Introducing Novelty Search in Evolutionary Swarm Robotics

Jorge Gomes[1,3], Paulo Urbano[1], and Anders Lyhne Christensen[2,3]

[1] LabMAg, Faculdade de Ciências da Universidade de Lisboa, Portugal
[2] Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
[3] Instituto de Telecomunicações, Lisboa, Portugal
{jgomes,pub}@di.fc.ul.pt, anders.christensen@iscte.pt

Abstract. Novelty search is a recent and promising evolutionary technique. The main idea behind it is to reward novel solutions instead of progress towards a fixed goal, in order to avoid premature convergence and deception. In this paper, we use novelty search together with NEAT, to evolve neuro-controllers for a swarm of simulated robots that should perform an aggregation task. In the past, novelty search has been applied to single robot systems. We demonstrate that novelty search can be applied successfully to multirobot systems, and we discuss the challenges introduced when moving from a single robot setup to a multirobot setup. Our results show that novelty search can outperform the fitness-based evolution in swarm robotic systems, finding (i) a more diverse set of successful solutions to an aggregation task, (ii) solutions with higher fitness scores earlier in the evolutionary runs, and (iii) simpler solutions in terms of the topological complexity of the evolved neural networks.

1 Introduction

Novelty search [10] is a divergent evolutionary technique. In traditional evolutionary computation, candidate solutions are scored by an objective function that has been derived directly from the task or problem for which a solution is sought. Novelty search does not drive the evolutionary process toward a fixed goal. In novelty search, candidate solutions are scored based on how different they are from solutions seen so far and the evolutionary process is therefore continuously driven towards novelty. As a result, novelty search has the potential to overcome deception [4]. Deception can be a challenging problem in evolutionary computation and occurs when the evolutionary process converges prematurely to a local optimum because the objective function fails to reward the intermediate steps needed to achieve the final goal. Lehman and Stanley [10] have shown that, although novelty search does not pursue a goal directly, it may be able to find the goal faster and more consistently than traditional fitness-based evolution. Novelty search has also proven to be able to find a greater diversity of solutions to a problem than traditional fitness-based evolution [11].

Novelty search has been successfully applied to many domains, including non-collective evolutionary robotics in tasks such as maze navigation [10,13], T-maze

M. Dorigo et al. (Eds.): ANTS 2012, LNCS 7461, pp. 85–96, 2012.

tasks that require lifetime learning [14], biped walking [10], and the deceptive tartarus problem [3]. There are many motivations behind the use of evolutionary techniques for the design of a control system for a robot [5]. In a multirobot domain in particular, the dynamical interactions among robots and the environment make it difficult to hand-design a control system for the robots that yields the desired macroscopic swarm behaviours. However, artificial evolution has been shown capable of exploiting these dynamic features and synthesise self-organised behaviours [18].

In this paper, we use novelty search to evolve neural controllers for swarm robotic systems, where fitness-based evolutionary approaches has been previously used. Our motivation for applying novelty search to swarm robotic systems is their high level of complexity, resulting from the intricate dynamics between many interacting units. As the complexity of a task or a system increases, artificial evolution is more likely to get affected by deception [19], and novelty search has been shown capable of overcoming deception [10]. The drive of novelty search towards behavioural diversity is also valuable because it can generate a diversity of solutions in a single evolutionary run, as opposed to fitness-based evolution, in which a particular run often converges to a single solution.

There are many works that describe the evolution of robot swarms with neuroevolution methods that optimise only the weights of the neural network. However, the evolution of the network topology along with the weights has proved to be beneficial in other domains [19,16]. In this paper, we use NEAT (NeuroEvolution of Augmenting Topologies) [17] to evolve the neural controllers used by the robots in a swarm. NEAT is a method that evolves both the network topology and weights, allowing solutions to become gradually more complex as they become better [17]. The use of novelty search together with NEAT is motivated by the complexifying nature of NEAT, which imposes some order in the exploration of the behaviour space, because simple controllers are explored before moving on to more complex ones.

We use an aggregation task for the experiments in this study. In this task, the robots should move around in an environment to search for each other and ultimately form a single aggregate. Aggregation is of particular interest since it stands as a prerequisite for other forms of cooperation in swarm robotics systems [18]. This task has been used in previous works in evolutionary swarm robotics [2,18,1]. In our experiments, the domain was made challenging by increasing the size of the arena and by reducing the sensors capabilities, compared to the previous studies on aggregation in robots.

