Novelty-based Multiobjectivization

Jean-Baptiste Mouret

HAL Id: hal-01300711

https://hal.archives-ouvertes.fr/hal-01300711

Submitted on 11 Apr 2016

Chapter 1
Novelty-based Multiobjectivization

Jean-Baptiste Mouret

Abstract Novelty search is a recent and promising approach to evolve neuro-controllers, especially to drive robots. The main idea is to maximize the novelty of behaviors instead of the efficiency. However, abandoning the efficiency objective(s) may be too radical in many contexts. In this paper, a Pareto-based multi-objective evolutionary algorithm is employed to reconcile novelty search with objective-based optimization by following a multiobjectivization process. Several multiobjectivizations based on behavioral novelty and on behavioral diversity are compared on a maze navigation task. Results show that the bi-objective variant "Novelty + Fitness" is better at fine-tuning behaviors than basic novelty search, while keeping a comparable number of iterations to converge.

1.1 Introduction

Two recent papers [14, 21] introduced a new but radical approach to evolve robot controllers: they propose to maximize the novelty of behaviors *instead of* the efficiency (the fitness) of these behaviors. This method, called "novelty search", relies on a user-defined distance between behaviors and the NEAT [22] evolutionary framework to synthesize neural networks. It could have a least two benefits over traditional fitness-based evolutionary methods: (1) it could override the deceptiveness of some fitness functions and (2) the evolutionary process may be more open-ended.

Abandoning the efficiency objective(s) may be too radical in many contexts and especially in engineering processes – in which *solutions* to a *problem* are

J.-B. Mouret

Université Pierre and Marie Curie – Paris 6, Institut des Systèmes Intelligents et de Robotique (ISIR), CNRS UMR 7222, F-75005, Paris, France

e-mail: mouret@isir.upmc.fr

sought. If the search space is very large, novelty search can run for a very long time before finding any interesting solutions. In such situations, the objective is often the best guide to explore an interesting subset of the search space. Moreover, novelty search can not fine-tune results because the process is not aware that it should stop the exploration and begin a local search. Even if near-optimal solutions are found, the evolutionary pressure is kept towards finding individuals with a significantly different behavior; however, when the evolutionary process has almost found an optimal solution, we can reasonably expect it to cover the last small steps without any difficulty.

In this paper, a Pareto-based Multi-objective Evolutionary Algorithm (MOEA, see [4]) is employed to reconcile novelty search with objective-based optimization. Once defined a novelty objective *and* a fitness objective, the Pareto dominance relation can ensure that novel but inefficient candidate solutions will be selected, as in novelty search, but also that efficient but less novel ones will be considered equally interesting. As it will be demonstrated in this paper, this *multiobjectivization* – the transformation of a single-objective optimization problem to a multiobjective problem – has the potential to overcome the previously described limitations of novelty search.

To cope with the same limitations, an alternative to multiobjectivization is to switch from novelty search to fitness-based search once a criteria is met. We rejected this approach for three main reasons: (1) such a method would require to define an additional and hard to define switch criteria, (2) Pareto-based MOEA are well-studied algorithms that are preferable to custom ones and, (3) a MOEA "automatically" switches back and forth between fitness-based search and novelty search, according the context (in effect, it follows both approaches at the same time, but also all the trade-offs between them). From another point of view, novelty search is an exploration procedure and fitness-based algorithms are an exploitation procedure. Using a MOEA, we thus explicitly model the classic trade-off between exploration and exploitation (see e.g. [23]).

Hence this paper explores one main question: can novelty search be combined with objective-based search using a MOEA? Additionally, we also want to improve the knowledge of novelty search and especially how it compares to diversity maintenance techniques and whether the NEAT encoding is required.

To that aim, we prolong the maze experiments described in the original novelty search paper [14] by comparing them to our multi-objective method. We also compare these results to behavioral diversity [17, 18], another multiobjectivization [13, 11] that uses Pareto dominance to improve the exploration abilities of evolutionary algorithms.

1.2 Related Work

1.2.1 Novelty Search

Novelty search relies on three main concepts:

- a distance between *behaviors* of robots instead of a distance between genotypes as used in classical diversity preservation mechanism [8];
- a novelty measure, based on this distance and an archive of previously encountered behaviors;
- the search for novel behavior instead of the optimization of a fitness function.

Lehman and Stanley [14] measured the novelty $\rho(i)$ of an individual i by computing the mean behavioral distance between i and its k nearest neighbors:

$$\rho(x) = \frac{1}{k} \sum_{j=0}^{k} d_b(x, \mu_j)$$

where k is a user-defined parameter and μ_j is the j-th nearest neighbor of x with respect to the distance d_b. The neighbors are computed using the current population and an archive of all the previous novel individuals. An individual is added to the archive if its novelty is above a minimal threshold ρ_{min}. This last parameter is adjusted dynamically during an experiment:

- if more than 2500 evaluations occurred and no new individual were added in the archive, ρ_{min} is multiplied by 0.95;
- if more than 4 individuals were added during the same generation, ρ_{min} is multiplied by 1.05.

