Behavioral diversity measures for Evolutionary Robotics

Stephane Doncieux Jean-Baptiste Mouret

Abstract— **In Evolutionary Robotics (ER), explicitly rewarding for behavioral diversity recently revealed to generate efficient results without recourse to complex fitness functions. The principle of such approaches is to explicitly encourage diversity in the robot *behavior* space instead of in the space of genotypes (the space explored by the evolutionary algorithm) or the space of phenotypes (the space of robot controllers and morphologies). To implement such approaches, a similarity between behaviors needs to be evaluated but, up to now, used similarity measures are problem-specific.**

The goal of this work is to explore *generic* behavioral similarity measures that only rely on sensori-motor values. With such a measure, we managed to evolve the topology and the parameters of neuro-controllers that make a simulated robot go towards a ball, take it, find a basket, put the ball into the basket, perform a half-turn, search and take another ball, put it into the basket, etc. In this experiment, two objectives were simultaneously optimized with NSGA-II: the number of collected balls and the generic behavioral diversity objective. Several generic behavioral measures are compared. To confirm the interpretation of behavioral diversity objective and in an attempt to characterize behavioral similarity measures, they are also compared to human-made behavioral similarity evaluations. They reveal to classify behaviors globally as humans did, but with no clear correlation between the closeness to human classification and the efficiency within an evolutionary run.

I. INTRODUCTION

Darwin's theory of evolution relies on natural selection but also on diversity of life forms. In Evolutionary Computation (EC), stochastic search operators, i.e. mutation and crossover operators, were introduced to create such a diversity. However, it has been observed in EC that these two operators are often insufficient to keep a population diverse enough to avoid a premature convergence. To tackle this problem, researchers have suggested to alter the search landscape in order to explicitely foster diversity. This idea led to fitness sharing methods [11], which decrease the fitness of similar individuals. Multi-objective formulations, in which fitness is not aggregated with diversity, have also proved to be efficient in improving evolutionary algorithms [7], [4], [18].

These methods rely on a *distance between genotypes*. Unfortunately, this makes them difficult to use in Evolutionary Robotics (ER) because complex genotypes are often used for which distances are either computationally infeasible or almost meaningless. Thus, many works in ER study neural networks whose topology is evolved: in the worst case, neural networks are encoded as directed graphs whereas typical graph distances is a NP-hard problem [5]; in the best case, a distance can be written for a specific encoding but it cannot be translated to other encodings. Besides these algorithmic considerations, the behavior of a robot results from the evaluation of a dynamic system made with the robot and its environment. Two individuals with close genotypes can exhibit a very different behavior but, conversely, two individuals with very different genotypes can have exactly the same behavior. For instance, two robots can be controlled by different neural networks but act in the same way if the parts that differ are not connected to the ouputs.

Several papers [12], [18], [19], [15] recently described an alternative to genotypic distance in ER: computing distance between behaviors. However, while the authors reported substantial improvements in their ER experiments by encouraging *behavioral diversity* using behavior similarity measures, they employed distances that relied on problem-specific descriptions of behaviors. For instance, in a maze navigation experiment, Lehman and Stanley [15] used the final position of the robot to compare behaviors. As another example, Mouret and Doncieux [18], evolved controllers for robots that had to switch some lights in a particular order; the vector of light states at the end of the experiment was used as a behavior descriptor. In each of these experiments, a vector of problem dependent features was used, allowing to compute a simple Euclidean distance to evaluate how similar behaviors were. All these measures rely on expert knowledge; but this knowledge may be unavailable and its use contradicts one of the main goals of ER: minimizing human intervention in the design process.

As a consequence, having recognized the importance of behavioral diversity in ER but also the lack of generic behavior distances, this paper focuses on defining and comparing problem-independent similarity measures. The questions addressed by this work can be summarized as follows:

- Does a problem-independent behavioral diversity enhance the search as with problem-specific measures?
- How critical is the choice of the behavioral similarity measure?
- If it is, what makes a behavioral similarity measure more efficient than another?

The first goal of this paper is to introduce and benchmark several problem-independant behavior similarity measures. The focus will be on the evolutionary design of behaviors for mobile robots. All of the considered measures will rely on easily available data in this context, i.e. sensori-motor values. In a second step, the considered similarity measures will be analyzed to understand why they perform differently. In the absence of any ground truth for behavior similarity, our analysis is based on the comparaison between these measures

Stephane Doncieux and Jean-Baptiste Mouret are with the Université Pierre et Marie Curie (UPMC) - Paris 6, Institut des Systèmes Intelligents et de Robotique (ISIR), CNRS UMR 7222, F-75005, Paris, France; e-mail: stephane.doncieux,jean-baptiste.mouret@isir.upmc.fr.

Doncieux, S., Mouret, J.-B. *"Behavioral diversity measures for Evolutionary Robotics"* IEEE Congress on Evolutionary Computation, 2010 (CEC 2010)

and measures performed by *humans*.