2 Background

In this section, we review the related work on aggregation in evolutionary robotics, the NEAT neuroevolution method used in our experiments, and the novelty search method.

2.1 Evolution of Aggregation Behaviours

Several works describe the evolution of aggregation behaviours in swarms of robots, where neural networks with fixed topologies are evolved via evolutionary algorithms guided by fitness. Baldassarre et al. [2] successfully evolved controllers for a swarm of robots to aggregate and move towards a light source in a clustered formation. Trianni et al. [18] describe the evolution of a swarm of simple robots to perform aggregation in a square arena. In this experiment, two different behaviours were evolved: a *static clustering* which forms compact and stable aggregates and a *dynamic clustering* which creates loose but moving aggregates. Bahgeçi et al. [1] used a similar experimental setup as [18], and studied how some parameters of the evolutionary methods affect the performance and the scalability of the behaviours in swarm robotic systems.

In these studies, the robots used directional sound sensors and sound signalling to identify other robots in the environment. Sound signalling enabled robots to follow sound gradients in order to aggregate. In fact, these works show that neural networks without any hidden neurons are sufficient to successfully solve the task. In our work, we make the aggregation task more challenging: we remove the sound gradient, decrease the range of the sensors, and increase the size of the arena. These modifications increase the difficulty of the task and may require quite different strategies for aggregation because it is harder for the robots to find each other [15].

2.2 NEAT

NEAT [17] is a neuroevolution method that optimises both 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, potentially leading to increasingly sophisticated behaviour. A key feature in NEAT is its distinctive approach to maintaining a healthy 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 having to rely on complex topological comparisons. Speciation in NEAT protects new structural innovations by reducing competition between differing networks, giving time for newer and more complex structures to have their weights 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 later elaborate on them. In effect, NEAT searches for a compact, appropriate network topology by incrementally complexifying existing structures.

2.3 Novelty Search

In novelty search [10], individuals in an evolving population are selected based exclusively on how different their behaviour is when compared to the other

behaviours discovered so far. Implementing novelty search requires little change to any evolutionary algorithm aside from replacing the fitness function with a domain dependent novelty metric. This metric measures how different an individual is from the other individuals with respect to their behaviour. The use of a novelty measure creates a constant pressure to evolve individuals with novel behaviour features.

The novelty of a newly generated individual is computed with respect to the behaviours of an archive of past individuals and to the current population, giving a comprehensive sample of where the search has been and where it currently is. However, the archive does not contain all of the behaviours previously explored, in order to minimise the impact in the algorithm's computational complexity. The archive is initially empty, and behaviours are added to it if they are significantly different from the ones already there, i.e., if their novelty is above some threshold. The purpose of the archive is to allow the penalisation of future individuals that exhibit previously explored behaviours.

The novelty metric characterises how far away the new individual is from the rest of the population and its predecessors in behaviour space, determining the sparseness at any point in that 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

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

where μ_i the ith-nearest neighbour of x with respect to the distance metric $dist$, which is a domain-dependent measure of behavioural difference between two individuals in the search space. Candidates from more sparse regions of the behaviour space thus receive higher novelty scores, guiding the search towards what is new, with no other explicit objective.

3 Aggregation Experiments

In this section, we apply novelty search to the aggregation task and compare it with fitness-based evolution. Three experiments were performed using different novelty measures: one highly correlated with the fitness function, an alternative measure only weakly correlated, and finally a combination of the two. In each experiment, the performance of novelty search was compared to the performance of traditional fitness-based evolution. NEAT with random selection is used as a baseline for performance comparisons.

3.1 Experimental Setup

The simulated environment is modelled in a customised version of the Simbad 3d Robot Simulator [7]. The environment is a 5 m by 5 m square arena bounded by walls. The robots are modelled based on the the e-puck educational robot [12],

but do not strictly follow its specification. Each simulated robot has 8 IR sensors evenly distributed around its chassis for the detection of obstacles (walls or other robots) within a range of 10 cm, and 8 sensors dedicated to the detection of other robots within 25 cm range. An additional sensor calculates the percentage of nearby robots, relative to the size of the swarm, within a radius of 25 cm.