It is important to note that by using both the archive and the population, this nearest neighbors computation combine a diversity preservation mechanism (maximizing the novelty with respect to the current population) and a history-based search. It should also be noticed that, from a computational point of view, the time to find the k nearest neighbors grows continuously (at best logarithmically if a suitable data structure is employed).

1.2.2 Multi-Objective Evolutionary Algorithms

Recent research in evolutionary computation proposed numerous algorithms to simultaneously optimize several conflictual objectives without aggregating them (see [4]). Most of them rely on the concept of Pareto-dominance and generate the so-called Pareto Front:

Definition 1. A solution $\mathbf{x}^{(1)}$ is said to dominate another solution $\mathbf{x}^{(2)}$, if both conditions 1 and 2 are true:

1. the solution $\mathbf{x}^{(1)}$ is not worse than $\mathbf{x}^{(2)}$ with respect to all objectives;
2. the solution $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ with respect to at least one objective.

The non-dominated set of the entire feasible search space is the globally Pareto-optimal set (Pareto front).

MOEA are designed to find an approximation of the Pareto-front, i.e. the set of all the Pareto-optimal trade-offs. Most of the time, they implement two mechanisms:

- a selection procedure that ensures that non-dominated individuals have a selective advantage;
- a diversity-preservation method designed to spread non-dominated individuals along the whole Pareto front, that is to ensure that as many different trade-offs as possible are explored.

A typical MOEA sorts individuals with respect to dominance. Non-dominated individuals may, for instance, be ranked 1, making them the most suitable for reproduction. Individuals which are only dominated by non-dominated ones may be ranked 2, and so on. The diversity on the Pareto front can be maintained using several ways such as fitness sharing [7], crowding [5] or clustering methods [25].

1.2.3 Multiobjectivization

While MOEAs have been primarily designed to optimize several conflicting objectives, a growing trends in the MOEA community is to apply them to improve the optimization of single-objective problems. The term "multiobjectivization" [13, 11, 10] has recently been coined to describe this cast of a single objective problem to a multiobjective one. Two paths can be followed to achieve this reformulation: adding new objectives or decompose the original objective function.

The decomposition of the main fitness function was studied in evolutionary robotics in a few papers, with very encouraging results. In [15] and [16], objectives were designed for each sub-task of a complex problem. This resulted in a fully automatic evolutionary process that can exploit sub-tasks without having to specify their order and when to switch between them. Moreover, several hypothesis about the usefulness of each sub-task were explored in parallel.

The addition of objectives were mostly investigated by adding a genotype-based diversity objective to the fitness [12, 24, 1, 3]. All the published papers about this topic concluded that adding this objective substantially improved the efficiency of evolutionary algorithms. Following this idea, [18], [6] and [17] proposed to employ a behavior-based distance to compute a diversity objective. It had, in particular, been employed to evolve neural networks to

compute a Boolean function [18] and to evolve neuro-controllers for a mobile robot [6, 17]. According to this approach, called *behavioral diversity*, instead of maximizing the fitness $F(i)$, two objectives have to be maximized using a MOEA:

$$\begin{cases} F(i) \\ B(x) = \frac{1}{|P_n|} \sum_{j \in P_n} d_b(x, j) \end{cases} \tag{1.1}$$

where P_n denotes the population at generation n, $d_b(ij)$ the distance between the behaviors of individuals i and j, similar to the one used in novelty search.

1.3 Method

1.3.1 Experiment

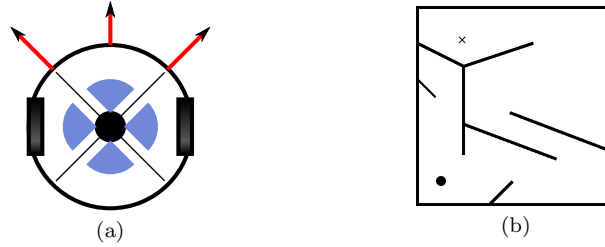(a)   (b)

Fig. 1.1 (a) Overview of the simulated mobile robot. Each arrow outside of the robot's body is a range sensor (e.g. a laser range finder) that returns the normalized distance to the closest obstacle in that direction. The robot is also equipped with four pie-slice sensors that act as a compass towards the goal. This compass can "see" through walls. The robot is moved by two motors, the neural network setting the speed of each of them. (b) Map employed in this set of experiments (size: 512×512 units). The circle denotes the starting position and the cross denotes the goal.

To draw a fair comparaison with the original novelty search, we used the same maze navigation task as in [14]. In this task, a simulated mobile robot has to find a goal zone in a maze by relying only on its three laser range sensors and a goal sensor (see Fig. 1.1(a)).