II. Related works

A. Fitness sharing

Keeping a diverse population requires to balance the efficiency of a solution with its originality within the population. Initial attempts to take it into account relied on the idea of lowering the fitness of an individual by an amount equal to the number of similar individuals in the population [11].

Besides this method that directly modifies the fitness value, crowding methods aims at adapting the selection scheme to take into account the similarity between individuals when choosing which individual to replace [9]. Many over methods allow to build and maintain different *niches*, a niche being defined as a set of similar solutions, generally sharing a common resource, i.e. a fitness value, see [24] for a review and comparison of such methods.

The diversity can also be used as a separate selection pressure, just like any other problem-dependent fitness function. It was shown that this last choice, within a multi-objective approach, led to better results [7], [4], [1], [19], [18]. Likewise, different ways of evaluating a diversity objective have been compared and the distance to the whole population revealed to be more efficient [4].

Few work has focused on the use of *behavior* instead of genotype or phenotype to evaluate the similarity between individuals. This notion appears only when the fitness relies on the observation of a dynamical process, as for a robot in interaction with its environment. The behavior is then a description of this interaction that depends on the environment, and in particular on the initial conditions, on the robot features, i.e. on the phenotype, and on time.

[12] used a behavioral distance within a crowding selection scheme [9] to solve the Tartarus problem—a box pushing problem in a discrete world—while comparing different behavior similarity measures.

[19], [18] defined behavioral diversity as the use of a diversity objective within a multi-objective scheme. In [19], such an approach revealed to compensate the deceptiveness of a XOR-AND-XOR boolean function and in [18] this approach allowed to solve a sequential light-seeking task as efficiently as with a more directed incremental approach. The work presented here follows this approach.

B. Similarity measures

Whatever method is employed to foster behavioral diversity, a similarity measure is required. In the typical set-up, for each generation and for each individual, the distance to the rest of the population is evaluated. This implies to compute, for each generation, n^2 similarity measures, if n is the size of the population[1]. Short behavior observations might be used to accelerate computations, but these sequences must be long enough to actually be representative of robots behaviors. Computational time is then a critical issue. As an example, consider a population of size 100. For each generation, at

[1]Actually $\frac{n*(n-1)}{2}$ if the measure is symmetric.

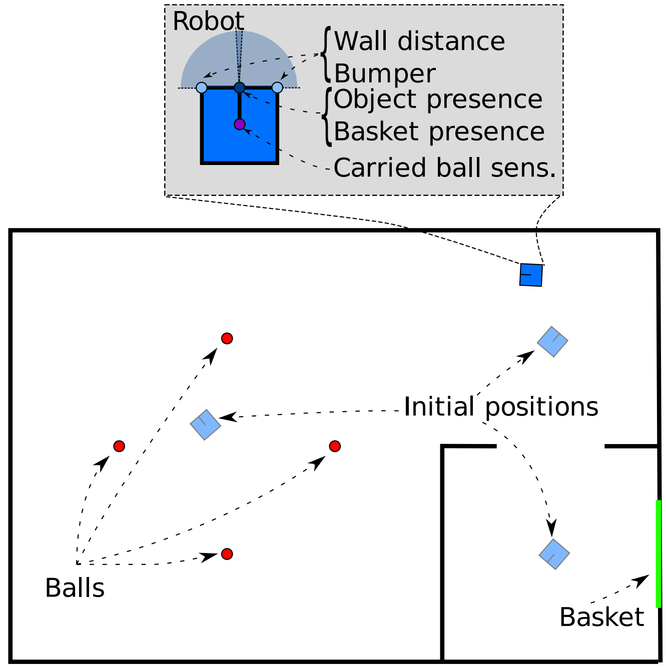

Fig. 1. Overview of the arena and of the robot. The robot has a square shape and has nine different sensors. Object and basket presence field of views are represented on the picture. Four balls are placed in the environment. The goal of the experiment is to put as many balls as possible into the basket. The initial positions of the three evaluation experiments used for the fitness computation are plotted on the figure.

least 4950 comparisons have then to be performed. If the similarity measure requires $0.01s$, the diversity objective needs $49.5s$ per generation, i.e. more than 27 hours for 2000 generations.

Comparing sequences of values, may they be binary or real is a critical issue for numerous applications like genome study, data search for video or audio data or plagiarism detection. In all of those applications, algorithms have been designed to evaluate the similarity between sequences. Once a discretization has been performed, real value vectors can also be compared using similarity measures operating on discrete values. These methods will consequently be presented first before reviewing measures working on real values.

a) Similarity measures for sequences of discrete values: Evaluating the distance between sequences of discrete values may be done thanks to an *edit distance* [16] that measures the minimum number of operations that have to be done to transform one sequence into the other. In our context, such a distance can hardly be used as computing it is in $O(m^2)$ in time [13], if m is the length of the sequences. For the considered sequence length, it was far too slow to be of use.

