



**HAL**  
open science

# Behavioral Diversity with Multiple Behavioral Distances

Stéphane Doncieux, Jean-Baptiste Mouret

► **To cite this version:**

Stéphane Doncieux, Jean-Baptiste Mouret. Behavioral Diversity with Multiple Behavioral Distances. IEEE Congress on Evolutionary Computation, 2013 (CEC 2013), 2013, Cancun, Mexico. pp.1-8. hal-01300703

**HAL Id: hal-01300703**

**<https://hal.science/hal-01300703v1>**

Submitted on 11 Apr 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Behavioral Diversity with Multiple Behavioral Distances

Stephane Doncieux

ISIR, Université Pierre et Marie Curie-Paris 6,  
CNRS UMR 7222  
4 place Jussieu, F-75252, Paris Cedex 05, France  
Email: doncieux@isir.upmc.fr

Jean-Baptiste Mouret

ISIR, Université Pierre et Marie Curie-Paris 6,  
CNRS UMR 7222  
4 place Jussieu, F-75252, Paris Cedex 05, France  
Email: mouret@isir.upmc.fr

**Abstract**—Recent results in evolutionary robotics show that explicitly encouraging the behavioral diversity of candidate solutions drastically improves the convergence of many experiments. The performance of this technique depends, however, on the choice of a behavioral similarity measure (BSM). Here we propose that the experimenter does not actually need to choose: provided that several similarity measures are conceivable, using them all could lead to better results than choosing a single one. Values computed by several BSM can be averaged, which is computationally expensive because it requires the computation of all the BSM at each generation, or randomly switched at a user-chosen frequency, which is a cheaper alternative. We compare these two approaches in two experimental setups – a ball collecting task and hexapod locomotion – with five different BSMs. Results show that (1) using several BSM in a single run increases the performance while avoiding the need to choose the most appropriate BSM and (2) switching between BSMs leads to better results than taking the mean behavioral diversity, while requiring less computational power.

## I. INTRODUCTION

Evolutionary Robotics (ER) aims at designing robot controllers and, optionally, morphologies to reach a particular behavior whose efficiency is described by a fitness function [9], [8]. This function is then optimized thanks to an Evolutionary Algorithm (EA) that explores a search space of parameters, e.g.  $\mathbb{R}^n$ , or structures, e.g. neural networks topologies. Discovering interesting solutions implies to balance exploration – to avoid local optima – and exploitation – to get the nearest optimal solution.

Preserving diversity in the population is known to be an efficient way to deal with a natural tendency of EAs to favor exploitation. In classic EA, this diversity preservation is computed in the space of genotypes, phenotypes or fitness values [10], [16]. In ER, it has been widely observed that many genotypes and phenotypes lead to similar behaviors, whereas a very small difference in a genotype can substantially modify the behavior of the robot[20]. Since ER is ultimately seeking behaviors, the diversity that appears to matter is the diversity in behavior space. In accordance with this intuition, explicitly encouraging diversity in this space revealed to drastically improve convergence of ER experiments in multiple contexts [11], [19], [18], [20].

Behavioral diversity methods foster diversity by rewarding individuals whose behavior is different from the rest of the population. The similarity is computed using behavior

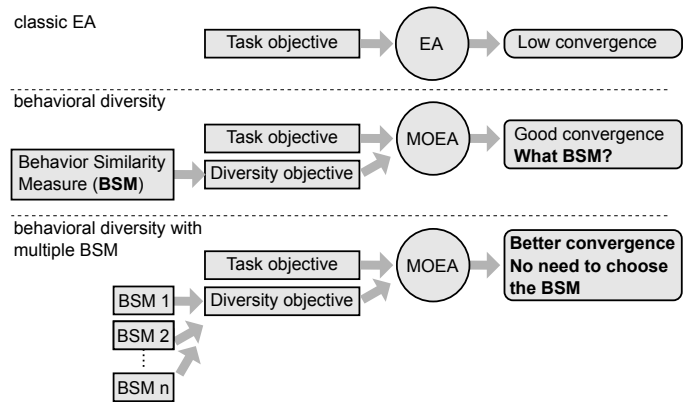


Fig. 1. Main proposition. Encouraging behavioral diversity drastically improves the convergence of evolutionary robotics experiments[20], but it requires defining a Behavior Similarity Measure (BSM). Using many BSM avoids the need to choose and leads to similar or better convergence rates.

similarity measures (BSM) that can be problem-specific [18], [15], [20], [21] or designed to be used on a large class of robotic tasks [7], [25], [21], [11], [20]. Empirical comparisons between BSM have been made in several papers [11], [20], but, although different BSM yielded different results, no measure has proved to be “the one” for every contexts. The present work investigates another approach: if the best BSM is not known, *is it possible to efficiently exploit all the different BSM an experimenter can think of without having to choose among them?*

Besides removing the need to choose, relying on several BSM instead of one may also increase the impact of behavioral diversity. A BSM is not perfect when it fails to recognize new behaviors that could substantially increase the best fitness on the long term. Using several different BSM may then lead to a better exploration as their criteria for recognizing a behavior as new and different are not the same. More behaviors will then be recognized as new along a run, leading to a better exploration.

How to take into account several BSM? Two different approaches have been compared. In the first approach, the behavioral diversity objective is simply the average value of the behavioral diversities associated to each BSM<sup>1</sup>. In the second

<sup>1</sup>Each behavioral diversity is normalized by its maximum value in the current population.

approach, one BSM is used at a time. At a given a period, a new BSM is randomly chosen out of the available BSM and used to compute the behavioral diversity. This second approach decreases the computational load as only one BSM is used at a time. The sensitivity of the results to the switching period is studied.

Hence, this article aims to validate the following hypotheses:

- 1) using several BSM instead of a single one doesn't decrease performance;
- 2) a unique BSM does not capture and appropriately reward every original behavior;
- 3) using several BSM gives statistically more chance to original behaviors to be rewarded during an evolutionary run.

