Progressive Minimal Criteria Novelty Search

Jorge Gomes[1], 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. We propose progressive minimal criteria novelty search (PMCNS), which is an extension of minimal criteria novelty search. In PMCNS, we combine the respective benefits of novelty search and fitness-based evolution by letting novelty search freely explore new regions of behaviour space as long as the solutions meet a progressively stricter fitness criterion. We evaluate the performance of our approach in the evolution of neurocontrollers for a swarm of robots in a coordination task where robots must share a single charging station. The robots can only survive by periodically recharging their batteries. We compare the performance of PMCNS with (i) minimal criteria novelty search, (ii) pure novelty search, (iii) pure fitness-based evolution, and (iv) with evolutionary search based on a linear blend of novelty and fitness. Our results show that PMCNS outperforms all four approaches. Finally, we analyse how different parameter setting in PMCNS influence the exploration of the behaviour space.

Keywords: Novelty search, evolutionary swarm robotics, deception

1 Introduction

Deception is one of the biggest challenges in evolutionary robotics (ER). Because of deception, some fitness functions misguide the search towards local optima, ultimately resulting in poor solutions to the problem. The more complex the goal task is, the harder it may be to define a non-deceptive fitness function. The interactions between a robot and its environment are often complex, even in simple tasks. Fitness functions in ER are therefore prone to be deceptive. The problem is exacerbated in multirobot systems in which numerous, distributed local interactions can result in distinct self-organised global behaviours.

Recently, Lehman and Stanley [5] proposed a radically different evolutionary approach called *novelty search* (NS). NS searches for novel behaviours regardless of their fitness quality, and thus overcomes deception by ignoring the objective. In NS, behaviours are scored based on how different they are from previously evaluated behaviours. The approach has been successfully applied to many different domains, including evolutionary robotics [6,8,2,10]. Besides avoiding getting stuck in local optima, it was demonstrated that NS is able to find more diverse and less complex solutions, when compared to objective-based evolution [5].

As NS is guided by behavioural innovation alone, its performance can be greatly affected when searching through vast behaviour spaces [4,1], since it may spend most of its time exploring behaviours that are irrelevant for the goal task. To address this problem, Lehman and Stanley [4] proposed *minimal criteria novelty search* (MCNS). MCNS is an extension of NS where individuals must meet some domain-dependent minimal criteria to be selected for reproduction. In [4], the authors applied MCNS in two maze navigation tasks and demonstrated that MCNS evolved solutions more consistently than both novelty and fitness-based search. However, MCNS suffers from two major drawbacks: domain knowledge is required to define suitable minimal criteria; and it may be necessary to bootstrap the search with a genome specifically evolved to satisfy the criteria.

To address the problem of vast behaviour spaces, Cuccu and Gomez [1] proposed to base selection on a linear blend of novelty and fitness (henceforth referred to as *linear blend*). They have applied the approach to the deceptive Tartarus problem, and found that linear blend outperformed both novelty and fitness-based search. Mouret [8] proposed novelty-based multiobjectivisation, which is a Pareto-based multi-objective evolutionary algorithm. A novelty objective is added to the task objective in a multi-objective optimisation. The technique was applied to a deceptive maze navigation problem. Compared with pure novelty search, the multiobjectivization obtained only slightly better results. Other evolutionary techniques that combine behavioural diversity with fitness-guided evolution are presented and compared in [9].

In recent work [2], we successfully applied NS to evolutionary swarm robotics. In this paper, we extend our study by applying variants of NS that combine novelty and fitness, and by studying a different task. We also introduce *progressive minimal criteria novelty search* (PMCNS), which extends MCNS in two ways: (1) PMCNS uses a fitness threshold as the minimal criteria, avoiding the necessity of specifying criteria by hand; (2) starting from the lowest possible fitness score, the criterion is increased dynamically, thereby limiting novelty search to regions of the behaviour space with an increasingly higher fitness. The criterion's monotonous increase depends on the fitness profile of the current population.

