Some Distance Measures for Morphological Diversification in Generative Evolutionary Robotics

Eivind Samuelsen
Institute of Informatics, University of Oslo
Postboks 1080 Blindern
0316 Oslo, Norway
eivinsam@ifi.uio.no

Kyrre Glette
Institute of Informatics, University of Oslo
Postboks 1080 Blindern
0316 Oslo, Norway
kyrrehg@ifi.uio.no

ABSTRACT

Evolutionary robotics often involves optimization in large, complex search spaces, requiring good population diversity. Recently, measures to actively increase diversity or novelty have been employed in order to get sufficient exploration of the search space, either as the sole optimization objective or in combination with some performance measurement.

When evolving morphology in addition to the control system, it can be difficult to construct a measure that sufficiently captures the qualitative differences between individuals. In this paper we investigate four diversity measures, applied in a set of evolutionary robotics experiments using an indirect encoding for evolving robot morphology. In the experiments we optimize forward locomotion capabilities of symmetrical legged robots in a physics simulation.

Two distance measures in Cartesian phenotype feature spaces are compared with two methods operating in the space of possible morphology graphs. These measures are used for computing a diversity objective in a multi-objective evolutionary algorithm, and compared to a control case with no diversity objective.

For the given task one of the distance measures shows a clear improvement over the control case in improving the main objectives, while others display better ability to diversify, underlining the difficulty of designing good, general measures of morphological diversity.

Categories and Subject Descriptors

I.2.9 [**Computing Methodologies**]: Artificial Intelligence—*Robotics*

General Terms

Design, experimentation

Keywords

Evolutionary robotics, indirect encoding, multiple solutions / niching

GECCO'14, July 12-16, 2014, Vancouver, BC, Canada.

1. INTRODUCTION

In situations where human intervention is difficult or impossible, automatic robot adaptation or repair would be desirable. With the recent and frequent advances in 3D printing technology, such as an increasing number of materials, higher speeds, and portability, new possibilities open up for the design or repair of robotic systems. For example, one could imagine a team of robots including a mobile 3D printer, capable of repairing or producing new robot morphologies in situ [21].

Evolutionary robotics (ER) approaches the challenge of automatic design and adaptation of robotic systems through the use of evolutionary algorithms (EAs); population-based search algorithms inspired by evolution in nature [20]. While ER research has mainly been concentrated on the optimization of robotic control systems, e.g. for legged robots [9], using software simulations it is also possible to address the challenge of simultaneously optimizing robot morphology and control [16].

Automated robot design without a fixed topology introduces an encoding challenge, as more complex data structures are needed to describe the space of possible solutions. Thus, ER research has produced a wide variety of coding schemes for describing morphology. In [13] a tree-based structure is used, where each node represents one of several possible standard modules like joints or variable sized rigid elements. A wide variety of relatively realistic robotic tasks are optimized for, starting from initial populations of hand-made designs.

In [16], simulated locomoting robots with co-evolved morphology and control were reproduced and tested in real life. The reproduction was done using 3D-printing and standard actuator components. The encoding used is direct, and very low-level: The control system is a fully connected, variable size artificial neural network, and the morphology is represented as a number of rigid bars or linear actuators connecting points in space. While this approach succeeded in creating real-life robots capable of forward locomotion, the encoding lacked support for the kind of modularity and structural regularity seen in nature.

In generative and developmental systems, complex phenotypes are generated from simpler genotypes through some sort of indirect encoding. This encoding is used as a developmental program, or set of instructions that can be evaluated to produce the phenotype. These encodings are highly productive - they can produce highly complex and structured results from very terse information [10] and are known to

outperform direct encoding in simple morphology optimization tasks [11].

In the perhaps most well-known example of generative coevolution of morphology and control [24], animal-like creatures are evolved in a virtual environment using a graph-based representation. Morphology is described by means of a directed graph, which may also contain cycles, and advanced morphologies with repeating or fractal structures are achieved. The control system is a network with nodes representing mathematical functions. A variety of behaviors like walking, swimming and following are evolved.

Other generative approaches to evolution of robot morphology and control in virtual environments include gene regulatory network encodings [3], producing robot morphologies consisting of smaller building blocks, and compositional pattern producing network methods [1, 2], resulting in morphologies with complex surfaces.

A generative co-evolutionary approach which also takes the step to producing the evolved results in the real world is presented in [10]. Here, robots are encoded with initial values and production rules for an L-system, which are then interpreted into a morphology and control components. The resulting robots display aspects of symmetry and modularity, and have a higher complexity and fitness than directly encoded robots.