Using several BSM may seem at first sight to be a practical limitation of this approach compared to single BSM approaches because it requires more work to prepare an experiment. In practice it reveals to be easy to find different BSM and quite difficult to choose the one to be used. The only way to do it now is to make some preliminary experiments to evaluate the relative efficiency of each candidate BSM. Not having to choose between different alternatives therefore substantially decreases the amount of preparation for an experiment.

Two setups have been considered: a ball collecting task [21] and an hexapod walking task.

## II. RELATED WORK

While most EA have diversity preserving mechanisms based on the genotype or fitness space, the introduction in ER of a behavior space, may be exploited in this context to foster exploration. Three different approaches dedicated to ER can be identified: novelty search [15], behavior speciation [25] and behavioral diversity [20]. Novelty search aims at abandoning any task-related objective to reward behavioral novelty only [15]. Behavior speciation subdivides the population in several sub-populations on the basis of individuals behavior, thus replacing a global competition by a local competition between individuals with similar behaviors [25]. Lastly, behavioral diversity defines an helper objective that evaluates the difference in terms of behavior of an individual relative to the current population. This objective is then maximized in a multi-objective scheme together with goal related objectives [20]. All these methods rely on comparisons of behaviors.

How to compare behaviors? A robot behavior results from its interaction with the environment. Different behavior descriptors can be used:

- robot trajectory [7], [25], [21]
- robot perceptions and/or actions [11], [20]
- adhoc measures depending on the problem to solve [18], [15], [20], [21].

Once such descriptors are available, and depending on their features, a similarity can be computed with different kinds of distances, like for instance an Euclidean distance [18], [15], a hamming distance [11], [7], [20] or an edit distance [25], [21].

Few comparisons have been performed on the choice of the descriptor and distance to use [11], [7], [20]. They highlighted that this choice has a significant impact on the performance and that performing a hamming distance on a temporal sequence of discretized robot actions – and eventually perceptions – gives an overall good performance even if has failed in several setups [20].

Some features of BSM, like, for instance, the adequacy to human classification [7], were also investigated in an attempt to characterize efficient BSM, but such studies did not lead to any clear conclusion yet. Up to now, it is therefore hard to know whether a single BSM will outperform other BSM in any context or if some method can be built to design an efficient, if not optimal, BSM for a particular problem. The rationale of the present work is different: if several BSM are known and no criteria to decide which one is the best, is it possible to use all of them instead of just one?

## III. MULTI-DIST BEHAVIORAL DIVERSITY

Following the results of [4], [1], [20], a multi-objective diversity mechanism has been used. The general framework of this approach is to add a *helper* objective in addition to the objectives rewarding the performance on the task [12]. The helper objective measures the mean distance of the evaluated solution to the rest of the population on the basis of *behavior* comparisons [19], [18], [20].

Instead of using a single BSM during the whole run, the proposed methods consists in using several BSM. The first method consists in averaging the behavioral diversities computed with each BSM. The mean behavioral diversity objective is then defined as follows:

$$BD_{mean}(x) = \frac{1}{M} \sum_{k=0}^{M-1} \left( \frac{1}{N} \sum_{y \in P} \frac{d_k(x, y)}{\max_{a, b \in P} (d_k(a, b))} \right) \quad (1)$$

where  $P$  is the current population and then  $x, y, a, b$  individuals in the population.  $N$  the size of the population and  $d_k(x, y)$  a BSM between  $x$  and  $y$  ( $k \in [0, M - 1]$ ),  $M$  being the number of available BSM.

The second method consists in changing the BSM used along the generations. For each generation, a single BSM is used, but at a given generation period  $p_g$ , the BSM is changed and chosen out of a set of  $M$  BSM.

$$BD_{switch}(x) = \frac{1}{N} \sum_{y \in P} d_k(x, y) \quad (2)$$

where  $k$  the index of the BSM to be used at the current generation,  $k \in [0, M - 1]$ . Every  $p_g$  generation, a new value of  $k$  is chosen out of a uniform distribution over  $[0, M - 1]$ :

$$k = rand(M) \quad (3)$$

A typical experiment consists then in maximizing the following objectives:

$$\text{Maximize } \begin{cases} F(x) \\ BD_{mean}(x) \text{ or } BD_{switch}(x) \end{cases} \quad (4)$$

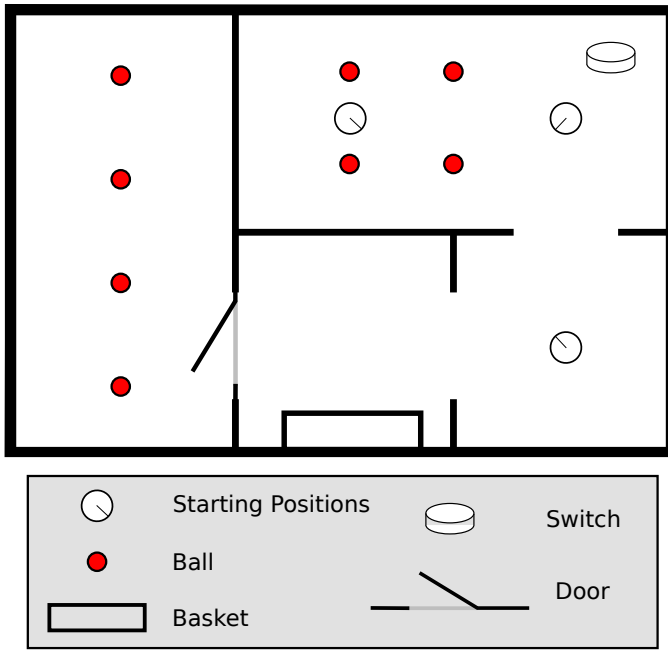


Fig. 2. Overview of the arena and of the robot. The goal of the experiment is to place as many balls as possible into the basket. A robot controller is evaluated three times at three different initial positions. To reach the four balls in the left room, the robot has to open the door by first pressing the switch button.