We compare the performance of PMCNS against four related methods: NS; fitness-based evolution; linear blend; and MCNS. We use a swarm robotics task in which multiple robots must share a single battery charging station in order to survive. The charging station only has room for one robot and the robots must therefore evolve effective coordination strategies. We use NEAT to evolve the neurocontrollers for the robots in the swarm. NEAT uses speciation to maintain genetic diversity and evolves both the neural network topology and synaptic weights, allowing solutions to become gradually more complex.

2 Background

2.1 Novelty Search

Implementing novelty search [5] requires little change to any evolutionary algorithm aside from replacing the fitness function with a domain dependent novelty metric. The novelty metric measures how different an individual is from other individuals with respect to behaviour. In NS, there is a constant evolutionary pressure towards behavioural innovation. The novelty of an individual is computed with respect to the behaviours of an archive of past novel individuals and to the current population. 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 is above a dynamically computed threshold.

The novelty metric characterises how far the new individual is from the rest of the population and its predecessors in behaviour space, based on the sparseness at the respective point in the behaviour space. A simple measure of sparseness at a point is the average distance to the k-nearest neighbours at that point, where k is a constant empirically determined. Intuitively, if the average distance to a given point's nearest neighbours is large then it is in a sparse area; it is in a dense region if the average distance is small. The sparseness at each point is given by Eq. 1, where μ_i the ith-nearest neighbour of x with respect to the behaviour distance metric $dist$, which typically is the Euclidean distance between domain-dependent behaviour characterisation vectors.

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

Candidates from more sparse regions of this behavioural search space then receive higher novelty scores, thus guiding the search towards what is new, with no other explicit objective.

2.2 Minimal Criteria Novelty Search

Minimal criteria novelty search [4] is an extension of NS that relies on a task-dependent minimal criteria. In MCNS, if an individual satisfies minimal criteria, it is assigned its normal novelty score, as described above. If an individual does not satisfy the minimal criteria, it is assigned a score of zero and is only considered for reproduction if there are no other individuals in the population that meet the criteria. That implies that until an individual is found that satisfies the criteria, search will be random. Therefore, it may be necessary to seed MCNS with a genome specifically evolved to meet the criteria, in case it is unlikely to generate individuals satisfying them in the initial population.

2.3 Linear Blend of Novelty and Fitness

Cuccu and Gomez [1] proposed a linear blend of novelty and fitness score, as a form of sustaining diversity and improving the performance of standard objective search. Their approach constrains and directs the search in the behaviour space. Each individual i is evaluated to measure both fitness, $fit(i)$, and novelty, $nov(i)$, which after being normalised (Eq. 2) are combined according to Eq. 3.

$$\overline{fit}(i) = \frac{fit(i) - fit_{min}}{fit_{max} - fit_{min}}, \quad \overline{nov}(i) = \frac{nov(i) - nov_{min}}{nov_{max} - nov_{min}} \ , \tag{2}$$

$$score(i) = (1 - \rho) \cdot \overline{fit}(i) + \rho \cdot \overline{nov}(i) \ . \tag{3}$$

The parameter ρ controls the relative weight of fitness and novelty, and must be specified by the experimenter through trial and error. fit_{min} and nov_{min} are the lowest fitness and novelty scores in the current population, and fit_{max} and nov_{max} are the corresponding highest scores. The linear blend was applied to the deceptive Tartarus problem, with a large behaviour space, and performance was compared for different values of ρ. The best results were produced with values of ρ between 0.4 and 0.9.

3 Progressive Minimal Criteria Novelty Search

We propose an extension to Minimal Criteria Novelty Search. The objective is to take advantage of the behaviour space restriction provided by MCNS, without having to pre-define domain dependent minimal criteria. In our algorithm, the minimal criterion is a dynamic fitness threshold – individuals with a fitness score greater than the threshold meet the criterion.