The swarm is homogeneous and the controllers of the robots are time recurrent neural networks. For fitness-based evolution, we used the NEAT implementation available in the Encog 3.0.1 library [6]. For novelty search, we extended the same NEAT implementation following the description and parameters in [10], with a k value of 15 and a dynamic archive threshold [9]. This dynamic threshold ensures a constant and reasonable flow of individuals to the archive, at an average rate of 2 individuals per generation. The NEAT parameters were the same in both evolutionary methods: the crossover rate was 25%, the mutation rate 10%, the population size 200, and each evolution runs for 250 generations. The rest of the parameters were the default of the Encog implementation.

To evaluate each controller, 10 simulations are run with it, varying the number of robots and their starting positions and orientations. The starting positions and orientations are random but ensure a minimum distance between the robots. The group size varies from 3 to 10, with each controller being run at least once with every group size. Each simulation lasts for 500 s of simulated time.

The fitness function that evaluates each simulation is based on the average distance to the centre of mass, also used in [18]. The average distance is sampled throughout the simulation at regular intervals of 10 s. The samples are then combined in a single fitness value using a weighted average, with linearly more weight towards the end of the simulation. The fitness F of a simulation with T time steps and N robots is defined as:

$$F = 1 - \frac{1}{\sum \frac{t}{T}} \sum_{t=1}^{T} \frac{t}{T} \sum_{i=1}^{N} \frac{dist(\mathbf{R}_t, \mathbf{r}_{i_t})}{N} \tag{2}$$

where $\mathbf{R}_t$ is the centre of mass at each instant, and $\mathbf{r}_{i_t}$ is the position of each robot. The fitness values obtained in each of the 10 simulations are combined in a single value using the harmonic mean, which gives more weight to the lower values, as advocated in [1].

As mentioned above, the novelty measure characterises the distance between one controller and the others in behaviour space. We use the Euclidian distance between vectors that represent the level of aggregation along time. These vectors are built by measuring behaviour features at regular intervals throughout the simulation (every 10 s). We devised three ways of measuring the group behaviour, which will be explained in the next sections. As 10 simulations are conducted to evaluate each controller, its behaviour vector is the average of the vectors obtained in each of the simulations. In order to compare novelty search with the fitness-based evolution, the controllers evolved by novelty search were also evaluated with the fitness function F. It is important to note that the fitness scores did not have any influence in the novelty search experiments.

3.2 The First Experiment

The first behaviour measure uses the same metric as the fitness function; a vector is built with the average distance to centre of mass, sampled throughout the simulation. Considering a simulation with N robots and T temporal samples, the behaviour vector $\mathbf{b_{cm}}$ that characterises a controller is given by:

$$\mathbf{b_{cm}} = \frac{1}{N}\left[\sum_{i=1}^{N} dist(\mathbf{R}_1, \mathbf{r}_{i_1}), \cdots, \sum_{i=1}^{N} dist(\mathbf{R}_T, \mathbf{r}_{i_T})\right] \quad . \tag{3}$$

In our experiments, the sampling rate was 10 s and the simulation time 500 s, resulting in a behaviour vector of length 50.

The fitness scores of the highest scoring individuals evolved using novelty search and fitness driven evolution, respectively, are listed in Table 1. There is not a significant difference between the fitness of the controllers evolved in these experiments, but both methods are significantly better than the random evolution (Student's t-test with p-value < 0.05). If we look at the behaviours of the best controllers evolved by both methods, significant differences are found, despite the similar fitness values. In the fitness-based evolution, the highest scoring controllers were always very similar, displaying only one distinctive behaviour: the robots explore the environment in large circles, and form static clusters when they encounter one another. If the cluster is small, the robots abandon it after a while and start exploring again.

Novelty search, on the other hand, found several distinct high-scoring controllers that could perform the aggregation task: (1) The robots go straight forward until they hit the wall, and then, depending on the impact angle, they stay there for a while or start moving along the wall until they find other robots; (2) Similar to (1), but when they meet each other they continue to follow the wall until they hit a corner, aggregating there; (3) Similar to the behaviour evolved by fitness, but without splitting the small clusters; (4) Similar to (3) but navigating in the environment only in straight trajectories instead of curves. It is important to note that each evolutionary run of novelty search could evolve several different solutions, finding many (if not all) of the solutions described above and variants of them.