An alternative to this approach may be to simultaneously use several behavioral diversity objectives, each of them relying a different BSM. Although it may in theory lead to the similar results, the performance in practice of multi-objective EA quickly decreases when the number of objectives grows [23]. The design of the proposed multi-dist behavioral diversity objectives takes it into account. Furthermore, using several diversity helper objectives would reduce the selection pressure exerted by  $F(\cdot)$ .

In both experiments, NSGA-II [5], a state-of-the-art multi-objective evolutionary algorithm, has been used with a population size of 200. When only one objective is optimized, NSGA-II is a classic elitist evolutionary algorithm with a tournament-based selection. The best-of-run individual is the individual with the best  $F(\cdot)$  value<sup>2</sup>. The source code of both experiments is available at [http://pages.isir.upmc.fr/evorob\\_db](http://pages.isir.upmc.fr/evorob_db).

#### IV. BALL COLLECTING TASK

##### A. Experimental setup

The goal of this task is to collect balls in an arena and put them into a basket (figure 2). The task has been first described in [21]. The arena contains four balls and a switch button that allows the robot to open the door of a room containing four more balls. The robot has two wheels and twelve sensors: three wall distance sensors, two bumpers, two ball detection sensors, two switch sensors, two basket detection sensors and one "carrying ball" sensor. It has three effectors: left and right motors as well as an "action" effector. A nearby ball is

<sup>2</sup>the behavioral diversity is used as an helper objective during an optimization. It can be discarded once the results have been generated.

collected or the switch is activated if the value of the "action" effector value is above 0.5. If it is below 0.5, a carried ball is released. A released ball disappears from the arena. If, at that time, the robot was in front of the basket and touching it, then the ball is considered to be collected.

The fitness function is the number of collected balls divided by the maximum number of balls available. The performance is evaluated in three different contexts (with different starting positions, as shown on figure 2). The maximum number of balls is then 24 and a fitness of 0.5 corresponds to 12 collected balls.

Four different BSM have been considered:

- adhoc [21]: the behavior of a robot is described by the final position of the balls, i.e. the initial position of balls that have not been moved and the last position of the robot if it was still carrying the ball at the end of the run or else the position of the robot when it has released the ball. The behavior descriptor is the vector of ball positions and the BSM is the Euclidian distance between these vectors;
- hamming [20]: the value of each sensor and effector is discretized and memorized during 4000 time steps. It results in a binary vector of 60000 bits ( $4000 \times (12 \text{ sensors} + 3 \text{ effectors})$ ). The BSM is then the Hamming distance that measures the number of bits that differ between the two vectors;
- trajectory [21]: the discretized position of the robot is recorded each 50 time-steps. This vector is the behavior descriptor. The BSM is then the edit distance between the two vectors [24];
- entropy: the entropy is computed for each sensor and effector. For binary sensors, i.e. bumpers, balls, basket and switch sensors, the entropy has been computed as follows:

$$E = - \sum_{i=0,1} p_i \frac{\ln(p_i)}{\ln(2)}$$

with  $p_i$  the probability that the corresponding sensor takes the value  $i$  during the evaluation. For sensors and effectors with a continuous value, the range of possible values has been divided in ten different subranges and the entropy has been computed as follows:

$$E = - \sum_{i=1}^{10} p_i \frac{\ln(p_i)}{\ln(10)}$$

with  $p_i$  the probability that the sensor or effector value belongs to the  $i$ -th subrange. The behavior descriptor is then the vector of the entropies of each sensors and effectors. The corresponding BSM is an Euclidean distance between those vectors;

The robot is controlled by a neural network with twelve inputs and three outputs. Both structure and parameters of the neural network are evolved with DNN [20]. DNN (Dynamic Neural Network) is a simple direct encoding with no crossover. Mutations can add or remove neurons and connections and change connection weights. As in NEAT (NeuroEvolution of Augmenting Topologies) [22], DNN starts with a perceptron

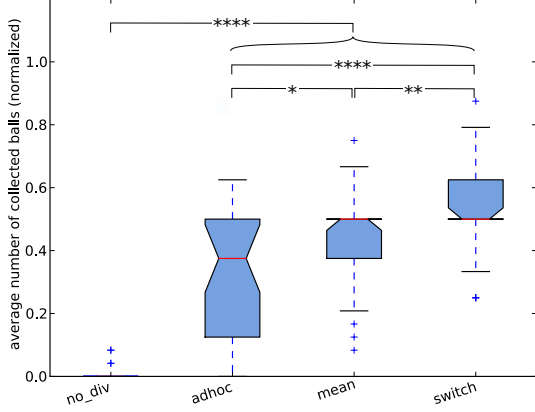


Fig. 3. Number of collected balls for 30 runs after 4000 generations and for the treatments without the diversity objective (no\_div), with the most performing single BSM diversity objective (adhoc) and with the proposed  $BD_{mean}$  and  $BD_{switch}$  ( $p_g = 50$  generations). The statistical significance is represented with the horizontal lines: '\*\*\*\*' means  $p$ -value  $< 10^{-4}$ , '\*\*\*' means  $p$ -value  $< 10^{-3}$ , '\*\*'  $p$ -value  $< 10^{-2}$  and last '\*'  $p$ -value  $< 0.05$ . No horizontal line between two treatments means no statistical difference in the results. The boxes extend from the lower to upper quartile values of the data, with a line at the median. The whiskers show the range of the data.

with no hidden layer. DNN parameters used for these experiments are listed in the appendix.