Finding common subsequences may be the basis for a similarity search. Measures tolerant to noise and scaling have thus been developped [6], but they give only a Boolean answer relative to a given similarity threshold.

b) Similarity measures for sequences of real values: As for discrete sequences, a similarity measure has been developed to find out which sequences are similar to a particular sequence out of a given set [3]. This method is designed to be robust to noise, scaling and translation; these features are interesting for our application, but it only provides a Boolean answer— similar or not — relying on a threshold provided beforehand. It can't then be used directly.

To speed up measures, smaller sequences supposed to be representative of the initial one can be used. The comparison may then rely on simple Euclidian distances, even if the initial sequence is long. [2] suggests to use the first coefficients of a Fourier transform as a descriptor of sequences, at least for a preliminary filtering. Likewise, [27] suggests to use the first components of a Principal Components Analysis in the case of multivariate sequences and directly compare them.

A more general information distance based on the notion of Kolmogorov complexity has also been proposed [17]: Normalized Compression Distance (NCD), in which an approximation of Kolmogorov complexity is evaluated with the help of real-world compressors. Such a distance was used in [12] but it revealed to be too slow for our application.

C. Other related works

Evaluating the novelty of a behavior is also an issue for developmental robotics [26]. Focusing learning on new behaviors, or at least behaviors for which the learning rate is the fastest [23], discovering how to build internal representations of its own body [14], using embodiment to structure input spaces [25]: all these tasks require to find patterns or similarities from high dimensional streams of robot perceptions and actions.

Nonetheless, developmental robotics is mainly centered on experiments with a single robot and a long life span—at least longer than for the robot considered in the present paper. Moreover, DR compares and finds similarity within a unique stream of sensor-effector values whereas behavioral diversity need to compare many different streams of sensor-effector values. On the long term, such issues may converge, but as for now, drawing a direct link between diversity in ER and DR is not straightforward.

III. Method

As suggested in [19], [18], [4], [7], we add a diversity objective to the fitness function in a Pareto-based multi-objective optimization. Following the conclusions of [4], the behavioral diversity objective to maximize $o_{bd}(x)$ is the average distance to the rest of the population. Hence the maximization of the fitness function $F(x)$ is transformed to the multi-objective maximization:

$$\text{maximize} \begin{cases} F(x) \\ o_{bd}(x) = \frac{1}{size(P)} \sum_{y \in P} \sigma(x,y) \end{cases}$$

where P denotes the current population, x, y two individuals, $\sigma(x,y)$ the measure of the similarity of x and y. $\sigma(x,y) = 0$ if $x = y$ and the greater $\sigma(x,y)$, the more different they are.

A. Considered similarity measures

The set of sensor and effector data ϑ is defined as follows:

$$\vartheta = [\{\mathbf{s}(t), \mathbf{e}(t)\}, t \in [0,T]]$$

where $\mathbf{s}(t)$ is the vector of size n_s of the perceptions at time t, i.e. the values coming from the n_s sensors; $\mathbf{e}(t)$ is the vector of size n_e of the effector values at time t; T is the observation length. For simplicity, in the following $\mathbf{s}(t)$ is in $[0,1]^{n_s}$ and $\mathbf{e}(t)$ is in $[0,1]^{n_e}$.

a) Hamming distance: The Hamming distance counts the number of bits that differ between two binary sequences. It can be used to evaluate behavior similarity with, as inputs, ϑ_{bin}, the binarized version of ϑ, computed as follows:

$$\vartheta_{\mathbf{bin}} = [\vartheta_{\mathbf{bin}}(t), t \in [0,T]] = [\{\mathbf{s_{bin}}(t), \mathbf{e_{bin}}(t)\}, t \in [0,T]]$$

with $\mathbf{s_{bin}}(t) = \{s_{bin,0}(t), \dots, s_{bin,n_s}(t)\}$,
$\mathbf{e_{bin}}(t) = \{e_{bin,0}(t), \dots, e_{bin,n_s}(t)\}$ and

$$s_{bin,i}(t) = \begin{cases} 1 & \text{if } s_i(t) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$$e_{bin,i}(t) = \begin{cases} 1 & \text{if } e_i(t) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

with this definition, the hamming distance is computed as follows:

$$\sigma_{ham}(\vartheta_1, \vartheta_2) = \sum_{t=0}^{T} h(\vartheta_{\mathbf{1,bin}}(\mathbf{t}), \vartheta_{\mathbf{2,bin}}(\mathbf{t}))$$

$$h(\vartheta_1, \vartheta_2) = \sum_{i=0}^{len(\vartheta_1)} \delta(\vartheta_1[i], \vartheta_2[j])$$

where $len(\vartheta_1) = len(\vartheta_2)$ denotes the length of the binary sequences ϑ_1 and ϑ_2 and where $\delta(i,j)$ is the Kronecker delta:

$$\delta(i,j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

b) Measure based on Fourier coefficients: Parseval's theorem guarantees that the distance between two sequences is the same in the frequency or in the temporal domain [22]. As suggested in [2], the very first coefficients of the Discrete Fourier transform may carry information descriptive of the sequence they are associated to. Using it instead of the complete sequence has the advantage to reduce the dimensionality: the size of vectors to consider goes from $(n_s + n_e) * T$ to $(n_s + n_e) * n_F$, if n_F is the number of retained coefficients. The measure is then not dependent on the observation length anymore and simple Euclidean distances between the resulting vectors can be used.