The robot is controlled by an evolved neural network (the topology and the synaptic weights are evolved) with 7 inputs (3 range sensors and 4 inputs for the goal sensor) and two outputs (the speed of the wheels). To keep our experiments simple and repeatable, we employed a typical graph-based direct encoding for neural-networks in which two kinds of mutations are possible[1]:

---

[1] The NEAT [22] encoding could have been used, as in the original novelty papers. One argument to use NEAT is its incremental nature: the search begins with simple neural

- structural mutation: addition/removal of a random neuron or a random connection;
- parametric mutation: change of a randomly chosen synaptic weight or a neuronal bias; we used here a change in a set of 9 possible values (see appendix).

Cross-over was not employed. This encoding has been used in several previous works [17, 18, 6] in which a basic encoding for neural-networks was needed.

This maze navigation task is interesting because maps with dead ends near the goal can be easily designed. This makes the typical fitness function (the opposite of the distance to the goal) very deceptive. Moreover, we can visualize the part of the maze explored by the robots and thus understand how the search space has been explored.

The map used in this paper (Fig. 1.1(b)) is similar to the "hard map" used by [14], with a very deceitful cul-de-sac in the direction of the goal.

1.3.2 Fitness Function and Distance Between Behaviors

The fitness $F(x)$ of an individual i (to be maximized) is the opposite of the minimum Euclidean distance to the goal obtained during the whole trajectory:

$$F(x) = -\min_{t=0}^{t=T} \left| \left| \mathbf{p_r}(t) - \mathbf{p_g} \right| \right|$$

where $\mathbf{p_r}(t)$ is the position of the robot at time t, $\mathbf{p_g}$ the position of the goal and T the total time of one evaluation.

To describe the behavior of an individual x, we use the position of the robot *at the end of the experiment*:

$$\mathbf{b}(x) = \mathbf{p_r}(T)$$

The distance between individual i and j is then the Euclidean distance between their position at time T:

$$d_b(i,i) = \left| \left| \mathbf{b}(i) - \mathbf{b}(j) \right| \right|$$

Many other distances between behaviors can be designed, such as measuring the difference between trajectories or exploiting the Kolmogorov complex-

---

networks that are complexified during the evolution. This way of exploring topologies well suits the concept on novelty search in which simple behaviors are first explored. However, in this paper we would like to deal with the selective pressure with the simplest encoding as possible. Positive results with a basic graph-based direct encoding would ensure that novelty search does not critically depends on NEAT (which adds noticeably speciation and cross-over)

ity [9]. We will not explore these ideas in this paper in order to match the original novelty search algorithm.

1.3.3 Variants

To evaluate the contribution of a novelty-based multiobjectivization and improve the knowledge about novelty search, we investigate 6 variants of the experiment. Each of them aims at answering a particular question about novelty search.

1. *Control experiment.*
   Question: is this fitness really too deceptive for a classic evolutionary approach?
   $$\text{Maximize } F(x)$$

2. *Novelty search (basic).*
   Question: can we confirm/infirm the published results about novelty search [14] without using NEAT?

   $$\text{Maximize } \rho(x) = \frac{1}{k} \sum_{j=0}^{k} d_b(x, \mu_j)$$

   where μ_j is computed using the archive *and* the population.

3. *Novelty search (no diversity).*
   Question: is it necessary to compute novelty using both the archive and the current population?

   $$\text{Maximize } \rho(x) = \frac{1}{k} \sum_{j=0}^{k} d_b(x, \mu_j)$$

   where μ_j is computed using the archive.

4. *Fitness + Novelty (MOEA).*
   Question: can novelty and fitness be reconciled with a MOEA?

   $$\text{Maximize } \begin{cases} F(x) \\ \rho(x) = \frac{1}{k} \sum_{j=0}^{k} d_b(x, \mu_j) \end{cases}$$

   where μ_j is computed using the archive *and* the population.

5. *Fitness + Diversity (MOEA).*
   Question: how does novelty search compare to multiobjective behavioral diversity [17]?

   $$\text{Maximize } \begin{cases} F(x) \\ B(x) = \frac{1}{|P_n|} \sum_{j \in P_n} d_b(x, j) \end{cases}$$

6. *Fitness + Diversity + Novelty (no diversity) (MOEA).*
   Question: can diversity and novelty be combined with a multi-objective
   approach instead of aggregating them?

$$\text{Maximize} \begin{cases} F(x) \\ B(x) = \frac{1}{|P_n|} \sum_{j \in P_n} d_b(x, j) \\ \rho(x) = \frac{1}{k} \sum_{j=0}^{k} d_b(x, \mu_j) \end{cases}$$

where μ_j is computed using only the archive.