The main difference between the behaviours was that novelty search evolved controllers that exploited the wall to achieve better solutions, while in the fitness-based evolution robots always avoided navigating near the walls. Our hypothesis is that learning to navigate along the walls requires going against the

Table 1. Highest fitness found with each evolutionary method. The values were obtained with 10 runs for each experiment. Individuals with fitness value over 0.8 are considered to be solutions to the task. Note that in practice the minimum fitness value is not 0, since an initial random population has an average fitness of around 0.6.

Evolutionary Setup	Average	Max.	Min.
Fitness-based NEAT	0.863	0.892	0.826
NEAT with novelty search	0.864	0.906	0.828
NEAT with random selection	0.725	0.752	0.706

fitness gradient. If the robots go towards the walls, they will often end up in different ones, and staying there will result in a low fitness because the centre of mass will be in the centre of the arena, far from the robots. On the other hand, avoiding the walls results in better fitness because they will be on average closer to the centre. If the fitness evolution misses the stepping stone of being close to the walls, it will hardly be able to reach behaviours that require the use of walls to achieve aggregation. This is an important result because it demonstrates that the fitness function is preventing the evolution of certain types of solutions.

To confirm our hypothesis, we analysed the behaviour space explored in novelty search and in fitness evolution. To facilitate this analysis, all the individuals evolved in fitness evolution were also evaluated with the same behaviour measure that was used in novelty search. Since each behaviour description is a long vector, we applied a dimensionality reduction method in order to visualise the behaviour space. We used a Kohonen self-organising map [8], a type of neural network trained using unsupervised learning to produce a two-dimensional discretisation of the input space of the training samples, preserving the topological relations. The map was trained with all the behaviours found both in novelty search and in fitness evolution, and then the behaviours found by each method were mapped individually to the trained map. The resulting maps can be seen in Figure 1.

As it can be seen in the maps, the fitness-based evolution avoids the zones where the average distance to the centre of mass rises beyond the initial value, preventing the evolution of good solutions that might require traits found only in those behaviour zones. The evolution is much more focused in behaviours that express a monotonic fall of the average distance to the centre of mass, which is consistent with the observable performances of the best controllers. On the other hand, novelty search is not subject to this fitness pressure, and can therefore explore and discover a wide range of solutions to the task.

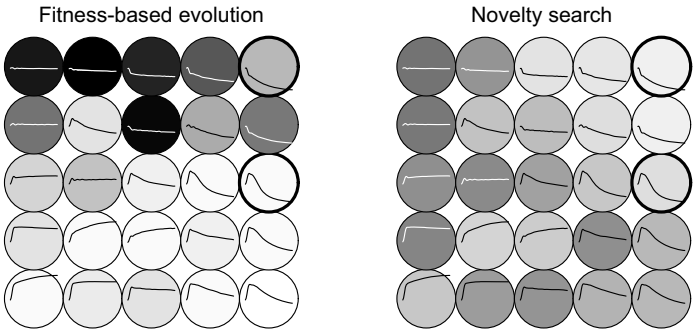

Fig. 1. Kohonen maps representing the explored behaviour space in fitness evolution (left) and in novelty search (right). Each circle is a neuron that is characterised by the vector depicted by the line inside (the average distance to the centre of mass over time). Each behaviour vector is mapped to the most similar neuron. The darker the background of a neuron is, the more behaviours were mapped to it. The neurons corresponding to the best behaviours have a bold circle.

3.3 The Alternative Novelty Measure

We devised a new behaviour description, based on the metric used in [1], in order to determine how the novelty measure influences the evolved solutions. The new description consists of measuring the number of robot clusters along the simulation. Two robots belong to the same cluster if the distance between them is less than 30 cm. Applying this iteratively we can obtain the number of clusters. The number of samples was the same as in our previous experiments (50). The behaviour vector $\mathbf{b_{cl}}$ is described by:

$$\mathbf{b_{cl}} = \frac{1}{N} \left[clustersCount(1), \cdots, clustersCount(T) \right] \quad . \tag{4}$$

The best fitness found in each evolutionary run was 0.83 on average, which is significantly lower (p-value < 0.05) than the novelty search with the centre of mass behaviour measure (0.864 on average). This might be explained by the use of a novelty measure that is less related to the fitness function. But again, we have to look at the evolved behaviours to determine the significant differences. The following distinct successful behaviours were evolved: (1) The robots go towards walls, navigate along it and when they find another one, they form a single file, keeping a fixed distance; (2) They navigate in circles in the environment, forming a static cluster when they meet each other; (3) Similar to (2), but they randomly abandon their respective clusters; (4) They navigate in circles and when two robots meet at some distance, one tries to follow the other. When robots collide, they form a cluster and remain aggregated.