Note that although in NS the fitness score does not influence the evolution, typically a fitness function must be specified anyway, in order to be able to identify the best controllers found by NS. In this way, our algorithm does not require the definition of task-specific minimal criteria or any other additional measures. As pre-defining a fixed fitness threshold would raise the same issues as in MCNS (choosing the criteria and bootstrapping the search), we progressively increase the minimal criterion (fitness threshold) during the evolutionary process. The idea behind the increasing fitness criterion is to progressively restrict the search space, to avoid spending much time on the least fit 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, the new criterion is found by determining the value of the P-th percentile of the fitness scores in the current population, i.e., the fitness score below which P percent of the individuals fall. The P-th percentile ($0 \leq P < 100$) of N ordered values is obtained by first calculating the ordinal rank n:

$$n = \frac{P}{100} \times N + \frac{1}{2} \ , \tag{4}$$

rounding the result to the nearest integer, and then taking the value v_n that corresponds to the rank n. Only increases in the minimal criterion are allowed, and in order to smooth out the changes, the minimal criterion from the previous generation is used to compute the criterion for the current generation:

$$mc_g = mc_{g-1} + \max(0, (v_n - mc_{g-1}) \cdot S) \ , \tag{5}$$

where mc is the minimal criterion, and S is the smoothing parameter. The score of each individual in the population is then calculated according to:

$$\text{score}(i) = \begin{cases} nov_i & \text{if } fit_i \geq mc_g \\ 0 & \text{otherwise} \end{cases} \ , \tag{6}$$

where nov_i and fit_i is the novelty score and the fitness score of the individual i, respectively. 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). The smoothing parameter S controls the speed of the adaptation of the minimal criterion (0 – no changes at all, 1 – the value from the previous generation is not considered).

The operation of the novelty archive was not modified, and works as in NS [5]. Even individuals that do not meet the minimal criterion are still added to the repository if their behaviour is sufficiently novel.

4 Experiments

4.1 Setup

The experiments used a resource sharing task, where a swarm of 5 homogeneous robots must coordinate in order to allow each member periodical access to a single battery charging station. The charging station only has room for one robot. To survive, each robot will have to possess several competencies: navigate and avoid walls, find and position itself on the charging station to recharge, and effectively share the common resource with the other robots.

The simulated environment is modelled in a customised version of the Simbad 3d Robot Simulator [3]. The environment is a 4 m by 4 m square arena bounded by walls. The charging station is placed in the centre of the arena. The robots are based on the physical characteristics of the the e-puck educational robot [7], 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) up to a range of 10 cm, and 8 sensors dedicated to the detection of other robots up to a 25 cm range.

Each robot starts with full energy (1500 units) and lose energy over time. In order to charge, the robots must remain still (maintain the same position) inside the charging station, which has the same diameter as a robot. Each robot is additionally equipped with (1) a ring of 8 sensors for the detection of the charging station up to a range of 1 m; (2) a boolean sensor that indicates whether the robot is inside the charging station or not; (3) an internal sensor that reads the current energy level of the robot. If a robot runs out of energy, it stops working, and remains immobile until the end of the simulation.

We test each controller 10 times in varying initial conditions. The set of possible initial positions only includes those from where a robot cannot sense the charging station. Each simulation lasts for 400 s of simulated time.

The controllers of the robots are time recurrent neural networks. The implementation of NEAT used in the evolution is the Java-based NEAT4J (version 1.0).[1] NS was implemented over NEAT following the description and parameters in [5], with a k value of 15 and a dynamic archive threshold [4]. This dynamic threshold ensures a reasonable flow of individuals to the archive (an average

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

rate of 3 individuals per generation). The NEAT parameters are the same for all evolutionary methods: the crossover rate is 25%, the mutation rate 10%, the population size 200, the compatibility threshold is dynamic, targeting 10 species, and each evolution runs for 250 generations. The remaining parameters are set to their default values in the NEAT4J implementation.

We used two slightly different setups in our experiments. In setup A, the robots lose a fixed 10 units of energy per second. In setup B, the robots lose energy proportionally to the power used by their motors, at a rate between 5 and 10 units of energy per second. In both setups, the charging station charges a robot at a rate of 100 units of energy per second.

