Kaufmann.

Whitley, L. D. (1991). Fundamental principles of deception in genetic search. In *Foundations of genetic algorithms* (pp. 221–241). San Mateo: Morgan Kaufmann.

Zaera, N., Cliff, D., & Bruten, J. (1996). (Not) Evolving collective behaviours in synthetic fish. In *International conference on simulation of adaptive behavior (SAB)* (pp. 635–644). Cambridge: MIT Press.