While evolutionary algorithms are effective on large and complex search spaces, they may still suffer from premature convergence and thus it is of interest to encourage an exploratory search based on a diverse population of solutions. Several approaches exist to encourage population diversity, such as niching or fitness sharing [23]. Another efficient approach is to include a diversity measure as a separate objective in a multi-objective setting [4]. This approach has also shown to be efficient in evolutionary robotics, where an added diversity objective can help overcome the so-called bootstrap problem [18]. Another recent approach to evolutionary search is the novelty search, which abandons the fitness and explores purely based on new behavior [14], e.g. solutions which are different than those already explored. This approach can also be combined with performance objectives in hybrid approaches [17, 15].

Common for all of the methods is the need for a distance measure which defines the distance between two solutions. While such a distance can easily be defined as e.g. the Euclidean distance between two constant-sized real-value genomes, it is less straightforward to define in genotype space where the solutions have developed to various sizes and structures, like a neural network graph. Further, for the evolution of robot control systems, the links between the genotypic description, the phenotypic controller structure, and the final behavior may be complex, and thus approaches based on comparing the differences in the resulting behavior may be preferred. In [19], a behavior-based distance outperformed a neural network structure distance in the phenotype space.

While there are some approaches to general distance measures based on behavior [8, 6], these methods assume a fixed robot morphology and operate on clearly defined behavior vectors such as sensory-motor values or leg-ground contact. It is less obvious how such measures would be defined in a context where the morphology changes, e.g. when new limbs with new motors are generated and radically different locomotion strategies appear.

On the other hand, with generative mappings specifying morphology there is an interest for differentiating the solutions on a phenotypic level, as the mapping from genotype to phenotype may be highly nonlinear. One approach to this would be to quantify some properties of the morphologies, such as width, height, and number of joints. These properties were applied in the evolution of virtual creatures in [15], using a novelty search approach.

However, such a list of properties does not necessarily distinguish well between different structural differences. One approach may be to implement more specific morphology descriptors, such as e.g. the number of legs, if this kind of information is available. As several morphology constructions can be seen as graphs, a general graph distance metric could be applied, such as graph probing [19] or the graph edit distance (GED) [7]. However, finding the general GED is an NP-hard problem, and may thus easily become a computational challenge.

In this paper we investigate some distance measures for generative robot morphologies. The distance measures are evaluated in an evolutionary robotics setting, where we evolve simulated robot morphologies for a relatively simple locomotion task, using two different control strategies. We compare a diversity-neutral search with the different distance measures, employed in the diversity objective of a multi-objective evolutionary algorithm.

The first measure is similar to the one proposed in [15], which operates in a three-dimensional Cartesian morphology space. We then define a second measure which also operates in a three-dimensional Cartesian space, but gives a more detailed description of the topology of the body. The next two distances are graph-based measures, one morphology-specific and one which builds on a restricted case of the graph edit distance.

The remainder of this paper is organized as follows: In Section 2 an overview of the generative coding is given, followed by a definition of the distance measures, and then the experimental setup. The results of the experiments are shown in Section 3. Section 4 contains discussion of the results, before a conclusion and pointers for future work are given in Section 5.

2. METHODS

To examine the performance of the four diversity measures, a set of experiments is performed in which the a multi-objective evolutionary algorithm is to maximize forward movement, while minimizing mass and maximizing one of the measures. The four measures are also compared to the basis case of optimizing forward movement and mass alone, with no diversity objective. All five cases are run with two different control system encodings to reduce any bias introduced by the control system.

The evolutionary algorithm used is NSGA-II [5], and the evaluations are done in simulation using the PhysX physics simulation engine. The encoding that is used will be described in Subsection 2.1, followed by a description of the physical parameters and control system parameters that are evolved, in Subsection 2.2. Then the evaluation and objective functions are described in Subsection 2.3, before the diversity measures are detailed in Subsection 2.4. Finally, the general setup of the experiments is explained in Subsection 2.5.

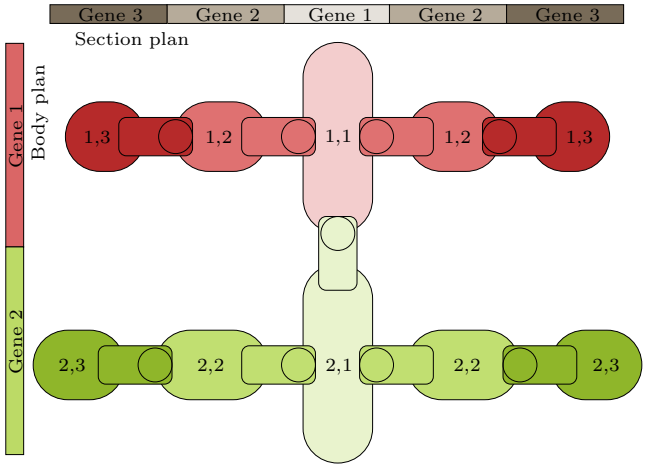


Figure 1: Morphological encoding. The two sets of genes are expressed in different combinations in different parts of the phenotype.