c) State count: This measure relies on discretizing available data in order to define perception-action states; the number of times the robot was in a particular state is then evaluated. This results in a vector of n integers, n being the number of such states. States are user-defined. The similarity measure is defined as the mean distance between the vectors:

$$\sigma_{sc}(x, y) = \frac{1}{n} \sum_{i=1}^{n} d_i(x, y)$$

$d_i(x, y)$ is a squared Euclidean distance normalized by the maximum number of times state st_i has been reached by x and y. This avoids individuals reaching a huge number of times a particular state to be over-rewarded:

$$d_i(x, y) = \begin{cases} \frac{(n_i(x)-n_i(y))^2}{max(n_i(x),n_i(y))^2} & \text{if } max(n_i(x), n_i(y)) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$n_i(x)$ is the number of times x has spent in a state st_i. States can be defined either from a coarse discretization, i.e. using $\vartheta_{\mathbf{bin}}$, for instance, or they can include some expert knowledge. The two different cases will be considered here.

As this objective has to be maximized by evolution, individuals that are the only ones to reach a particular state will be rewarded, as do individuals that reach a state a different number of times relative to other individuals of the current population. The normalization increases the reward granted to individuals that are the first to reach a particular state, even if it is reached only one time.

d) Trajectory based similarity measure: This measure consists in discretizing the environment and counting the number of times spent in each square. It is then similar to the state count measure, but with position data instead of perception/action data. This similarity measure should be considered as a control measure that directly exploits the knowledge of the trajectory of the robot; it doesn't exploit the data coming from sensors and effectors.

Other measures could be defined on the basis of the trajectory, but the focus here was on those exploiting information easily available: trajectory is easy to get in simulation, but is more tricky to get on a real robot[2].

B. Robotics experimental setup

The choice of the setup stems from several motivations:

- it should be a difficult task from an ER point of view:
  - the fitness function must not be too directive; hence, we chose a sequential task that rewards only the achievement of the whole sequence;
  - the simulation should make a random search as inefficient as possible;
- the problem should not be on the controller abilities but rather on the fitness design; we selected then sensors that are easy to interpret for an evolved neural network;
- it should be easy and fast to simulate to facilitate re-implementations and comparisons.

The chosen task consists in picking up some balls in an arena and putting them into a basket (fig. 1). The robot is a two-wheeled robot with the following sensors:

- two wall distance sensors (linear between 0 and 8 times the robot size and then constant, the output is normalized between 0 and 1, see figure 1);
- two bumpers (1 if it touches a wall, 0 otherwise);
- two ball detection sensors (1 if a ball is in the view field of the sensor, 0 otherwise);
- two basket detection sensors (1 if the basket is in the view field of the sensor, 0 otherwise);
- one carry ball sensor (1 if a ball is carried, 0 otherwise).

The effectors are left and right wheel motors and a "catch ball" motor (if greater than 0.5, pick up a ball if possible, or keep the carried ball, else throw the carried ball if any).

Figure 1 shows the arena with the details of the sensor configuration of the robot.

This setup is difficult from an evolutionary perspective:

- the robot may catch a ball when it moves on it if the catch effector is above 0.5; as soon as the catch effector goes below 0.5, the ball is thrown and disappears from the arena; if the robot touches the basket at this time, it will be counted as a collected ball and otherwise it just disappears; it is then difficult for the robot to release a ball by chance into the basket as it previously has to learn to keep the ball, reach the basket and then, and only then, release the ball;
- the robot shape is a square and the collision detection system avoids the robot to slide along the walls: once the robot collides with a wall, it has to go backwards or to turn, if possible, to get out of it; once a ball has been put into the basket, the robot is then blocked against the wall and need to learn how to escape from this situation;
- the basket is surrounded by walls to make it more difficult to find and to make new ball catching longer and trickier, as the robot first has to go out of the basket room before being able to see any new ball.

[21] used a similar garbage collecting experiment, but in a different setup and for a different purpose. The fitness counted the objects put outside of the arena no matter where and a small reward was also granted to individuals able to pick up objects. The pick-up process was more realistic than our and closely resembling the one of a Khepera robot.

IV. Behavioral diversity within an evolutionary process

These first experiments aim at evaluating how the different similarity measures behave within an evolutionary context. To test the efficiency of the proposed behavior similarity measures, several different setups were designed.

A. Setups

1) Control experiments: The first control experiment use the ball count objective only. It aims at evaluating the difficulty of the task with a straightforward approach.

[2] a simple integration of the odometry has too much drift, a precise trajectory evaluation must then rely on a SLAM algorithm or on an external motion tracking device.

The second control experiment use the ball count objective together with a random objective. This experiment aims at evaluating the role of the second objective and check if a random objective also enhances the search.