Most behaviours were quite different from the ones found in the previous experiment. The reason the previous experiment did not find these behaviours (and vice-versa) is conflation (see [10]). Conflation occurs when individuals with distinct observable behaviours have very similar behaviour descriptions. The consequence is that an individual with a distinct observable behaviour might not be considered novel by the novelty measure, thus eventually disappearing from the population. Conflation can represent both an advantage because it reduces the search space, and a disadvantage, when different successful solutions or important stepping stones are dismissed. In our experiments, what happens is that the centre of mass novelty measure is conflating some solutions that are not conflated in the clusters measure and vice-versa, thus evolving different solutions in both the experiments.

Two examples of behaviours that can be conflated are shown in Figure 2. When the centre of mass measure is used, for example, the clustering of the robots is irrelevant. The search will therefore avoid behaviours that have an already explored centre of mass progression but differ in the clustering of the robots, possibly bypassing interesting solutions. This effect can also be seen in the evolved behaviours: with the centre of mass measure, there were more solutions that exploited the use of the walls, because navigating near them has a great impact in that novelty measure; while with the number of clusters measure, the solutions focused on the interactions between the agents and clusters, including following each other and leaving the cluster.

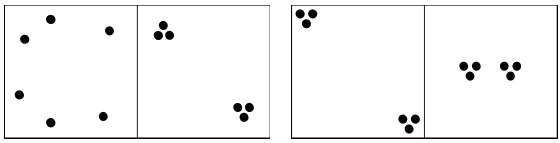

Fig. 2. An illustration of conflation in the centre of mass measure (left) and in the number of clusters measure (right). In both cases, if the robots evolved from the left configuration to the right, that change would not be captured by the respective behaviour description, despite potentially being relevant.

3.4 Combining Novelty Measures

In order to reduce conflation, we setup a new experiment with a richer behaviour description, by combining the novelty measures proposed in the two previous experiments. To combine the two behaviour descriptions presented before in Equations 3 and 4, we simply concatenate the two vectors. But as the novelty measure is based on the Euclidean distance between the vectors, caution must be displayed to ensure that both components have similar contributions to this distance. Namely, we want the vectors to have the same length and the items in the vectors to have the same range, which can be achieved by normalising each of the components. The new behaviour description $\mathbf{b_{comb}}$ is thus defined as:

$$\mathbf{b_{comb}} = (\mathbf{b_{cm}}, \mathbf{b_{cl}}) \ . \tag{5}$$

The fitness performance of the search with this new measure was improved, evolving individuals with high fitness scores much sooner than in the other experiments, as seen in Figure 3. The fitness values in novelty search were higher than fitness-based evolution until generation 150. It is also interesting to look at the explored behaviour space (Figure 4). We can see that there was a greater diversity of solutions, exploring many combinations of the progression of the number of clusters and the distance to the centre of mass. Observing the best controllers in action, we notice that this combined measure evolved all the behaviours that were generated using the previous two measures independently.

To determine why novelty search with the combined measure was faster than fitness-based evolution in finding good individuals, we evaluated the network complexity of the solutions. On average, novelty search finds the first good individual (with fitness value over 0.8) at the generation 33 with a network of 1 hidden neuron and 39 links, while the fitness evolution finds the first good individual at the generation 83 with a network of 4 hidden neurons and 44 links. Looking at the early solutions found by novelty search, we discovered that in some cases they are the ones that the fitness-based evolution could not evolve at all (behaviours that used the wall). In other cases, they were apparently the same solutions that the fitness-based evolution would find in later generations with more complex networks. Due to the incremental nature of NEAT, more complex networks take more generations to evolve. If fitness starts to converge to more complex structures, it takes more time to evolve effective controllers.