The fitness function F used to evaluate the controllers is a linear combination of the number of robots alive at the end of the simulation and the average energy of the robots throughout the entire simulation:

$$F = 0.9 \cdot \frac{|a_T|}{N} + 0.1 \cdot \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{e_{i_t}}{TNe_{max}} \quad , \tag{7}$$

where $|a_T|$ is the number of robots alive in the end of the simulation, 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 instant t, and e_{max} is the maximum energy of a robot.

The behaviour characterisation, that is used to compute the behavioural difference in NS and its variants, is closely related to the fitness function. It is composed by just two measures: (1) the number of robots alive at the end of the simulation; and (2) the average energy of the alive robots throughout the simulation. The behaviour characterisation is defined by:

$$\mathbf{b} = \left(\frac{|a_T|}{N}, \sum_{t=1}^{A} \sum_{i \in a_t} \frac{e_{i_t}}{A \cdot |a_t| \cdot e_{max}} \right) \quad , \tag{8}$$

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

4.2 Results

Fitness performance To study how PMCNS and the four related evolutionary methods are influenced by the deceptiveness of the problem, we evaluated and compared their performance in two different setups. Despite being intuitively similar, the setup B leads to deception, while the setup A does not. In setup B, where energy consumption depends on wheel speed, the fitness function is deceptive. It often leads the fitness-based evolution to a very poor local maxima where all the robots remain static, in order to conserve energy and survive more time. Naturally, no one charges and none of the robots reach the end of simulation alive, resulting in a low fitness score. The results can be seen in Figure 1.

In both setups, PMCNS significantly outperforms both fitness-based evolution and NS. PMCNS is also significantly better than linear blend in setup A,

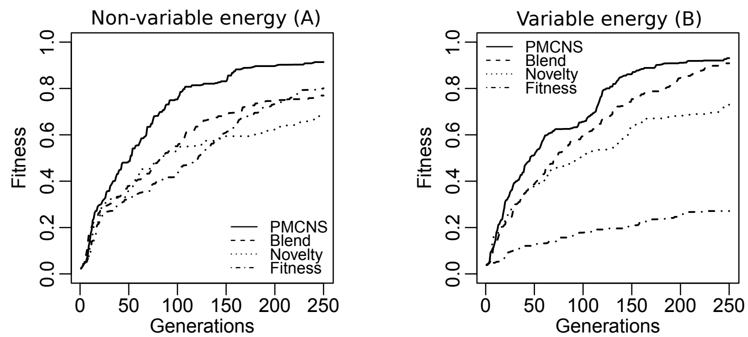

Fig. 1. Average fitness score of the best individual found so far in each generation, with each method. The values are averaged over 10 evolutionary runs for each experiment. Linear blend has $\rho = 0.75$ and PMCNS has $P = 0.5$. Other parameter values were tested but these ones gave the best results.

and can on average achieve high fitness scores sooner than linear blend in setup B. Statistical significance was verified with Student's t-test with $p < 0.05$.

The original MCNS was also tested by defining a fixed fitness threshold as the minimal criterion. Values of 0.03, 0.07, 0.10 and 0.20 were tested for the fitness threshold. Evolution was only able to bootstrap with a fitness threshold of 0.03. In this case, the fitness trajectory was slightly worse than pure NS. With greater fitness thresholds, the evolution could not find individuals with a fitness score that surpassed the threshold, and thus MCNS effectively acted as a random evolution, achieving on average a best fitness of 0.065.

Behaviour space exploration The behaviour space exploration (Figure 2) is similar in PMCNS and linear blend. However, PMCNS clearly has a greater focus in the behaviours with higher fitness scores. It finds behavioural diversity where it is most relevant – in the zones of successful behaviours. It is interesting to note that although PMCNS might be viewed as technique to restrict the search space, it was actually able to find a broader behavioural diversity than NS alone, with respect to the novelty measure used. The explanation for this is that the growing minimal criterion creates a pressure to explore behaviour zones associated with higher fitness – which, in a complex task, are typically the hardest ones to reach. The analysis of the space explored by fitness-based evolution confirms its poor performance. It gets stuck in local maxima with low fitness.