The third control experiment use the state count measure including some expert knowledge in the choice of states. It aims at comparing the proposed generic measures with a problem-specific one. The states are chosen to be representative of situations were we know (or at least suppose) what the robot has to do. Following [10], the states are (the action supposed to be done in this circumstance is given in paren, but not used at all in the fitness):

1) no ball carried, no basket around and no ball around (look for a ball);
2) no ball carried, no basket around, ball nearby (go towards a ball);
3) ball carried, no basket around (look for the basket);
4) ball carried, basket ahead (go towards the basket);
5) ball carried, basket ahead, bumpers on (release the ball);
6) no ball carried, basket ahead, bumpers on (go back to escape from the wall);
7) no ball carried, basket ahead, bumpers off (go away from the basket room).

By construction, the state count objective will encourage individuals to reach at least once each of these states, thus putting individuals in a situation were they can collect balls.

The fourth and last control experiment also uses, as a behavioral diversity objective, a state count, but that relies on the trajectory. The arena has been discretized in 24 different squares measuring $5 \times 5m^2$ (arena size is $20 \times 30m^2$, robot length is $0.5m$). It aims at comparing sensory-motor values with another kind of information, i.e. trajectory.

2) Experiments on tested behavior similarities: The Hamming experiment use the Hamming distance on $\vartheta_{\mathbf{bin}}$ to evaluate behavior similarity. Each sensor or effector data is then transformed into a single bit binary value.

The Discrete Fourier Transform (DFT) experiment relies on the Euclidian distance between the two first components of a DFT, for each separate data stream.

The systematic state count experiment uses the state count similarity measure with a straightforward definition of states based on $\vartheta_{\mathbf{bin}}$. As there are 9 sensors and 3 effectors, this leads to 4096 different states.

Experiments are thus the following:

- *ballcount*: 1 objective, ballcount;
- *random*: 2 obj., ballcount and random;
- *state count (inf.)*: 2 obj., ballcount and behavioral diversity with state count distance on "informed" states;
- *trajectory*: 2 obj., ballcount and behavioral diversity with state count distance based on the trajectory.
- *hamming*: 2 obj., ballcount and behavioral diversity with Hamming distance;
- *DFT (2 comp.)*: 2 obj., ballcount and behavioral diversity with Euclidean distance between the 2 first coefficients of the Discrete Fourier Transform;
- *state count (sys.)*: 2 obj., ballcount and behavioral diversity with state count distance on "systematic" states;

An individual is evaluated in three different contexts defined by the initial position of the robot (see figure 1). The position of the balls and the three initial positions of the robot are constant to guarantee that the difficulty of the problem is the same for all and to make behavior comparisons easier. The evaluation in a particular context lasts 2000 time steps. The theoretical maximum fitness value is 12 (4×3), but within 2000 time steps, only 3 balls can be collected; the maximum value is then 9 in practice. The first 2000 time steps out of the total 6000 are kept for behavior similarity measures. This is empirically chosen as a compromise between the ability to discriminate behaviors and similarity measure computation time. NSGA-II [8] is used with a population size of 600 and for 2000 generations. Each experiment is repeated five times. The topology and the synaptic weights of the neural network controlling the robot are evolved with a simple direct encoding. The random generation process creates neural networks with 10 to 20 hidden neurons, 20 to 100 connections, 9 inputs (directly linked to sensors) and 3 outputs (directly linked to actuators). Network size is unbounded after the initial generation.

Several mutation operators are used (probability in paren):

- adding a neuron (0.025)
- deleting a neuron (0.025)
- adding a connection (0.15)
- deleting a connection (0.25)
- modifying a connection weight, possible values are: $\{-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2\}$ (0.2)

No crossover is used. Further details on the neural encoding can be found in [19].

The source code of the experiments is available for download (`http://www.isir.fr/evorob_db`). Experiments were implemented in the Sferes$_{\mathrm{v2}}$ framework [20].

B. Results

The mean number of maximum collected balls per run together with the variability is shown on figure 2. The p-values of a Mann-Withney test comparing experiment results is shown on figure 3.

The *ballcount* experiment does not generate any controller able to put more than one ball into the basket per context, as do experiments with a second random objective. No matter what similarity measure is used, using it always allowed to generate controllers, at least once out of the 5 runs, that collect more than one ball per evaluation. The trajectory experiment had a large variability and is thus not statistically different ($p < 0.05$) from every other experiment (except the *random* one).

The *Hamming* experiment generated the most efficient controllers (statistically different from every other setup except trajectory, $p = 0.167$ and state count informed, $p = 0.329$). The performance of this measure was similar in [12] where it almost equalled NCD. It should anyway be emphasized that most of the employed sensors return binary values. Only one sensor and the three effectors return real values. Experiments in other contexts should be performed to confirm these results. Furthermore, sensor and effector values are comparable as the initial conditions are always the same; with changing initial conditions, such a distance might not be as efficient.

The *state count (inf.)* is the next most successful experiment, confirming results of [10]. The *trajectory*, *DFT (2 comp.)* and *state count (sys.)* give statistically similar results.