## B. Experiments and results

a) *Main results*: Figure 3 shows the number of collected balls for each of the 30 runs and for the treatments without diversity (no\_div), with a single BSM (adhoc) and with the proposed multi-dist behavioral diversity objectives that use all of the four BSM previously defined. Both the  $BD_{mean}$  and  $BD_{switch}$  outperform the runs without diversity and also the runs relying on a single BSM<sup>3</sup> ( $p$ -value  $< 1e-4$  in the worst case for  $BD_{switch}$ ,  $p$ -value  $< 0.022$  in the worst case for  $BD_{mean}$ ). Furthermore, the runs using  $BD_{switch}$  outperform  $BD_{mean}$ . This is particularly interesting as computing a behavioral distance is expensive, in particular for certain BSM, like hamming distance, for instance.  $BD_{switch}$  is then more interesting than  $BD_{mean}$  both in terms of efficiency and required computational power. Further results aim at more deeply exploring the efficiency of each BSM, the influence of  $p_g$  parameter and the influence of number of available BSM.

b) *Control: single BSM*: Treatments using a single BSM were considered as control experiments. Two other treatments were added: one without diversity (one single objective to optimize: the number of collected balls) and another one with a BSM that returns a random value. Results are plotted on figure 4. Median fitnesses range between 0.125 and 0.375 (medians of the control experiments are 0). The low fitness of the experiments using hamming distance seems contradictory with the results reported in [20]. In this paper, the setup was the "simple" collectball, i.e. the collectball without the switch and the other room. As new sensors have been added in the

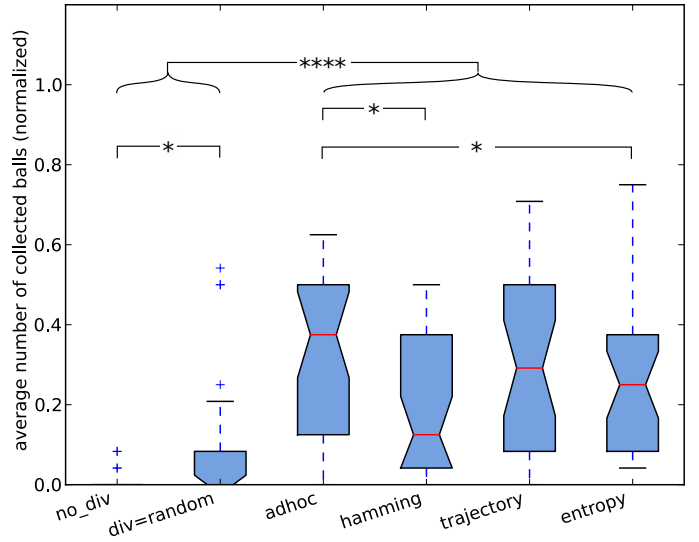


Fig. 4. Control experiments using a single BSM during evolution on the ball collecting task. Medians of 30 runs after 4000 generations.

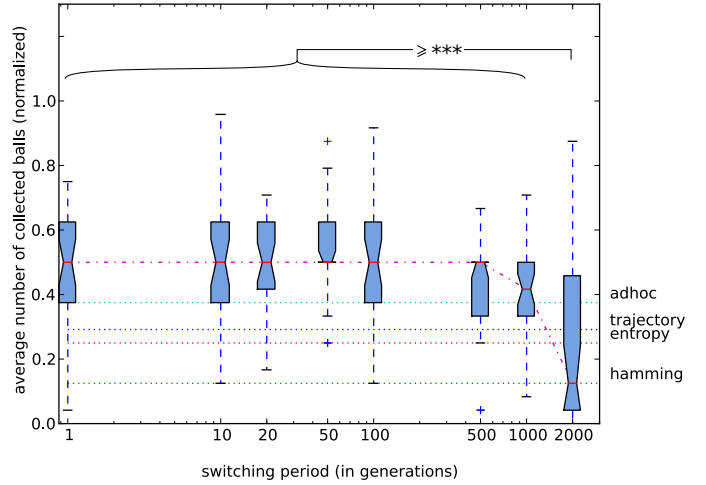


Fig. 5. Influence of the switching period between BSM. Experiments with 4 BSM (adhoc, hamming, trajectory, entropy). Medians of 30 runs after 4000 generations. The medians of the control runs have been reported on the plot to ease comparisons. Only statistical difference between  $p_g = 2000$  and the other treatments are shown.

setup used here, the dimensionality of the binary vector used to compute hamming distances has increased. To explain these bad results of hamming BSM, we hypothesize that either the vector size has grown too much or that some features of the added sensor values decrease the efficiency of this measure. These results highlight that, even a BSM that revealed to be efficient on some problems may well not be that efficient on new problems, even if they are very similar. This highlights the difficulty of choosing the right BSM and confirms the potential of an approach that do not need to choose a single BSM.

c) *Random choice between 4 BSM: influence of the switching period  $p_g$* : In these experiments, the  $BD_{switch}$  objective is used. Every  $p_g$  generations, one of the four BSM previously introduced is randomly selected (adhoc, hamming, trajectory and entropy) and used for the next  $p_g$  generations. Treatments with  $p_g \in \{1, 10, 20, 50, 100\}$  give better results

<sup>3</sup>adhoc BSM is one of the most efficient BSM on this task, see figure 4.