PMCNS could find a broad diversity of successful behaviours (where every robot survives), as it can be seen in Figure 2. There are successful behaviours with average energy ranging from 800 to 1150 units. Observing some of these behaviours confirms this diversity: (1) The robots go towards the charging station and stay there, when another one arrives, the first moves away from the station and returns after a period of time; (2) similar to (1), but they never go farther

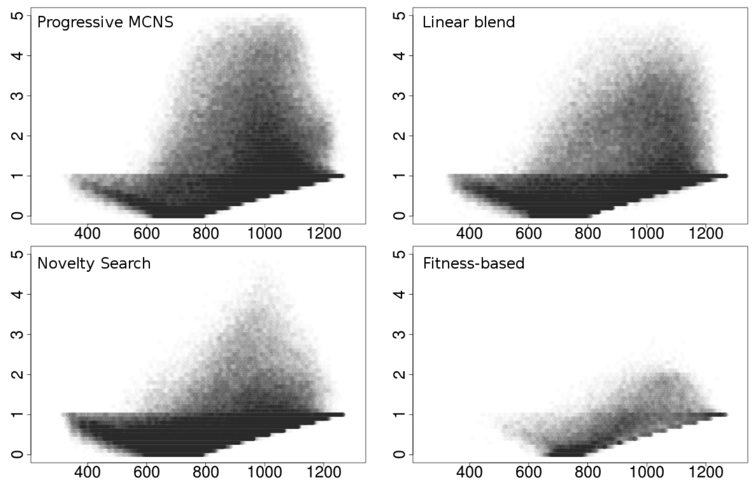

Fig. 2. Behaviour space exploration in setup B. The x-axis is the average energy level of the robots still alive, the y-axis is the number of robots alive at the end of the simulation. Each individual evolved is mapped according to its behaviour. Darker zones mean that there were more individuals evolved with the behaviour of that zone.

than the station sensor range (1 m); (3) the robots go towards the station, circle around it at a very close distance and when their energy level reaches 1000 units, they enter the station to charge and leave when they are full; (4) similar to (3), but they start charging with an energy level below 400 units and only charge until their energy level reaches 1000 units.

Algorithm parameters P is the most important parameter in the PMCNS algorithm. It determines the exigency of the minimal criterion, and consequently, the percentage of population individuals that receive a non-zero score. Three values of P were tested: 25%, 50% and 75%. Figure 3 shows how this parameter affects the fitness trajectory, the progression of the minimal criterion, and the number of individuals that are above the minimal criterion in each generation.

The results show that a high value of P (75%) is prejudicial to the evolutionary process, because the minimal criterion is too strict. Lower values of P are preferred, where only a smaller percentage of the population does not meet the minimal criterion. Analysing the behaviour space explored for each parameter setting, we found that with $P = 50\%$, the search had a greater focus on the high fitness behaviour zones, when compared to the variant with $P = 25\%$. With $P = 75\%$, the search covered a very narrow zone of the behaviour space, and actually explored the high-fitness zones less. A possible explanation for this is that the search got stuck in low-fitness behaviour zones, probably due to the high level of elitism associated with a strict minimal criterion.

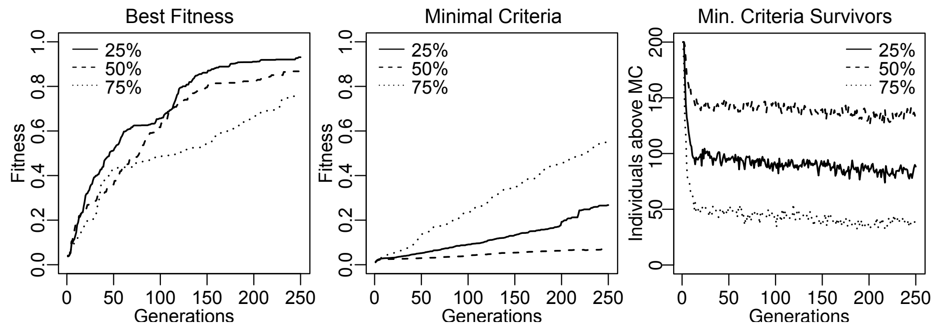

Fig. 3. Left: how the P parameter of PMCNS affects the fitness trajectory. Middle: the progression of the minimal criteria value over the generations. Right: the average number of individuals above the minimal criterion in each generation. Results are from the experiments with setup B.