2.1 Encoding

The encoding used in the experiments is based on the one used in [22]. Inspired by how morphology is encoded in animals, the encoding is restricted to creating symmetric bodies with a spine-limb structure by describing a single limb-generating program and a set of parameters specific to each spinal section. To generate the complete morphology, the limb-generating program is instanced once for each spinal section, generating a unique limb consisting of a spinal segment and one or more limb segments by using that sections parameters, as illustrated in Figure 1.

As in [22], the limb-generating program is simply a variable-length list of genes, where each gene encodes the parameters used to describe the morphology of one segment. However, here a more generalized method is used to apply the section parameters during phenotype development. The body plan genes for each section k contain a fixed number of values $h_1^{(k)}, h_2^{(k)}, \ldots, h_N^{(k)}$. These genes encode the values that differentiate different sections, and are encoded differentially: each differentiating value $H_i^{(k)}$ of section k is calculated as

$$H_i^{(k)} = \begin{cases} H_i^{(k-1)} + h_i^{(k)} & \text{if } k \geq 0 \\ 0 & \text{else} \end{cases}$$

All the parameters in the section plan genes are essentially duplicated $N+1$ times to match this, so that each parameter p of each segment becomes encoded as the $N+1$ parameters $p_0, p_1, p_2, \ldots, p_N$ in the genotype. Each expressed value $p^{(j,k)}$ of segment j in section k is then calculated as

$$p^{(j,k)} = p_0^{(j,k)} + \sum_{i=1}^{N} H_i^{(k)} p_i^{(j,k)}$$

where $h_i^{(k)}$ are the actual values stored in the genotype.

All parameters, both in body ans section plans, were initially set to zero and were mutated by applying a normal-distributed random variable with a zero mean to each. The body plan parameters, and the morphology-related and control system-related section plan parameters were mutated using three different standard deviation values, as summarized in Table 1.

Table 1: Evolution parameters

Parameter function	Symbol	Value
Population size	μ	200
Body plan parameter mutation size	σ_H	0.2
Control system parameter mutation size	σ_C	0.1
Morphology parameter mutation size	σ_M	0.05
Gene insertion probability	p_I	0.0025
Gene deletion probability	p_D	0.005

Crossover is done by cloning each parent and matching up genes in each clone by their historical markers. Each gene contains one such marker; they are always inherited unchanged, and new marker values only appear when a mutation causes new genes to appear. Unmatched genes are left untouched by the crossover operator. Crossover of the parameters in a gene is done by whole arithmetic recombination.

During mutation, each gene also has a probability of having a new gene inserted behind it or having its next gene deleted. In these experiments the maximum number of genes in the body plan and section plan is set to 3. This limits the outcomes to a maximum of 3 pairs of limbs and 2 joints in each limb. This limits the size of the morphology search space, and stops the diversity measures from promoting individuals with an increasingly large number of joints.

This was done partly because we wanted to focus the search on robots that were easy (and not too costly) to construct in real life, and partly because initial experiments indicated that removing these limits would require an increase in population size beyond what could be afforded for the experiments presented here.[1] The limits are implemented by setting the insert probability to 0 when the number of genes has reached its maximum. Otherwise the insertion and deletion probabilities are as shown in Table 1.

2.2 Parameters and Control Systems

The section plan genes encode a small number of physical parameters. The spine segment, of which there is always exactly one in the section plan, has two parameters: the segment length l and the side offset o that decides how far along the side of the spine segment the first limb segment will appear, if present. The limb segments, which constitute the rest of the section plan, has three parameters: the segment length l, the axis-of-rotation angle A_ϕ that controls the twist of the limb joint compared to the previous joint, and the rest angle A_ψ that adds an offset to the angle which the joint will start the simulation in, and which is considered the zero point in the control systems.