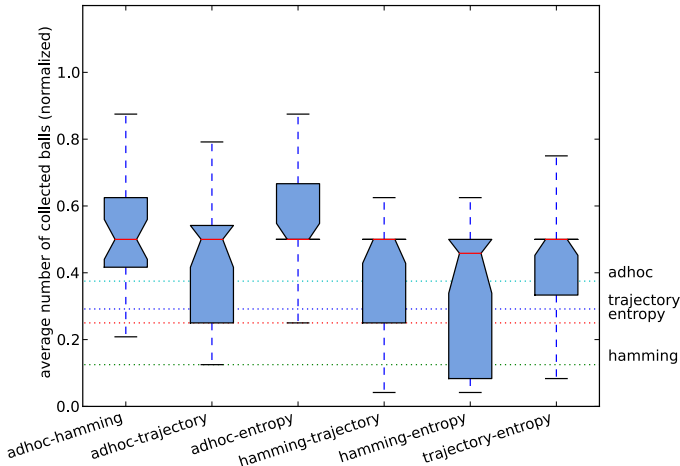


Fig. 6. Experiments with a switch between two BSM: influence of the available BSM.  $p_g = 50$ . Medians of 30 runs after 4000 generations. The medians of the control runs have been reported on the plot to ease comparisons. Statistical significances have not been reported for clarity reasons.

than any of the single BSM treatments (statistically significant, Mann-Whitney U test,  $p - value < 0.01$ )<sup>4</sup>. The median of all sets of experiments except for  $p_g = 1000$  and  $p_g = 2000$  equal 0.5. The results of treatments with  $p_g \in \{1, 10, 20, 50, 100\}$  are not statistically different between each other (Mann-Whitney U test,  $p - value > 0.05$ ) and they are all different from those with  $p_g \in \{1000, 2000\}$ . Treatments with  $p_g \in \{20, 50\}$  give statistically different results from the treatment with  $p_g = 500$ . As the runs last for 4000 generations, it is not surprising that above a period of 500, the effect of BSM switching starts to decrease as at  $p_g = 500$ , there are only 8 BSM switches during a run. For  $p_g = 2000$ , there is only one switch. This treatment gives then, not surprisingly, performances that are close to the control experiments with a single BSM. Figure 5 highlights then that as long as enough switches are done during the run,  $p_g$  has a weak influence. This parameter is then not critical.

*d) Random choice between 2 BSM: influence of the set of available BSM:* Figure 6 shows the results of treatments with a switch between two different BSM at a period of 50 generations. All treatments are significantly different from treatments with a single BSM, except the hamming-trajectory whose difference is not statistically significant with the adhoc treatment and the hamming-entropy with the adhoc and the trajectory treatments. It means in particular that any pair of BSM gives better results than each of these BSM considered separately. Even with only two BSM then, a significant increase in performance is observed. Although the median is the same for all runs, some results are statistically different. The adhoc-entropy run, for instance, is statistically different from all other treatments ( $p - value < 0.05$ ), except adhoc-hamming.

*e) 2 BSM vs 4 BSM:* All treatments with 2 BSM and 4 BSM with  $p_g \in \{1, 10, 20, 50, 100\}$  have the same median at 0.5. The results of treatments with adhoc-hamming and adhoc-entropy are not statistically different from any of the treatments with 4 BSM and  $p_g \in \{1, 10, 20, 50, 100\}$ . The

<sup>4</sup>Treatments with  $p_g \in \{500, 1000\}$  are also significantly better than single BSM treatments, except for adhoc BSM.

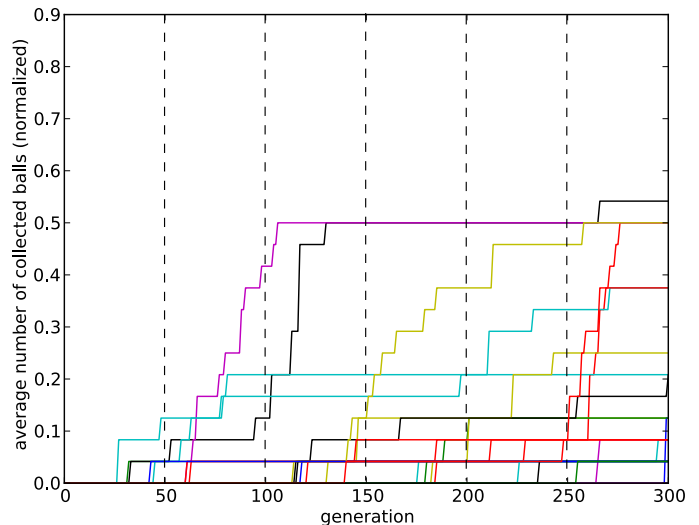


Fig. 7. Fitness along the 300 first generations for all the 30 runs of the treatments with  $p_g = 50$ . Vertical dotted lines represent the generations in which a switch occurred.

results of adhoc-trajectory are statistically below those of the treatment with 4 BSM and  $p_g = 50$  ( $p - value < 0.02$ ). The results of the trajectory-entropy treatment are statistically below those of the treatments with 4 BSM and  $p_g \in \{20, 50\}$  (resp.  $p - value < 0.05$  and  $p - value < 0.006$ ). Finally, the treatments with hamming-trajectory and hamming-entropy are statistically below any of the treatments with 4 BSM and  $p_g \in \{1, 10, 20, 50, 100\}$  ( $p - value < 0.031$  in the worst case). To sum up this part, the difference between treatments with 2 BSM and treatments with 4 BSM is low, but significant for several treatments and in every case in favor of 4 BSM treatments.

*f) Correlation between behavioral diversity switch and fitness increase:* Figure 7 shows the fitness of the 30 runs with  $p_g = 50$ . There is no clear and immediate effect on the fitness of the behavioral diversity switch. This is not surprising as this switch has an impact only on the exploration ability. A better exploration ability do not immediately result in best results as exploration may take time before interesting solutions are discovered.

*g) Example of result:* Individuals able to collect up to 23 balls out of a maximum of 24 have been generated (figure 8). Surprisingly, some of the most efficient neural networks do not contain any recurrent connection and only one hidden neuron (figure 9). The network topology only needs three structural mutations to be found, but probably requires an accurate tuning that makes it difficult to discover without specific exploration fostering mechanisms.

## V. LOCOMOTION TASK

### A. Experimental setup

The control of legged robots combines the challenge raised by the use of many redundant degrees of freedom with the need to tame dynamic behaviors. This difficult problem in robotics has a long history in evolutionary computation because it is often seen as an interesting benchmark to evolve controllers that could be useful in robotics[13], [17], [2], [26], [14].