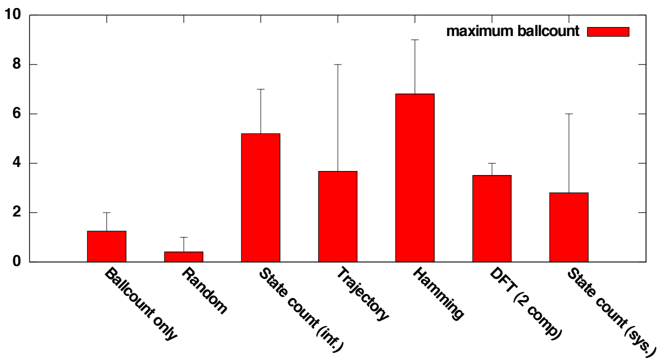

Fig. 2. Mean of the maximum number of collected balls in the three contexts used for evaluation and for each setup after 2000 generations (mean and standard deviation over five different runs).

The maximum value of 9 has been reached, meaning that individuals have been generated that are able to pick up a ball, go to the basket room, release the ball in the basket, go backwards and perform a half-turn, search other balls, collect one, ... Three balls have been collected at best per evaluation (out of four), but actually, some of the evolved individuals revealed to be able to collect the four with some more time.

Coming back to the initial motivations of each control experiment, the following conclusions can be drawn:

- the problem can't be solved directly;
- the behavioral diversity measure doesn't act as a random objective (and vice versa);
- generic behavioral diversity measures can be as efficient as an expert-designed behavioral diversity measure;
- sensori-motor values are competitive with trajectory-based values.

Behavioral diversity objectives can then be both generic and efficient. Furthermore, all behavioral diversity measures do not perform the same. There is a huge difference between the measures, so finding out what makes a similarity measure more efficient than another is an issue to investigate.

V. Behavior similarity measures study

Once it is proved that generic measures can be used, in front of the huge number of different similarity measures that can be defined, it is interesting to try to characterize those measures. The question studied here is then the following: *can we find a way to forecast the efficiency of a behavioral similarity measure within a behavioral diversity objective?*

As the goal of a behavioral similarity measure is to compare behaviors and in the absence of any ground truth, we now look at how it compares to human-made comparisons. Our goal is to answer three questions:

- is it easy to compare behaviors for a human?
- how far or how close are behavioral similarity measures to human-made measures?
- is it correlated to the efficiency within evolution?

If the answer to this last question is positive, looking for the most efficient behavior similarity measure for a behavioral diversity objective would be greatly simplified as it would be possible to make predictions on the efficiency without launching an evolutionary experiment.

Two different setups are considered. In each setup, a set of controllers is selected and, sequentially, for each behavior, we present two other behaviors to a human that has to say which one from the two is the closest to the first behavior. Such a single comparison defines a test case. Answers of different human subjects are compared and we next consider only the cases were all humans agree and see what the behavioral similarity measures answer in these cases. The two setups differ by the choice of controllers. In the first setup, we selected 7 controllers that show an increasing efficiency. In the second, we chose 10 controllers that show a more random behavior relative to the task.

A. Setup1

The chosen behaviors of the first setup can be described as follows:

- no balls collected and the robot ends against the wall;
- one ball is collected, the robot ends against the wall;
- one ball is collected and the robot goes towards the basket but collides with walls before reaching it;
- one ball is collected and put into the basket with no further movements;
- two balls are collected and put into the basket;
- three balls are collected and put into the basket;
- three balls are collected and put into the basket and the robot goes back to pickup the last ball.

This description is not known from the human subjects that can only look at the behavior after a short introduction to the problem, but it is relatively easy to deduce from observation, even for a non-expert. We have considered 35 test cases involving these behaviors.

B. Results for setup1

25 test cases out of 35 had identical results for all the 6 human volunteers. Even in this simple context were behaviors are easy to differentiate and compare, there is then almost 30% of variation. No advices were given on which feature to use to perform the comparison. We don't have any ground truth for the comparison, but at this stage, we can conclude that there are several different ways to perform the comparisons and that they agree on 70% of the data.

We now consider these 70% of data as a ground truth and have looked at how behavioral similarity measures perform on these cases. In order to explore the potential relations between behavior classification efficiency and efficiency of

	random	ballcount	trajectory	state count (inf.)	state count (sys.)	DFT(2 comp.)	hamming
random	1.000	0.041	0.018	0.011	0.018	0.010	0.010
ballcount	0.041	1.000	0.101	0.011	0.147	0.028	0.010
trajectory	0.018	0.101	1.000	0.596	0.523	0.595	0.167
state count (inf.)	0.011	0.011	0.596	1.000	0.070	0.043	0.329
state count (sys.)	0.018	0.147	0.523	0.070	1.000	0.662	0.033
DFT (2 comp.)	0.010	0.028	0.595	0.043	0.662	1.000	0.015
hamming	0.010	0.010	0.167	0.329	0.033	0.015	1.000

Fig. 3. p-values for Mann-Whitney statistical tests performed on each pair of experiments. Comparisons relied on the vectors of the max number of balls put into the basket for each run.

the behavioral diversity objective, figure 4 shows the percentage of closeness between human-made measures and a particular similarity measure relative to the performance in the evolutionary experiment.