As all parameters were coded as real numbers with no limits, the segment length and the side offset were translated

[1]The two graph-based measures, at least, will intuitively strive to spread the population about equally out across all $\sum_{n=1}^{N} M^n$ possible topologies, where N is the maximum number of sections and M is the maximum number of segments in each section, including the spine. $N = 3$, $M = 3$ gives 39 possibilities. Increasing both by only 1 would result in 340 possibilities, so the population would have to be increased dramatically to ensure that candidates from the 39 topologies in the smaller search space are properly represented.

using the formulas

$$\begin{aligned} L &= L' \cdot \exp(l) \\ O &= 1/(1+\exp(-o)) \end{aligned}$$

in order to map them into more suitable ranges ($(0,\infty)$ and $(0,1)$ respectively). Here, L' is the length of the previous segment on the limb for limb segments, or the previous spine segment for spine segments, so that the length is coded differentially. In the case of the first spine segment L' was set to 0.1 m. If the segment length is less than what is required to contain the necessary servo motor, the segment, and any following segments in the section plan would not be expressed. This mechanism enables different sections to have different number of limb segments. To be consistent with this, then if the infeasibly short segment is a spine segment, the following sections in the body plan would not be expressed either.

The control system parameters are a bit more complex. One of two control systems is used in the experiments. Both are simple open-loop systems, and produce periodical signals with a period of 1 s. The first control system is a simple amplitude-phase system, as used in [12], which can be expressed as

$$C_{\alpha\phi}(t) = \alpha \tanh(4\sin(2\pi(t+\phi)))$$

In principle, this control system has two parameters per joint, but here the amplitude is coded symmetrically along with the morphology, and the phases are coded differentially, reducing it to three parameters per joint pair: amplitude, phase, and phase offset for the second joint in the pair.

The second control system is a cyclic cubic B-spline with four knots, located at $t_0 = 0$, $t_1 = 0.25$, $t_2 = 0.5$ and $t_3 = 0.75$, with a period of one. Strictly speaking the B-spline then only needs four control values to be fully defined, but here two differentially coded phase parameters are added to be able to ensure that smooth phase changes are likely to occur during mutation. The encoding is symmetrical like the amplitude-offset system, resulting in six parameters per joint pair.

2.3 Evaluation

Each individual in the population is evaluated by simulating its behavior for 8 simulated seconds. After 1 s of settling time has passed, the position of the robot is reset to a fixed starting point in the environment. After the remaining 7 s have passed, the distance traveled by the robot is measured as the displacement of the tip of the front spine segment along the axis that segment was pointing at the beginning of the simulation. The mass objective value uses the mass calculated by the simulator.[2]

In order to make the task more challenging, low, wide obstacles were placed in the simulated environment at regular intervals. The obstacles were 2 cm tall and 2 cm deep, and extended 10 m in each direction. The first obstacle was put 0.5 m in front of the robot, with a new obstacle appearing every half meter after that. The thickness of the robot limbs were set to 4 cm, so the obstacles only add a degree of unevenness to the ground, forcing the robots to lift themselves up slightly to move forward past the first half meter.

The morphology of the robots was variable to a very large degree, making infeasible constructions probable. To minimize this problem robots that had no joints, or that clearly did not simulate properly (e.g. robots that resulted in movement scores ≥ 1 km) were removed before the survival selection.

2.4 Diversity

The diversity of an individual x in the population P_n is the average distance to all other individuals in the population, which can be written as:

$$D_d(x) = \frac{1}{|P_n|-1} \sum_{y \in P_n} d(x, y)$$

Here, d is the distance measure used. This paper compares four different distance measures, operating in different phenotype spaces. Two of these are Cartesian spaces, both mapping the robot morphology to points in $\mathbb{R}^3$, with each of the three axes corresponding to a scalar feature of the morphology. In the first measure, the three features are the height, mass and number of joints (HMJ), as seen in [15], except the Euclidean distance is used as the distance value instead of its square:

$$d_{HMJ}(x, y) = \|HMJ(x) - HMJ(y)\|$$

The three features in the second measure, length-branches-depth (LBD), is the sum of segment lengths, the number of branches in the robot topology (i.e. the number of sections with at least one limb segment), and the depth of the longest branch (i.e., the maximum number of joints in any limb). As with the first measure, the Euclidean distance is used:

$$d_{LBD}(x, y) = \|LBD(x) - LBD(y)\|$$

The phenotype space of the other two is a graph space, so other distance metrics have to be used. The first measure is the degree-2 graph edit distance (GED), as explained in [25], which limits the graph edit distance search to only inserting and removing nodes with no more than two neighbors. The insert and remove costs are $\gamma(\Lambda \to v) = 1$ and $\gamma(u \to \Lambda) = 1$, and the relabel cost is

$$\gamma(u \to v) = 1 - \exp(-|L_u - L_v|)$$

where l_x is the length of segment x. This way the differences between two matched segments can never be greater than the difference from removing or inserting a segment. The time complexity of this algorithm is bounded by $O(NME^2)$, where N and M are the number of nodes in the two graphs and E is the maximum number of outgoing nodes in any node in the two graphs. Here, E is guaranteed to be less or equal to 3, resulting in a time complexity of $O(NM)$, but with a large constant factor, because each of the NM operations are quite expensive.