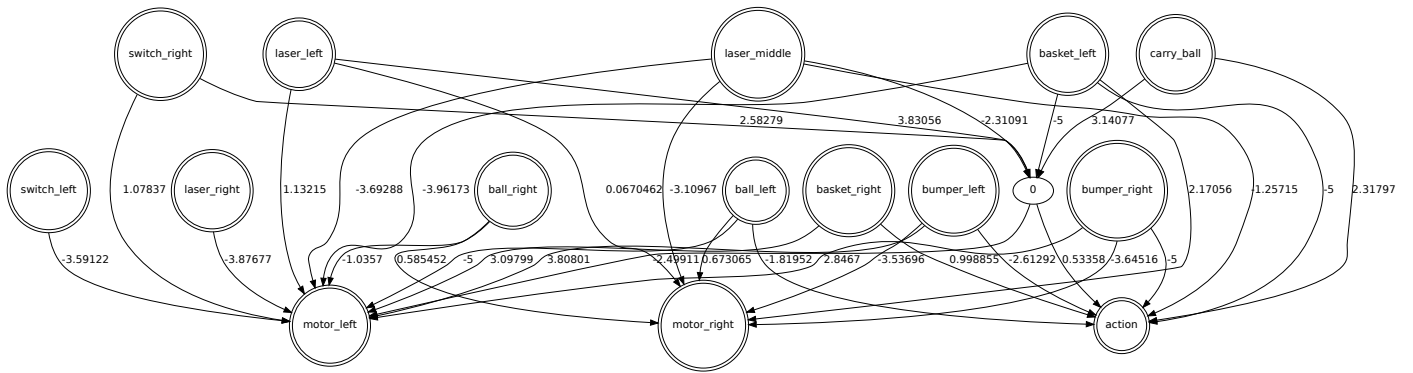


Fig. 9. Neural network generating the behavior shown on figure 8 for the ball collecting task.

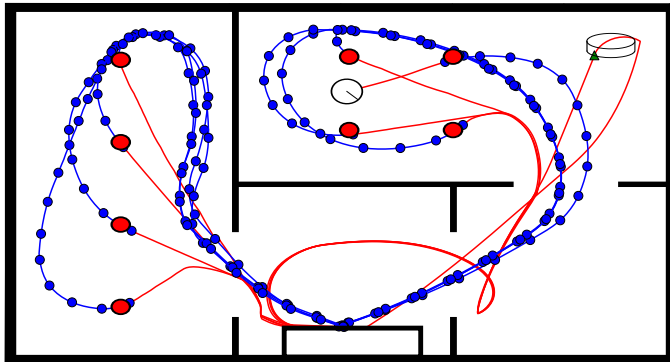


Fig. 8. Example of trajectory for an individual collecting 23 balls. Only the trajectory of the first evaluation is shown. Lines with circles show part of the trajectory where the robot carries a ball.

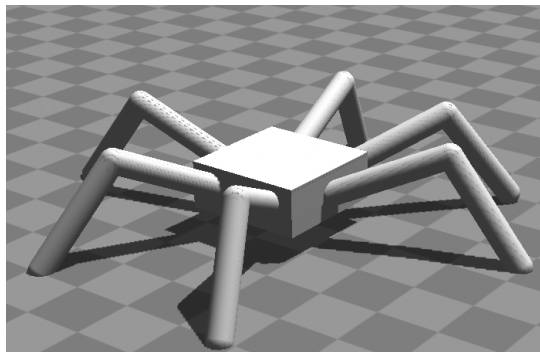


Fig. 10. Overview of the hexapod robot used in these experiments. 15 degrees of freedom must be controlled.

This continuing interest is also rooted in the observation that locomotion is one of the most fundamental ability of animals. Here we use this task to assess the advantages of using several BSM by testing it in a setup very different from ball collecting.

A hexapod robot is simulated in a dynamic simulator<sup>5</sup>(figure 10) and its 18 degrees of freedom must be synchronized to move as fast as possible. The angular position of each degree of freedom  $i$  is governed by the periodic function

$$\vartheta_i(t) [3]:$$

$$\vartheta_i(t) = c_j + A_j \cdot \varphi(F \cdot t + p_i) \quad (5)$$

$$\text{where } \varphi(x) = \tanh(3 \cdot \sin(x)) \quad (6)$$

$F$  is fixed and it is common to all degrees of freedom;  $c_j$  and  $A_j$  are shared by all joints of the same class (there are three classes of joints, each leg being moved by one joint of each class: knee, up-down hip and forward-backward hip);  $p_i$  is different for each degree of freedom. The total number of parameters for this controller is therefore  $3 \times 18 + 3 + 3 = 60$ . To make the discovery of regular behaviors easy, only 5 values are possible for each parameter. They are uniformly spread over the range of movement of each degree of freedom ( $A_1, A_2, A_3 : [0, \frac{\pi}{8}]$ ,  $c_1, c_2, c_3 : [0, \frac{\pi}{24}]$ ,  $p_i : [0, 2\pi]$ , see appendix).

The fitness is the distance covered in 25 simulated seconds (time-step is 0.01 seconds).

*h) Behavior similarity measures and distances:* The choice of the best BSM to improve diversity when evolving gait controllers is an open problem. Here we investigate four different BSM:

- Contact. Walking gaits have been historically described by entomologists using parallel diagrams in which each horizontal line corresponds to a leg (e.g. [6]). For each time step, a black dash is drawn when the leg is in contact with the ground, otherwise nothing is drawn (see e.g. figure 12). For instance, such diagrams make easy to distinguish a tripod gait from a tetrapod gait. Taking inspiration from this description, the gait of an hexapod robot can be described by a matrix  $C$ , in which each row corresponds to a time-step:

$$C_{tj} = \begin{cases} 1 & \text{if the leg } i \text{ is in contact at time-step } t \\ 0 & \text{otherwise} \end{cases}$$

Behaviors are compared by computing the number of differences between their matrix  $C$ .