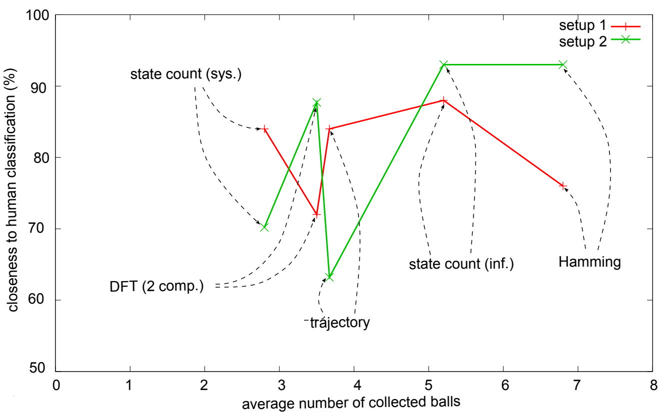

Fig. 4. Closeness of proposed similarity measures related to human-made classification in terms of the performance in a behavioral diversity context. Each dot represents a particular experimental setup. X-axis: average number of collected balls (results from section IV-B), Y-axis: percentage of similarity with human-made measures.

The similarity measures behave as human did as they give the same answer in between 70% and 90% of the cases. This is comforting for the interpretation of the behavioral diversity. Anyway, there is no clear dependency between the performance of the behavioral diversity objective within evolution and the closeness to human measures.

C. Setup2

This second setup aims at evaluating if the previous results are confirmed if we try to compare individuals, i.e. controllers, that appear only during the very first generations. The behavioral similarity measures efficiency is critical during the bootstrap of the process: its efficiency to differentiate individuals that have different ball count values might not be that important as the selection will then be possible according to the ball count objective.

Among the ten considered behaviors, only one has a ball count of one, all the others do not put any ball into the basket. This time, we have considered 120 different test cases. To facilitate and accelerate the comparisons, rather than the video of the behavior, we have shown only the trajectory with the "carrying-ball" information included. As before, in each case, the volunteers have to say, for a particular behavior, which one out of two other behaviors is the closest. In this setup, it revealed much more difficult to perform such comparisons. We have thus made a third choice possible: "impossible to decide with enough confidence".

D. Results for setup2

The first observation is that people confidence in their ability to compare behaviors is very variable. It goes from 76% to 96% with a mean of 88%. As before, we have considered the set of comparisons on which all humans agree. Out of 120 different test cases, only 57, i.e. 47.5%, gathered the same answers. This confirms that the difficulty is not the same as before, as agreements were obtained on 70% of the comparisons in setup1. It also confirms that similarity may differ a lot depending on the priority given to observed features of the behaviors. The closeness of behavioral similarity measures relative to this 47.5% of data, now considered as ground truth, is plotted on figure 4.

The trajectory based measure has a poor result (63.2%), but this is not surprising as the trajectory discretization is very coarse and most trajectories remain in the same area. If we except this value, the similarity measures give on average answers that are similar to the ground truth set in 86% cases (between 70.2 and 93%). This further confirms that such behavior similarity measures globally behave as expected. If we except the trajectory outlier, we can observe a positive correlation between the similarity to human measures and the performance of the evolutionary run.

E. Conclusion

In both setups, the behavior similarity measures give answers that are reasonably close to those of humans. The hypothesis that such measures really behave as a behavioral similarity measure is thus confirmed. These results do not allow to conclude concerning the links between the closeness to a human classification and the performance in a behavioral diversity approach. Characterizing efficient similarity measures, beyond a minimal behavior classification ability, thus remains an open question.

VI. Discussion

The chosen task can be solved by a succession of reactive behaviors. Contrary to incremental approaches, the fitness rewards the whole sequence of behaviors, when it is successful only, and not the intermediate behaviors. Applying it to problems requiring abilities that go beyond simple reactive controllers—memory or learning, for instance—should be investigated, as a useful evaluation of behavioral similarity might be more difficult to build in these cases.

Testing on other problems is also critical for another issue: do the measures perform globally the same on every problem? If not, the genericity of behavioral diversity approaches will be questioned and the expert knowledge on the problem at hand would just be replaced by another kind of expertise. It is interesting to notice that Hamming distance also gave very good results in Gomez's work [12] in a different setup.

Experiments were repeated a low number of times: five times only. It stems from the evolutionary run length. A single run lasted up to four days (the average is around two days) on a 2.4GHz PC. The total evaluation time of the presented experiments is then near 70 days of computation (7 experiments repeated 5 times, each lasting 2 days on average). More runs have to be launched to further confirms the tendencies observed here.