The second graph measure exploits the fact that all evolvable morphologies are symmetric and can be laid out in two dimensions with coordinates as shown in Figure 1. By doing a topology sweep (TS), an approximation of the graph edit distance can be found as the best correlation between two-dimensional images of the topologies. Each segment is given a coordinate (i, j), where i is the index of the body section the link belongs to (e.g. all segments in the head

[2]The simulations was done using PhysX version 3.3 beta-2, with a fixed timestep of 1/128 s. The robot models and obstacles were constructed with capsule and box primitives and the joints were free 1D revolute joints with additional custom constraints to simulate internal friction and torque exerted by a basic DC motor model.

section have $i = 0$), and $|j|$ is the distance to the spine in number of segments. The sign of j is positive on the right side and negative on the right. The finds the difference of the best match between the two morphologies by summing up the cost of replacing the segments in the first morphology with the segments of the second, displaced by some k along the i axis:

$$d_{TS}(x, y) = \min_k \sum_i \sum_j \gamma\left(s^x_{i,j} \to s^y_{i+k,j}\right)$$

if body x has no segment at position i, j then $s^x_{i,j} = \Lambda$. $\gamma(\Lambda \to \Lambda)$ is naturally zero. The implementation of this measure used here is not very efficient, using $O(NMS)$ time, where S is the number of sections in the longest robot. Since S here is less or equal to 3, this reduces to the same complexity as the GED, however, the operations are much cheaper here, resulting in run times comparable to that of the Cartesian measures.

2.5 Experimental Setup

Each diversity measure, along with the diversity-less control case was tested with both control systems in order to reduce any bias introduced by the specific control system used. For implementation reasons, the runs with no diversity measure were also run with a third objective, which would always evaluate to zero, thus effectively reducing the algorithm to two-objective optimization. In total, there is 10 different variants of diversity measure and control system. In these experiments, each variant was run 21 times. At the end of each generation, the objective values of each surviving individual were recorded. The number of branches and the depth features of each individual, as used in d_{LBD}, is also recorded for later analysis. Each run was terminated after the 600th generation. At the end of each run, the average diversity of the final population is recorded, as measured by all four diversity measures.

The population is initialized with genotypes based on a prototype with two body plan genes and three section plan genes. The segment length parameters were $l_0 = 1$, $l_1 = 0.9$ and $l_1 = 1.1$, resulting in segment lengths of 10 cm, 9 cm and 9.9 cm respectively. The other parameters of the first two segments were zero, while for the outer limb segment, the parameters were $A_\phi = \pi/2$ and $A_\psi = -\pi/6$. The robot generated by these parameters is shown in Figure 2. All control system parameters are initialized to zero. To generate the initial population, this prototype is then cloned μ times, and then each of the clones are mutated 10 times.

To measure how the diversity measures affects the main objective of forward locomotion, a linear regression model is estimated that predicts the 90th percentile forward moving individual[3] of the population at a given generation. Even though considerable effort has been spent to implement constraints so that impractical or faulty solutions are not scored well, some will always slip through[4] and appear as outliers on the fringes of the population. Thus the 90th percentile is used because it shows the performance achieved by the best individuals, while being more robust against outliers than the single best individual. The model is on the form

$$\hat{s}_U = \alpha_0 + \alpha_C C + \sum_{V \neq U} \alpha_V D_V$$

Here, U is the diversity case (i.e. one of the diversity measures or no diversity measure) used as the basis for the model and C is one for all runs with the cubic spline control system, and zero otherwise. D_V is one for all runs where diversity case V was used, and zero otherwise. The $\alpha_C C$ term of the model accounts for systematic differences in score between the two control systems. Setting $C = 0.5$ gives a model for the average expected score for the two control systems. α_0 is the expected score of diversity case U with the amplitude-phase control system, and α_V is the expected difference between cases U and V. This model is robust in the sense that when applied the predicted score will be the same regardless of the choice of U.

[3]Since the population size was 200 in all the experiments, this corresponds to the 20th best individual ranked by that objective.