1.3.4 Expected Results

We expect the control experiments not to converge at all because of the deceit-
ful local extrema. Following Lehman and Stanley [14], we can expect novelty
search to work well to go close to the goal. However, it should encounter
some difficulties to exactly reach the goal. "Novelty search + fitness" should
overcome this problem. This last setup should not be substantially faster
than novelty search because the fitness function does not provide a good
guide to reach the goal. Nevertheless, adding the fitness objective to nov-
elty search should not decrease the convergence speed. The three-objectives
approach should be mostly equivalent to the two objectives "fitness + nov-
elty" variant. However, MOEA give better results with two objectives than
three [20, 19], consequently this variant may encounter some difficulties.

Given the good results observed with the addition of a diversity objective
in previous experiments [18, 6, 17], we can reasonably expect a good con-
vergence rate. The convergence speed may be slower because less data are
employed to guide the search.

1.3.5 Experimental parameters

To get statistically significant data, 30 runs of each variant were launched.
NSGA-2 [5], a popular and efficient MOEA, was used to optimize the ob-
jectives. In the single objective case, it is equivalent to a tournament-based
evolutionary algorithm. The detailed parameters of the direct encoding are
available in appendix.

To compute the novelty, we used $k = 15$ (the same value as [14]) and
$\rho_{\min} = 10$. Population size was 200 and each run lasted 1000 generation; put
differently, our budget was of 200000 evaluations.
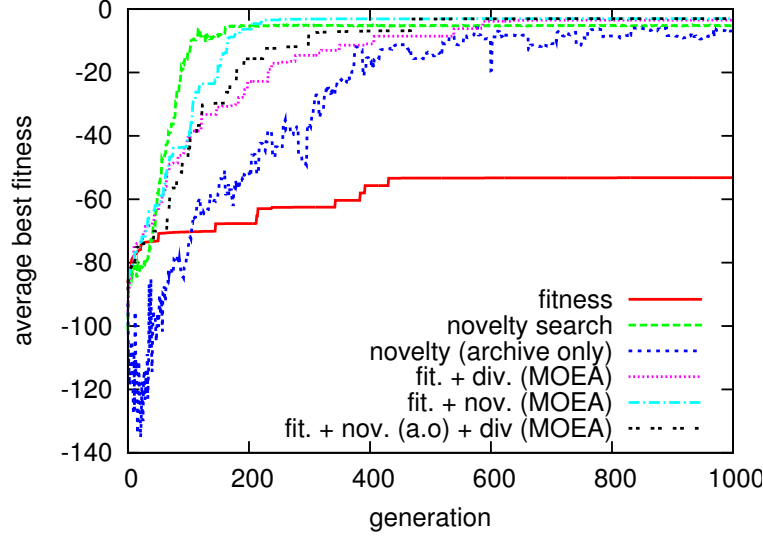
1.4 Results

1.4.1 Average Fitness

Fig. 1.2 Average maximum fitness in the population for each generation and for each variant (30 runs).

Figure 1.3 shows the average (over the 30 runs) best fitness for each variant.

The single objective control experiments converged to a fitness of 60 that corresponds to the distance between the main dead-end and the goal. This means that most runs were trapped in the local maximum (20% of them found a controller that managed to drive the robot near the goal, see section 1.4.2).

Novelty search and multiobjective novelty search ("fitness + novelty") required about 200 generations to get all the runs close to the goal. Novelty search was slightly faster but only the multiobjective variant is able to reach the optimal value. This difference can be easily spotted on Fig. 1.3(b), which displays a zoom on the high fitness values.

If novelty is computed using only the archive, novelty search still converges a lot faster than the control experiment but it requires significantly more generations to converge. Surprisingly, it is also more noisy, suggesting that individuals close to the goal are often lost from the current population in favor of more novel ones. Last, the three objectives approach lead to results
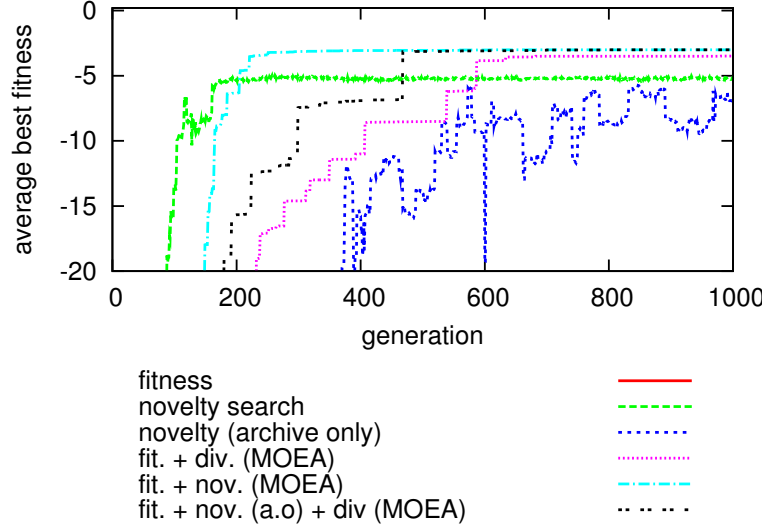
Fig. 1.3 Zoom of figure 1.3 on the best fitness values: the novelty-based multiobjectiviza-tion (fit + nov.) get the optimum fitness while the basic novelty search get close to it but never reach it.

very similar to those obtained with the multiobjectivization "Novelty + Diversity".