The smoothing parameter S was set to 0.5 in all experiments. Variations of this parameter within reasonable limits (0.25 – 0.75) did not have a profound impact on the performance of PMCNS.

5 Conclusion

We presented a new method, *progressive minimal criteria novelty search*, for combining fitness and novelty in evolutionary search. We extended *minimal criteria novelty search* by using a dynamic fitness threshold as the minimal criteria, pushing exploration of behaviour space towards zones of higher fitness. We experimented with a swarm robotics task where robots must share a resource in order to survive. We compared the new algorithm with MCNS with a fixed fitness threshold as minimal criteria, novelty search alone, fitness-based evolution, and a linear blend of novelty and fitness scores.

PMCNS could effectively overcome the drawbacks of MCNS while achieving a better performance. The fitness score was successfully used as minimal criterion. It was clearly advantageous to use a progressive minimal criterion, compared to a fixed minimal criterion in MCNS. The bootstrap problem was also overcome, as the minimal criterion starts from the minimum fitness score and only grows if the fitness profile of the current population also increases.

Both PMCNS and linear blend performed significantly better than pure NS and fitness-based evolution in the deceptive setup. In the non-deceptive setup, they were at least as good. Our experiments showed that the fitness performance of NS can be further improved by using the fitness function – even when this fitness function is deceptive. In both PMCNS and linear blend, the behaviour space exploration was greater and more uniform than in NS alone. This result suggests that the fitness function can actually help NS to explore the behaviour space, by creating an additional pressure to explore zones associated with higher fitness, which typically are more difficult to reach in complex tasks.

In terms of fitness trajectory, PMCNS was significantly better than linear blend. PMCNS also explored more the behaviour zones associated with higher fitness scores. This is relevant because it suggests that PMCNS creates a pressure to evolve a diversity of successful individuals. As opposed to the linear blend, where the fitness function is always influencing the score of the individuals, PMCNS only imposes a minimal criterion for selection, and so the fitness function does not have any influence on the score of the individuals, which is based only on the novelty measure.

Novelty search alone displayed a performance similar to the fitness-based evolution in the non-deceptive setup, confirming our previous results [2]. In the deceptive setup, novelty search was clearly superior. It confirms that NS can be used to overcome deception in the swarm robotics domain, even when using relatively simple novelty measures.

References

1. Cuccu, G., Gomez, F.J.: When novelty is not enough. In: European Conf. on the Applications of Evolutionary Computation. LNCS, vol. 6624, pp. 234–243. Springer Verlag, Berlin, Germany (2011)
2. Gomes, J., Urbano, P., Christensen, A.L.: Introducing novelty search to evolutionary swarm robotics. In: 8th International Conference on Swarm Intelligence (in press). Springer Verlag, Berlin, Germany (2012)
3. Hugues, L., Bredeche, N.: Simbad: An autonomous robot simulation package for education and research. In: 9th Int. Conf. on Simulation of Adaptive Behavior. LNCS, vol. 4095, pp. 831–842. Springer Verlag, Berlin, Germany (2006)
4. 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, NY (2010)
5. Lehman, J., Stanley, K.O.: Abandoning objectives: Evolution through the search for novelty alone. Evolutionary Computation 19(2), 189–223 (2011)
6. 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, NY (2011)
7. 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, Portugal (2009)
8. Mouret, J.: Novelty-based multiobjectivization. New Horizons in Evolutionary Robotics, Springer Verlag, Berlin, Germany pp. 139–154 (2011)
9. Mouret, J.B., Doncieux, S.: Encouraging behavioral diversity in evolutionary robotics: An empirical study. Evolutionary Computation 20(1), 91–133 (2012)
10. 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, NY (2009)