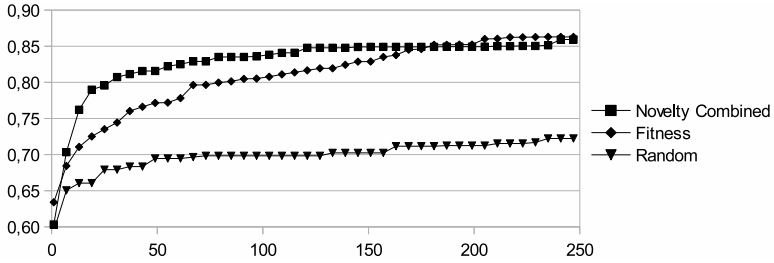

Fig. 3. Fitness value of the best individual found so far in each generation. The values are averaged over 10 evolutionary runs for each experiment. Individuals with fitness value over 0.8 are considered to be solutions to the task. The evolution was tested with more generations but there is no change in the fitness values after the 250th generation.

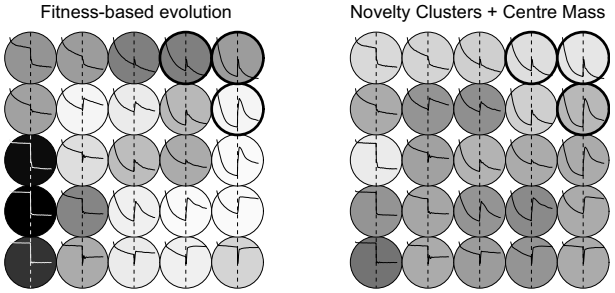

Fig. 4. The explored behaviour space in novelty search with the combined novelty measure and in the fitness-based evolution. In each neuron, the left half is the number of clusters measure and the right half is the centre of mass. The darker the neuron background is, the more behaviours were mapped to it. Neurons with the best behaviours have a bold circle.

4 Discussion

Our experiments revealed that novelty search could outperform fitness-based evolution in respect to the fitness values of the evolved individuals. Other works have shown that novelty search can perform better than fitness-based evolution in deceptive tasks, but fails to be better as the task gets less deceptive [10,13]. Our results suggest that the task is not notably deceptive, as fitness evolution can always find high-scoring solutions. Still, novelty search managed to outperform the fitness-based evolution.

Looking beyond the fitness of the solutions, we showed that the diversity found by novelty search can produce many different solutions to the same task. This can be used to provide a range of different solutions to the experimenter that is using the evolutionary process. This is especially relevant in the swarm robotics domain, because there are many behaviour possibilities and non-obvious relations between the agents that might be revealed. Another advantage of novelty

search was that it found solutions with simpler neural networks than the ones found by fitness evolution, confirming the results reported in [10].

The Kohonen maps proved to be useful in the visualisation of the behaviour search space. They allow the understanding of the behaviour zones that were explored by novelty search and the zones where the fitness-based evolution gets stuck. We verified that controllers mapped to different neurons typically have different observable behaviours. This suggests that analysing the differences in the behaviour vectors might be a way of automatically identifying distinct solutions.

The biggest challenge in using novelty search in the domain of swarm robotics was the definition of the novelty measure. Our experiments suggest that conflation can be a serious issue when evolving collective behaviours with novelty search. While in single robot systems, conflation can be mitigated by describing the full behaviour of the robot, for example its position in space over time [10], in swarm robotics that is not possible. Describing the behaviour of all the robots individually would open the search space too much. It would also introduce scalability issues, for example if the number of robots varies or if the swarm is very large. It is necessary to devise measures that evaluate the swarm as whole. Conflation is essential to cope with the greater diversity of collective behaviours, but caution must be displayed in order not to conflate aspects of the swarm that are relevant to the solution. Our last experiment showed that by combining different novelty measures, we can reduce conflation and improve the performance of novelty search. This combination can simply be the concatenation of the behaviour vectors associated with each measure, which was effective in our case.

5 Conclusion

This study showed that novelty search is a promising technique for evolving controllers for swarm robotic systems. Compared to the fitness-based evolution, novelty search could find a greater diversity of solutions, solutions with higher fitness earlier in the evolution, and solutions based on simpler neural networks. We studied the impact of the novelty measure in the evolved behaviours and showed how conflation can be mitigated by combining different novelty measures. In future research, we will use other novelty search variants that combine the fitness value and the novelty measure [3,9] to investigate if our results can be further improved. We will also use novelty search to evolve controllers for other swarm robotics tasks, to evaluate if the results presented in this paper generalise.