- End point. When evolving a biped gait with the novelty search algorithm – which also relies on a BSM –, Lehman and Stanley[15] proposed to describe each behavior by the position of the center of mass at the end of the experiment. Behaviors are compared by computing the Euclidean distance between end points.

<sup>5</sup><http://www.ode.org>

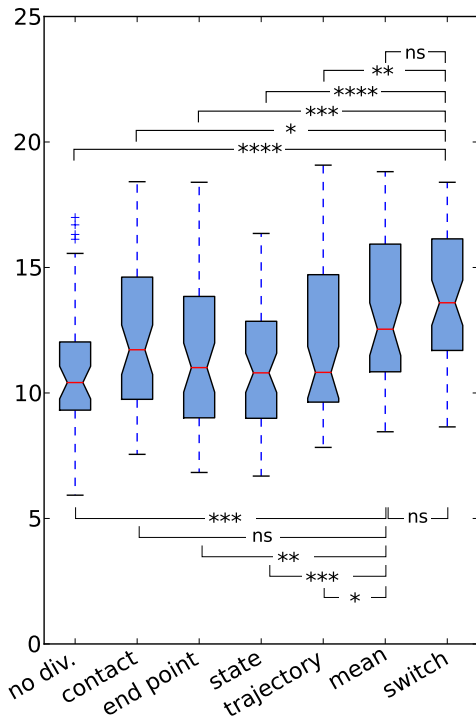


Fig. 11. Medians of the distance covered by the best individual of each run (60 runs) after 2000 generations (400000 evaluations). DBD means “Dynamic Behavioral Diversity”.

- State. At each time step, the position of each degree of freedom describes the current state of the robot. The behavior of the robot can therefore be described by the list of angular positions for each time step. Behaviors are then compared by computing the average Euclidean distance between vector states at each time-step.
- Trajectory. The behavior of the robot can also be described by the 3D trajectory of its center of mass. In this case, behaviors are compared by computing the average Euclidean distance between 3D positions at each time step.

We consider the following treatments:

- fitness only (no diversity, control experiment). Diversity is not encouraged at all; the EA is therefore a standard single-objective (rank-based) EA.
- fitness + a single BSM behavioral diversity objective;
- fitness +  $BD_{mean}$ ;
- fitness +  $BD_{switch}$  ( $p_g = 50$  generations).

We launched 60 runs of 2000 generations and we encouraged diversity using the same multi-objective approach as the one used in the previous section. The size of the population was 200. The 60 parameters have been mutated using a shift to the left value or to the right value (only 5 values are possible for each evolved parameter); cross-over was not used.

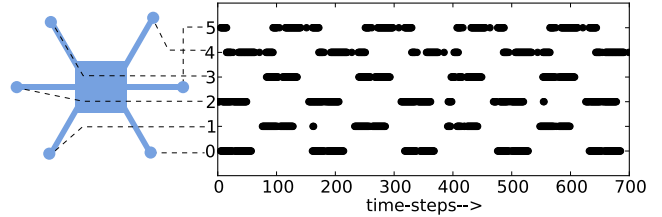


Fig. 12. Contact diagram of the best gait obtained with dynamic behavioral diversity. For each time-step, the presence of a black dot denotes that the leg is in contact with the ground. This gait is mostly stable because three legs are always in contact.

## B. Results

Figure 11 shows the median distance achieved with each of the treatment. Surprisingly, using a single BSM does not seem to improve performance much and only the “contact” BSM leads to results statistically different from the control experiment (no diversity). At first sight, this result appears to contradict recent results about behavioral diversity [11], [19], [18], [20]. However, the analysis of the controllers obtained with the control runs (no diversity) shows that they actually perform well, hence showing that this walking task is easy enough to not require any diversity preservation mechanism to obtain good controllers (the controller is designed so that many different gaits are possible but many of them are working).

There is nonetheless room for improvement: the “switch” the “mean” variants obtained better results than any of the other approach, with statistically significant p-values ( $p < 0.025$  in the worst case). This result confirms that using several BSMs removes the need to choose between BSMs and improves the performance. Figure 12 describes the best gait obtained with the “switch” variant.

## VI. CONCLUSION AND PERSPECTIVE

The obtained results can be summarized as follows:

- using several BSM in a single run increases the performance while releasing the constraint to choose the most appropriate BSM;
- switching between BSM is more efficient than taking the mean behavioral diversity while requiring less computational power;
- the difference is significant with two BSM only, but different BSM pairs may yield different results;
- treatments with four BSM slightly outperform those with two BSM;
- the period of the switch is a non critical parameter, as long as enough switches are performed during a run (more than eight).

The results validate the hypotheses we made, at least on the considered setups: using several BSM during a run actually do not decrease performance. It gives more chance to avoid local optima and converge towards more efficient solutions. It also



validates the hypothesis that a single BSM may not reward every potentially interesting original behavior.

The results suggest to use as many BSM as available, at least up to four. Although it may seem to be a limitation of this approach, as more BSM are required, in practice, defining different BSM is not a difficult problem, whereas choosing one out of these is very difficult. Switching between BSM simplifies then the preparation of an experiment, does not significantly increase the computational load and introduces no critical parameter.

The mean BSM approach could easily be extended to novelty search, but the BSM switch approach would need some adaptations in the archive management process as randomly changing the BSM during the run will impact the choice of individuals to include in it.

## VII. ACKNOWLEDGMENTS

This work is supported by the ANR CreAdapt project (ANR-12-JS03-0009).