1.4.2 Convergence Rate

Considering that a run "converged" when one robot get at less than 10 units to the goal (the optimum is 3 units in these experiments; the choice of 10 is here related to ρ_{min} in novelty search), the convergence rate of each variant can be analyzed (Fig. 1.5(a)). 29% of the control experiments (fitness only) converged in spite of the attractive local maximum. Nevertheless, 100% of the other experiments converged in less than 1000 generations (200000 evaluations). Novelty search and multiobjective novelty search are the fastest to converge while substantially more generations are needed for the novelty search variant that uses only the archive.

If we count as converged only the experiments that reach the optimum fitness (-3, Fig. 1.5(b)), only the novelty-based multiobjectivization runs get a 100% convergence rate whereas 90% of the novelty search runs converged.

To understand if the observed difference are statistically significant, the average generation to converge ($F(i) > -10$) has been computed (table 1.1). Unpaired Student T-tests (table 1.1) show that these results are statistically
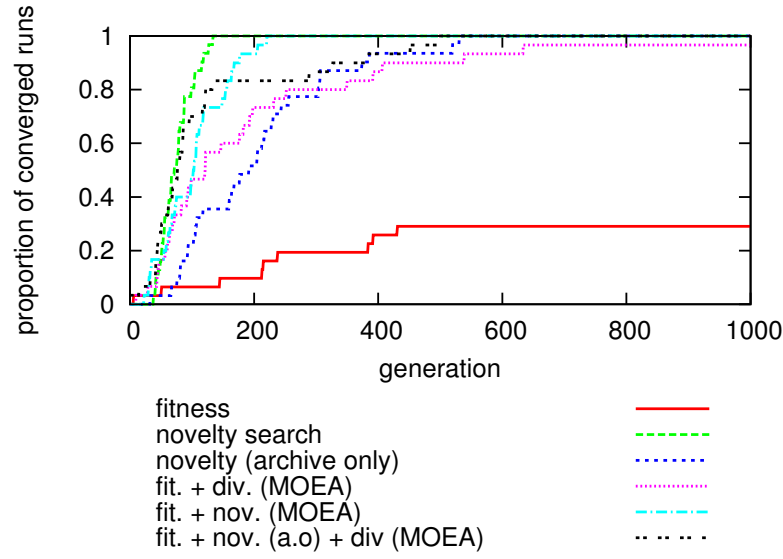
Fig. 1.4 Proportion of converged runs with regards to the generation number when considering that a run has converged when the best individual get at less than 10 units to the goal.
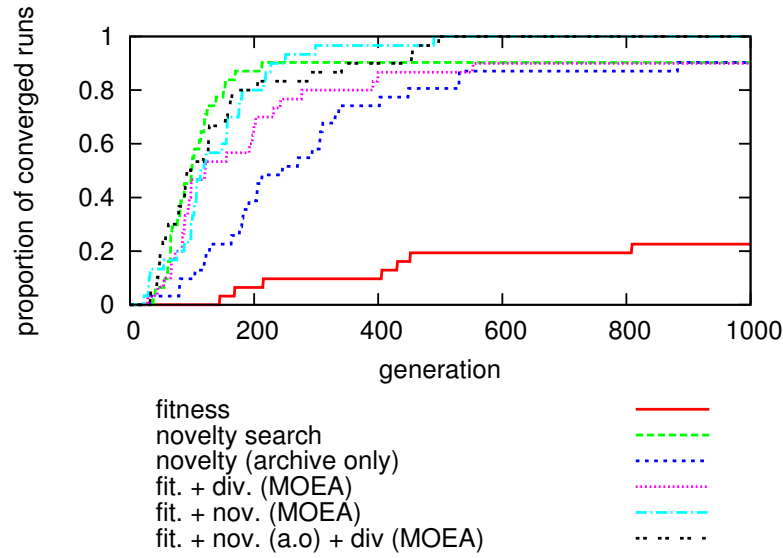