[4]For example, the runs using the HMJ measure were vulnerable to extremely tall robots that just tilted very slowly forward, since the height component of the diversity measure encouraged very tall designs.

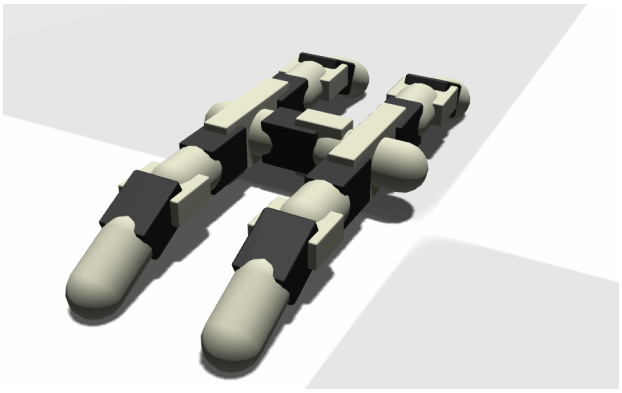


Figure 2: The initial prototype robot.

Table 2: P-values of pairwise T-tests on diversity measure forward movement performance differences after generation 600

	None	LBD	HMJ	TS	GED
LBD	0.0001				
HMJ	<0.0001	<0.0001			
TS	<0.0001	<0.0001	0.0002		
GED	<0.0001	<0.0001	0.0221	0.1553	

3. RESULTS

The expected forward locomotion performance of the different diversity measures over generations are shown in Figure 3. Table 2 shows the p-values of the T-test on the different pairwise differences in performance, as found by calculating $\hat{s}_U$ for all cases. The matrix is symmetric, so only the lower triangular part is supplied.

The control case performs better ($p < 0.01$) than the cases with a diversity measure approximately from generation 50 to generation 100, where the LBD measure catches up. This measure then outperforms the control case approximately from generation 200 until termination, and has been measurably better than the other measures since generation 10. The other measures remain inseparable until generation 450, when the HMJ measure becomes measurably worse than TS. HMJ becomes different only by a weak significance to GED,

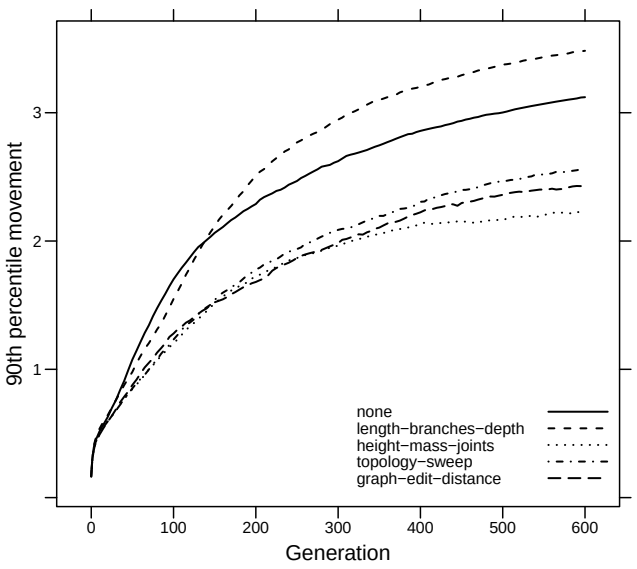


Figure 3: (Expected) average performance of the algorithm as measured by the 90th percentile forward moving individual for the different diversity measures after different generations.

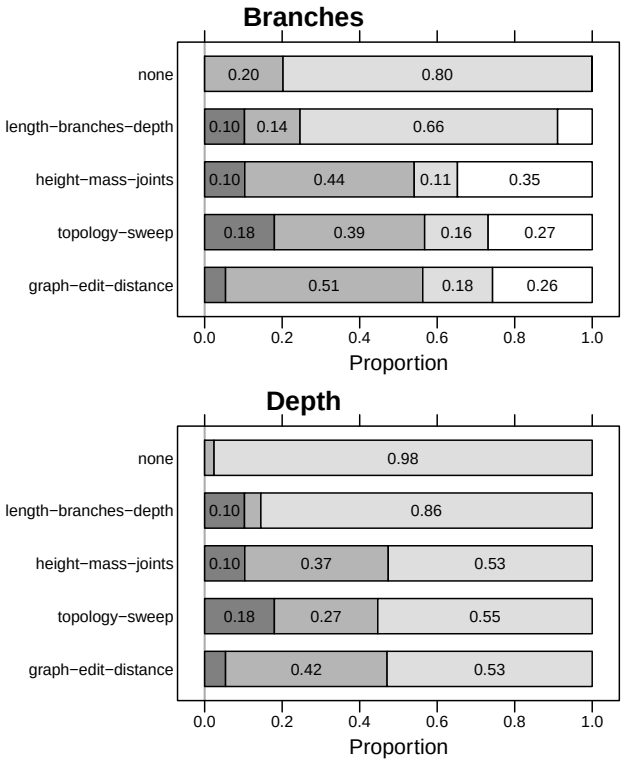


Figure 4: Limb length and branch count distribution - dark gray is 0, middle gray is 1, light gray is 2 and white is 3. The number inside the boxes is the corresponding proportion, no number means less than 0.10.