## REFERENCES

- [1] L. T. Bui, J. Branke, and H. A. Abbass. Diversity as a selection pressure in dynamic environments. In *GECCO'05: Proceedings of the 7th annual conference on Genetic and Evolutionary Computation*, pages 1557–1558. ACM, 2005.
- [2] J. Clune, C. Ofria, and R. T. Pennock. The Sensitivity of HyperNEAT to Different Geometric Representations of a Problem. In *Proc. of GECCO*, pages 675–682, 2009.
- [3] A. Cully and J.-B. Mouret. Behavioral Repertoire Learning in Robotics. In *Proceedings of GECCO*, 2013.
- [4] E. D. de Jong, R. A. Watson, and J. B. Pollack. Reducing bloat and promoting diversity using multi-objective methods. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 11–18, 2001.
- [5] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
- [6] F. Delcomyn. The Locomotion of the Cockroach *Pariplaneta americana*. *Journal of Experimental Biology*, 54(2):443–452, 1971.
- [7] S. Doncieux and J.-B. Mouret. Behavioral diversity measures for Evolutionary Robotics. In *IEEE Congress on Evolutionary Computation, 2010 (CEC 2010)*, pages 1303–1310, 2010.
- [8] S. Doncieux, J.-B. Mouret, N. Bredeche, and V. Padois. Evolutionary Robotics: Exploring New Horizons. In S. Doncieux, N. Bredeche, and J.-B. Mouret, editors, *New Horizons in Evolutionary Robotics: post-proceedings of the 2009 EvoDeRob workshop*, pages 3–25. Studies in Computational Intelligence. Springer-Verlag, 2011.
- [9] D. Floreano, P. Husbands, and S. Nolfi. Evolutionary Robotics. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 1423–1451, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [10] D. E. Goldberg. Simple genetic algorithms and the minimal, deceptive problem. In *Genetic algorithms and simulated annealing*, pages 74–88. San Mateo: Morgan Kaufman Publisher, 1987.
- [11] F. J. Gomez. Sustaining diversity using behavioral information distance. In *GECCO'09: Proceedings of the 11th annual conference on Genetic and Evolutionary Computation*, pages 113–120. ACM, 2009.
- [12] J. D. Knowles, R. A. Watson, and D. W. Corne. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. *First International Conference on Evolutionary Multi-Criterion Optimization*, 1993:268–282, 2001.
- [13] J. Kodjabachian and J.-A. Meyer. Evolution and development of neural networks controlling locomotion, gradient-following, and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks*, 9(5):796–812, Jan. 1997.
- [14] S. Koo, J.-b. Mouret, and S. Doncieux. The Transferability Approach : Crossing the Reality Gap in Evolutionary Robotics. *IEEE Transaction on Evolutionary Computation*, 17(1):122–145, 2013.
- [15] J. Lehman and K. O. Stanley. Abandoning objectives: evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, Jan. 2011.
- [16] S. Mahfoud. *Handbook of Evolutionary Computation*, chapter Niching Methods. Taylor & Francis, 1997.
- [17] J. Meyer, S. Doncieux, D. Filliat, and A. Guillot. Evolutionary approaches to neural control of rolling, walking, swimming and flying animats or robots. *Studies in Fuzziness and Soft Computing*, 109:1–44, 2003.
- [18] J.-B. Mouret and S. Doncieux. Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *IEEE Congress on Evolutionary Computation, 2009 (CEC 2009)*, pages 1161–1168, 2009.
- [19] J.-B. Mouret and S. Doncieux. Using Behavioral Exploration Objectives to Solve Deceptive Problems in Neuro-evolution. In *GECCO'09: Proceedings of the 11th annual conference on Genetic and evolutionary computation*, pages 627–634. ACM, 2009.
- [20] J.-B. Mouret and S. Doncieux. Encouraging Behavioral Diversity in Evolutionary Robotics: An Empirical Study. *Evolutionary computation*, 20(1):91–133, Aug. 2012.
- [21] C. Ollion and S. Doncieux. Why and How to Measure Exploration in Behavioral Space. In *GECCO '11: Proceedings of the 13th annual conference on Genetic and Evolutionary Computation*, pages 267–294, 2011.
- [22] K. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.
- [23] O. Teytaud. On the hardness of offline multi-objective optimization. *Evolutionary Computation*, 15(4):475–491, 2007.
- [24] L. Trujillo and G. Olague. Discovering several robot behaviors through speciation. *Applications of Evolutionary Computing*, 4974:164–174, 2008.
- [25] L. Trujillo, G. Olague, E. Lutton, F. Fernández de Vega, L. Dozal, and E. Clemente. Speciation in Behavioral Space for Evolutionary Robotics. *Journal of Intelligent & Robotic Systems*, 64(3):323–351, Jan. 2011.
- [26] V. Valsalam and R. Miikkulainen. Evolving Symmetry for Modular System Design. *IEEE Transactions on Evolutionary Computation*, 15(3):368–386, 2011.

## APPENDIX

- MOEA: NSGA-II (pop. size : 200)
- DNN (direct encoding, ball collecting task):
  - number of neurons  $\in [10, 30]$
  - number of connections  $\in [50, 250]$
  - prob. of changing weight/bias: 0.1
  - prob. of adding/deleting a conn.: 0.15/0.05
  - prob. of changing a conn.: 0.03
  - prob. of adding/deleting a neuron: 0.05/0.05
  - activation function for neurons:  

$$y_i = \varphi \left( \sum_j w_{ij} x_j \right) \text{ where } \varphi(x) = \frac{1}{1 + \exp(b - kx)}$$
- Mutation (locomotion task): each parameter has a 10% chance of being incremented or decremented. Five values are available:
  - $A_1, A_2, A_3 : \{0, \frac{\pi}{32}, \frac{\pi}{16}, \frac{3\pi}{32}, \frac{\pi}{8}\}$
  - $c_1, c_2, c_3 : \{0, \frac{\pi}{96}, \frac{\pi}{48}, \frac{3\pi}{96}, \frac{\pi}{24}\}$
  - $p_i : \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi\}$
- Source code :  
[http://pages.isir.upmc.fr/evorob\\_db](http://pages.isir.upmc.fr/evorob_db)