Fig. 1.5 Proportion of converged runs with regards to the generation number when considering that run has converged when the best individual get at less than 3 units to the goal (the optimum).
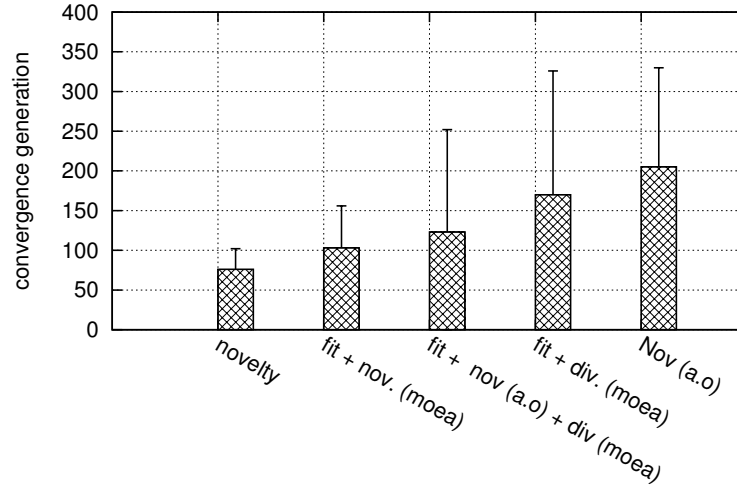
Fig. 1.6 Average generation to converge (i.e. $F(i) > -10$). Each generation is 200 evaluations.

	Nov.	fit.+nov.	fit.+div.	Nov.(a.o)	f.+n.+d.
Aver. gen.	76	103	170	205	123
Std. dev.	26	53	156	125	129
T-test					
nov.	$p = 1.0$	$p = 0.01$	$p < 0.01$	$p < 0.01$	$p = 0.05$
fit.+nov.	$p = 0.01$	$p = 1.0$	$p = 0.03$	$p < 0.01$	$p = 0.34$
fit.+div.	$p < 0.01$	$p = 0.03$	$p = 1.0$	$p = 0.34$	$p = 0.20$
nov.(a.o.)	$p < 0.01$	$p < 0.01$	$p = 0.34$	$p = 1.0$	$p < 0.02$
f.+n.+d.	$p = 0.05$	$p = 0.34$	$p = 0.20$	$p < 0.015$	$p = 1.0$

Table 1.1 Average generation to converge (i.e. $F(i) > -10$) and results of Student's T-test. Each generation is 200 evaluations.

significant except between the "diversity + fitness" and the "novelty (archive only)" variants and between "fitness + diversity + novelty" and most of the other variants. Despite a small difference in average generation of convergence (25 generations/500 evaluations), novelty search appears statistically faster than the multiobjective variant. This small difference is easily understood by taking into account that, in the multiobjective variant, a part of the population is kept only because of its good fitness; the deceptiveness of the fitness makes these individuals almost useless to the algorithm. This drawback may be compensated by the better fitness obtained with the multiobjective variant (Fig. 1.3(b)). Results obtained with novelty search and multiobjective novelty search are of the same order of magnitude when they are compared to the "fitness only" runs: in both cases, only a few generations are required

to converge. A more complex problem, with a larger search space, could allow to draw a real efficiency distinction between these two approaches.

The diversity multiobjectivization was significantly slower to converge than the novelty variants. However, its computational cost is substantially lower than novelty search. To compute the diversity, only N^2 distances (N being the size of the population) need to be computed. With novelty search, the archive is an order of magnitude larger than the population (in our experiments, the typical archive size at the end of the experiments was about 1500 whereas the population size was 200).

1.4.3 Exploration

Figure 1.7 depicts the position at the end of the experiment of each evaluated until the first one reaches the goal. The first surprising observation is that, in all runs (successful or not), only the bottom left part of the maze has been explored. This can be explained by understanding where lies the difficulty of this task; two behaviors are the key to reach the goal: turning towards the right at the beginning of the experiment and efficiently avoiding obstacles. The turn of the bottom right is difficult to negotiate so if an individual manages to overcome this part of the maze, it is likely able to also come close to the goal. Since the fitness rewards the minimum distance to the goal during the whole experiment, robots don't have to adjust precisely their trajectory to get above the convergence threshold (10 units), they only have to wander near the goal.

The second surprising observation is that the explored points are not as different as suggested by the convergence rates. The points may be more evenly distributed when a novelty objective is used but the difference is not obvious. The multiobjectivization "fitness + novelty" (Fig. 1.7(b)) put a clear emphasis on the top left part of the maze, near the goal but in the dead-end, especially if it is compared to the original novelty search (Fig. 1.7(e)). However, it seems hard to predict which run will succeed using only the explored points.