VII. Conclusion

Results show that behavioral diversity approaches can be defined on the basis of simple behavior similarity measures relying on sensor-effector values. Using a behavioral diversity objective with whatever tested similarity measure gave globally better results than not using such an objective. Among the tested measures, the Hamming distance on a roughly discretized vector of sensor and effector values gave better results, but more runs have to be performed to confirm this. Anyway the behavioral diversity using Hamming distance generated near optimal individuals, able to repeat the following sequence: pick up a ball, find the basket, release the ball in the basket, go back picking another ball, until almost all balls are put into the basket, whereas without the diversity objective, individual to collect one ball only at best have been observed. Sequential behaviors were thus generated on the basis of the desired effect only—besides the behavioral diversity objective, the only other objective was a simple count of the balls released in the basket—and without recourse to any fitness shaping or incremental approach.

Behavioral similarity measures revealed to compare favorably to measures made by humans, thus confirming the interpretation of behavioral diversity objectives. Although behavioral similarity measures gave very different results when used to evaluate the behavioral diversity objective, there didn't seem to be a clear correlation between the closeness to human classification and the performance of behavioral diversity experiment, at least not in every case.

VIII. Acknowledgement

The authors thank Benoît Girard, Tony Pinville, Sylvain Koos and Paul Tonelli for their help doing the comparisons.

References

[1] H.A. Abbass and K. Deb. Searching under multi-evolutionary pressures. In *Evolutionary Multi-criterion Optimization*, volume 2632 of *LNCS*. Springer, 2003.

[2] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *Foundations of Data Organization and Algorithms*, volume 730 of *LNCS*, pages 69–84. 1993.

[3] R. Agrawal, K.I. Lin, H.S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *VLDB '95: Proc. of the 21th Int. Conf. on Very Large Data Bases*, pages 490–501, 1995.

[4] L.T. Bui, H.A. Abbass, and J. Branke. Multiobjective optimization for dynamic environments. *In proc. of IEEE-CEC'05*, pages 2349–2356, Sept. 2005.

[5] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998.

[6] G. Das, D. Gunopulos, and H. Mannila. Finding similar time series. In *Principles of Data Mining and Knowledge Discovery*, volume 1263 of *LNCS*, pages 88–100. 1997.

[7] E. D. de Jong, R. A. Watson, and J. B. Pollack. Reducing bloat and promoting diversity using multi-objective methods. *In proc. of GECCO'01*, pages 11–18, 2001.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[9] K.A. DeJong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, Uni. of Michigan, Ann Arbor, 1975.

[10] S. Doncieux and J.-B. Mouret. Single step evolution of robot controllers for sequential tasks. In *Proc. of GECCO'09*, pages 1171–1172. ACM, 2009.

[11] D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *In Proc. of the Second Int. Conf. on Genetic Algorithms and their application*, pages 41–49, 1987.

[12] F.J. Gomez. Sustaining diversity using behavioral information distance. In *Proc. of GECCO'09*, pages 113–120. ACM, 2009.

[13] D.S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343, 1975.

[14] F. Kaplan and V. Hafner. Information-theoretic framework for unsupervised activity classification. *Advanced Robotics*, 20(10):1087–1103, 2006.

[15] J. Lehman and K.O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *Proc. of ALIFE XI. MIT Press*, volume 54, 2008.

[16] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 1966.

[17] Ming Li, Xin Chen, Xin Li, Bin Ma, and P. M. B. Vitanyi. The similarity metric. *Information Theory, IEEE Transactions on*, 50(12):3250–3264, November 2004.

[18] J.-B. Mouret and S. Doncieux. Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *proc. of IEEE-CEC'09*, pages 1161–1168, 2009.

[19] J.-B. Mouret and S. Doncieux. Using behavioral exploration objectives to solve deceptive problems in neuro-evolution. In *proc. of GECCO'09*, pages 627–634. ACM, 2009.

[20] J.B. Mouret and S. Doncieux. Sferes$_{v2}$: Evolvin' in the multi-core world. In *Proc. of IEEE-CEC'10*, 2010. to appear.

[21] S. Nolfi. Evolving non-trivial behavior on autonomous robots: Adaptation is more powerful than decomposition and integration. *Evolutionary Robotics*, pages 21–48, 1997.

[22] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1975.

[23] P.-Y. Oudeyer, F. Kaplan, and V. Hafner. Intrinsic Motivation Systems for Autonomous Mental Development. *IEEE Trans. on Evolutionary Computation*, 11(2):265, 2007.

[24] B. Sareni and Krähenbühl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106, 1998.

[25] O. Sporns and T.K. Pegors. Information-theoretical aspects of embodied artificial intelligence. In *Embodied Artificial Intelligence*, volume 3139 of *LNCS*, page 629. 2004.

[26] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Artificial intelligence: Autonomous mental development by robots and animals. *Science*, 291(5504):599, 2001.

[27] K. Yang and C. Shahabi. A PCA-based similarity measure for multivariate time series. In *MMDB '04: Proc. of the 2nd ACM international workshop on Multimedia databases*, pages 65–74. ACM, 2004.