Acknowledgments. This work was supported by FCT project PEst-OE/EEI/LA-0008/2011.

References

1. Bahgeçi, E., Şahin, E.: Evolving aggregation behaviors for swarm robotic systems: A systematic case study. In: Swarm Intelligence Symposium, pp. 333–340. IEEE, New York (2005)

2. Baldassarre, G., Nolfi, S., Parisi, D.: Evolving mobile robots able to display collective behaviors. Artificial Life 9(3), 255–268 (2003)
3. Cuccu, G., Gomez, F.: When Novelty Is Not Enough. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcázar, A.I., Merelo, J.J., Neri, F., Preuss, M., Richter, H., Togelius, J., Yannakakis, G.N. (eds.) EvoApplications 2011, Part I. LNCS, vol. 6624, pp. 234–243. Springer, Heidelberg (2011)
4. Goldberg, D.E.: Simple genetic algorithms and the minimal, deceptive problem. In: Genetic Algorithms and Simulated Annealing. Research Notes in Artificial Intelligence, pp. 74–88. Pitman Publishing, London (1987)
5. Harvey, I., Husbands, P., Cliff, D., et al.: Issues in evolutionary robotics. In: Second Int. Conf. on Simulation of Adaptive Behavior, pp. 364–373. MIT Press, Cambridge (1993)
6. Heaton, J.: Programming Neural Networks with Encog3 in Java. Heaton Research, Chesterfield (2011)
7. Hugues, L., Bredeche, N.: Simbad: An Autonomous Robot Simulation Package for Education and Research. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 831–842. Springer, Heidelberg (2006)
8. Kohonen, T.: The self-organizing map. Proc. of the IEEE 78(9), 1464–1480 (1990)
9. Lehman, J., Stanley, K.O.: Revising the evolutionary computation abstraction: minimal criteria novelty search. In: Genetic and Evolutionary Computation Conf., pp. 103–110. ACM, New York (2010)
10. Lehman, J., Stanley, K.O.: Abandoning objectives: Evolution through the search for novelty alone. Evolutionary Computation 19(2), 189–223 (2011)
11. Lehman, J., Stanley, K.O.: Evolving a diversity of virtual creatures through novelty search and local competition. In: Genetic and Evolutionary Computation Conf., pp. 211–218. ACM, New York (2011)
12. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.C., Floreano, D., Martinoli, A.: The e-puck, a robot designed for education in engineering. In: 9th Conf. on Autonomous Robot Systems and Competitions, pp. 59–65. IPCB, Castelo Branco (2009)
13. Mouret, J.: Novelty-based multiobjectivization. New Horizons in Evolutionary Robotics, pp. 139–154. Springer, Berlin (2011)
14. Risi, S., Vanderbleek, S.D., Hughes, C.E., Stanley, K.O.: How novelty search escapes the deceptive trap of learning to learn. In: Genetic and Evolutionary Computation Conf., pp. 153–160. ACM, New York (2009)
15. Soysal, O., Bahgeçi, E., Şahin, E.: Aggregation in swarm robotic systems: Evolution and probabilistic control. Turkish Journal of Electrical Eng. 15(2), 199–225 (2007)
16. Stanley, K.O.: Efficient Evolution of Neural Networks Through Complexification. Ph.D. thesis, Dep. of Computer Sciences, The University of Texas, Austin (2004)
17. Stanley, K.O., Miikkulainen, R.: Evolving neural network through augmenting topologies. Evolutionary Computation 10(2), 99–127 (2002)
18. Trianni, V., Groß, R., Labella, T.H., Şahin, E., Dorigo, M.: Evolving Aggregation Behaviors in a Swarm of Robots. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) ECAL 2003. LNCS (LNAI), vol. 2801, pp. 865–874. Springer, Heidelberg (2003)
19. Whitley, L.D.: Fundamental principles of deception in genetic search. In: Foundations of Genetic Algorithms, pp. 221–241. Morgan Kaufmann, San Mateo (1991)