with p-values in the range of 0.02 to 0.05 after generation 480. TS and GED distance remain inseparable, statistically speaking, until the end.

Figure 4 shows the distribution of number of branches and depth of the different diversity cases. The values are found using regression models similar to the one used for the forward movement score for each branch or depth value, but with a quasibinomial model instead of a linear one because the numbers are proportions. It can be seen that while HMJ, TS and GED all have a large proportion of 1 and 3-branched robots (i.e. bipeds and hexapods) and a smaller proportion of 2-branched robots, the diversity-less case and LBD produce dominantly 2-branched robots. Similarly, when looking at the depth distribution the control case and LBD mostly produce robots with two-jointed limbs, while for the other three cases the distribution between one and two-jointed limbs is much more (but not quite) even.

The final population diversity in the different cases as measured by all four diversity measures is shown in Table 3. In all measures the diversity of the control case is much lower than in the other cases. The diversities measured by TS and GED are similar in all cases, and the runs evolved with these two measures also result in similar values in all measurements. The two Cartesian distance measures show a different behavior. Both measures rate runs with HMJ diversity highly and runs with both are rated highly by HMJ, but runs with LBD rate much lower with by its own measure.

Figure 5 shows three examples of robots produced by the algorithm. The robot in Figure 5a has 1 branch and a depth of 2, and locomotes using a rolling motion. Since the obstacles are wide and relatively low, the robot is able to roll over it and stay on course. Figure 5b shows a tripod morphology (1 branch, depth 1) were the limbless rear spine segments are used as a third leg that helps push the robot forward. In Figure 5c one can see a much larger morphology with 2 branches and a limb depth of 2. This robot uses a gait where all four limbs move synchronously to lift the body up and forward. When the robot reaches the obstacles it gains extra traction from pushing against them with the rear limbs. Some variants of this morphology is able to propel itself far enough forward to grab on to a new obstacle each period, thus reaching a speed of 0.5 m/s.

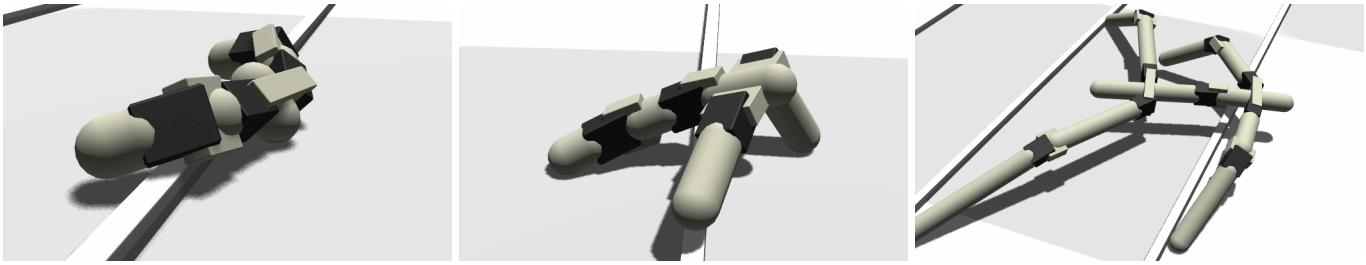
4. DISCUSSION

From the experimental results, we can make the following observations:

- Using diversity with the LBD distance gives the best-performing forward movement results, as seen in Figure 3, and thus LBD-based diversity search seems to explore most efficiently the set of useful morphologies for the given task. We hypothesize the following: In

Table 3: Mean cross-measure diversity of the final populations

		Measured by			
		LBD	HMJ	TS	GED
Evolved with	None	1.69	3.14	1.22	1.54
	LBD	3.66	9.00	3.05	3.17
	HMJ	8.18	8.94	5.90	6.28
	TS	5.62	6.76	5.74	5.80
	GED	5.54	5.99	5.23	5.71



(a) Rolling robot climbing the low obstacles (b) Robot with successful tripod gait (c) Robot pushing itself from the obstacle behind it

Figure 5: Some example results

the given scenario, the most efficient solutions tend to be quadruped configurations, as seen in Figure 4. Optimization of the LBD diversity may lead to a larger variation in the total length of the robot limbs, as there is a limited number of topologies available for the branches and depth combinations. The LBD measure may thus tend to explore different lengths within a quadruped configuration, leading to efficient solutions for the task, as opposed to the other measures which may emphasize less significant components.