This similarity may be interpreted in term of selective pressure: the key to converge is not to explore the good points but to recognize which point could lead to the solution. In the control experiment (Fig. 1.7(a), the robots that end at the bottom right part of the maze will be ranked last, because they are far away from the goal. However, with a novelty objective, the same behavior will be ranked first because there is almost no previous behaviors in this area. One may wonder why there is not many points in this part of the maze in the novelty experiments compared to the left part. A possible explanation is that such behaviors are hard to obtain, because a small variation of an individual with this behavior may more likely drive the robot in a wall near the starting point than in the same area. Consequently, it is only
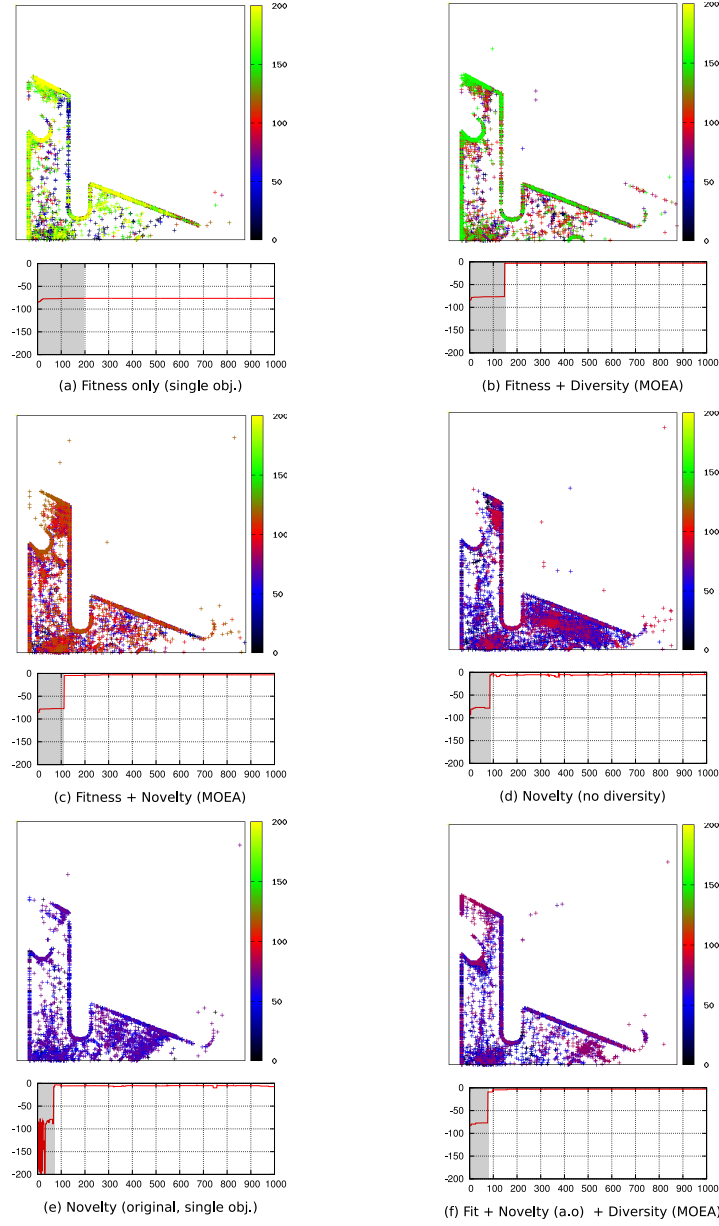
Fig. 1.7 (top) Position of each individual present at each generation until the first one reaches the goal (i.e. the robot is at less than 10 units from the goal), for a typical run of each variant. The control experiment (fitness only) did not converge so the first 200 generations are displayed. The color of a point denotes the last generation the corresponding individual has been part of the population. (bottom) Maximum fitness with regards to the generation number. The gray zone corresponds to the number of generations displayed in the top diagram.

by persistently trying to modify the bottom right individuals that a successful individual can emerge. To test this hypothesis, one should evaluate how many children of such individuals reach the same area as their parents.

1.5 Conclusion and Discussion

This paper investigated several ways to combine novelty search and objective-based search using a multiobjectivization. These methods were compared to behavioral diversity, a multiobjectivization that is also based on a distance between behaviors of candidate solutions.

The experimental results show at least that:

- novelty search is well adapted to the investigated deceptive task (this confirms the previously published results [14]);
- adding a fitness objective to novelty search does not harm the efficiency of the search;
- adding a fitness objective to the novelty objective allows to fine-tune the result more efficiently than with the original novelty search algorithm;
- diversity-based multiobjectivization requires more generations to converge but also lead to successful results in a few hundred of generations, whereas its computational cost is lower than novelty search;
- a straight-forward graph-based encoding leads to similar results than NEAT in this setup (at least qualitatively).

Overall, the successful experiments suggest that the difficulties encountered to evolve neural networks to solve complex tasks, especially observed in robotics, may currently be more a problem of good selective pressures than of bad encodings.

A more complex task, with a larger search space and/or with a less obvious dead-end, is needed to further compare novelty search to the multiobjectivizations introduced in this paper. The guide provided by the fitness in the maze experiment described here is deceptive by design but the success of evolutionary algorithms proves that fitness can be a good guide towards efficient solutions. We believe that the proposed multiobjectivization can combine the best of both worlds by using fitness when it is a good guide and novelty search when it is not.

The computational cost of novelty search compared to the one of behavioral diversity is a significant drawback of novelty search. One of the first idea to reduce the algorithmic complexity of novelty search is to efficiently structure the archive to compute the nearest neighbors as fast as possible. The typical structure used in computer graphics to this task is the kd-tree [2]; future work should investigate this idea. Another approach is to keep the size of the archive bounded by periodically removing the least significant elements.

References

1. Abbass, H.A., Deb, K.: Searching under multi-evolutionary pressures. In: Proceedings of the Fourth Conference on Evolutionary Multi-Criterion Optimization. Springer (2003)
2. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communication of the ACM **18**(9), 509–517 (1975)
3. Bui, L., Abbass, H.A., Branke, J.: Multiobjective optimization for dynamic environments. Proceedings of the IEEE-CEC **3**, 2349–2356 Vol. 3 (2005)
4. Deb, K.: Multi-objectives optimization using evolutionnary algorithms. Wiley (2001)
5. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Proceedings of PPSN VI pp. 849–858 (2000)
6. Doncieux, S., Mouret, J.B.: Single step evolution of robot controllers for sequential tasks. In: Proceedings of GECCO'09. ACM (2009)
7. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Proceedings of the Fourth International Conference on Evolutionary Programming, pp. 416–423 (1993)
8. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the Second International Conference on Genetic Algorithms, pp. 148–154. Morgan Kaufmann (1987)
9. Gomez, F.: Sustaining diversity using behavioral information distance. In: Proceedings of GECCO'09 (2009)
10. Greiner, D., Emperador, J., Winter, G., Galván, B.: Improving Computational Mechanics Optimum Design Using Helper Objectives: An Application in Frame Bar Structures. LNCS **4403**, 575 (2007)
11. Handl, J., Lovell, S.C., Knowles, J.: Multiobjectivization by decomposition of scalar cost functions. In: Proceedings of PPSN X, pp. 31–40. Springer (2008)
12. de Jong, E.D., Watson, R.A., Pollack, J.B.: Reducing bloat and promoting diversity using multi-objective methods. In: Proceedings of GECCO'01, pp. 11–18 (2001)
13. Knowles, J.D., Watson, R.A., Corne, D.W.: Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In: First International Conference on Evolutionary Multi-Criterion Optimization, pp. 268–282. Springer (2001)
14. Lehman, J., Stanley, K.: Exploiting open-endedness to solve problems through the search for novelty. In: Proceedings of Artificial Life XI, pp. 329–336 (2008)
15. Mouret, J.B., Doncieux, S.: Incremental Evolution of Animats Behaviors as a Multi-objective Optimization. In: From Animals to Animats 10, *LNCS*, vol. 4095/2006 (2008)
16. Mouret, J.B., Doncieux, S.: Evolving modular neural-networks through exaptation. In: Proceedings of IEEE Congress on Evolutionary Computation (IEEE-CEC) (2009)
17. Mouret, J.B., Doncieux, S.: Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In: Proceedings of IEEE-CEC (2009)
18. Mouret, J.B., Doncieux, S.: Using behavioral exploration objectives to solve deceptive problems in neuro-evolution. In: Proceedings of GECCO'09 (2009)
19. Praditwong, K., Yao, X.: How well do multi-objective evolutionary algorithms scale to large problems. In: Proceedings of IEEE-CEC, pp. 3959–3966 (2007)

20. Purshouse, R.C., Fleming, P.J.: On the Evolutionary Optimization of Many Conflicting Objectives. IEEE Transactions on Evolutionary Computation **11**(6), 770–784 (2007)
21. Risi, S., Vanderbleek, S.D., Hughes, C.E., Stanley, K.O.: How novelty search escapes the deceptive trap of learning to learn. In: Proceedings of GECCO'09 (2009)
22. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation **102**(2), 99–127 (2002)
23. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)
24. Toffolo, A., Benini, E.: Genetic diversity as an objective in multi-objective evolutionary algorithms. Evolutionary Computation **11**(2), 151–167 (2003)
25. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), pp. 95–100 (2001)

Parameters and Source Code

- MOEA: NSGA-II
- Population size: 400
- Neural network (direct encoding):

  – available weights / bias:
    $\{-2.0, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0\}$
  – proba. of weight/bias change (for each value): 0.1
  – number of inputs/outputs: 7/2
  – min./max. number of neurons (random generation): 10/20
  – min./max. number connections (random generation): 20/45
  – probability of adding/deleting a connection: 0.15/0.25
  – probability of changing a connnection: 0.1
  – probability of adding/deleting a neuron: 0.025/0.025
  – activation function for neurons:
    $y_i = \varphi\left(\sum_j w_{ij} x_j\right)$ where $\varphi(x) = \frac{1}{1+\exp(b-7x)}$

- source code: `http://www.isir.fr/evorob_db`