- For the given task, it is clear from Figure 3 that even though LBD diversity gives the best forward movement performance, the control case of no diversity still gives significantly better performance than diversity using the other measures. One reason for this may be that the more complex distances lead to a wider exploration of the search space, and thus limit exploitation, while in the current task such a wide exploration is not necessary in order to achieve good results. It is however possible that more complex and deceptive tasks may require a wider search and thus favor search based on diversity with the more complex, topology-based distances.

- While the GED distance gives a theoretically better measure of difference between topology graphs than the TS distance, we can see from Table 3 that the two distances produce relatively similar values, with TS being slightly higher on all measures. The slight favor of TS is also indicated by the even distributions in Figure 4. The preferable graph-based distance metric would therefore be TS, also taking into consideration that GED is computationally much more complex. As an example, the evolutionary experiments using GED diversity required about three times longer runtime than the other diversity experiments.

- From Table 3 one can observe that there are complex interactions between the distance metrics. For instance, evolving with the HMJ diversity will lead to high diversity scores over all measures, even outperforming the results of evolving directly with other measures. At the same time, evolving with LBD gives a very high HMJ score, but poor scores in the other measures. The mechanisms behind this seem complex; however we relate it to different morphology components getting different emphasis depending on the distance measure, directing the search in different directions.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented some distance measures for a generative robot morphology encoding, and analyzed their performance across two control systems in a simulated environment. For the given locomotion task with small obstacles, one of the distance measures shows a clear improvement over the plain evolutionary approach. Since the performances of the different methods vary considerably, an evaluation of the distance measure to employ according to the task at hand seems appropriate.

Most of the measures are morphology-specific, in principle requiring a rigid, acyclic graph-like morphology, limiting the scope of these results somewhat. However, the restrictions may not be as strict as they appear; many rigid generative morphologies should be translatable into acyclic structures on a functional level (e.g. by collapsing rigid structures and then collapsing duplicate edges (joints) between nodes), with differences in rigid structures handled by relabeling costs, and techniques may possibly be found to do the same for non-rigid morphologies as well (some Voronoi diagram-inspired approach, perhaps).

While the proposed diversity measures indeed show promising results, the search was limited to a relatively small set of possible topologies. It would be perhaps be prudent to verify the results with looser restrictions on the number of limbs and joints per limb. There are also other aspects about the morphological phenotype spaces that could be explored. For instance, the current distance measures do not take into account the joint angles, which may for otherwise similar topologies have a great impact on the resulting behavior. Similarly, the control systems are co-evolved with the morphology and should thus be taken into consideration. While there is no obvious approach to measuring behavior together with generative morphologies, as mentioned in Section 1, it would be highly interesting to explore measures which would give more information of the combined morphology and control.

In order to fully explore the possibilities and limitations of the diversity methods, and the capabilities of the generative system, it would be necessary to experiment with even more complex tasks, such as environments with more difficult obstacles. This could again lead to more complex morphologies, as indicated in [2]. Furthermore, the generality of the results should be verified for a range of different encodings, environments and tasks, and also over a wider range of control systems.

6. REFERENCES

[1] J. E. Auerbach and J. C. Bongard. Evolving cppns to grow three-dimensional physical structures. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 627–634. ACM, 2010.

[2] J. E. Auerbach and J. C. Bongard. On the relationship between environmental and morphological complexity in evolved robots. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, GECCO '12, pages 521–528, New York, NY, USA, 2012. ACM.

[3] J. Bongard. Evolving modular genetic regulatory networks. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1872–1877. IEEE, 2002.

[4] E. D. de Jong, R. A. Watson, and J. B. Pollack. Reducing bloat and promoting diversity using multi-objective methods. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 11–18. Morgan Kaufmann, 2001.

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

[6] S. Doncieux and J.-B. Mouret. Behavioral diversity measures for evolutionary robotics. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[7] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010.

[8] J. C. Gomes and A. L. Christensen. Generic behaviour similarity measures for evolutionary swarm robotics. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2013*, pages 199–206. ACM, 2013.

[9] G. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto, and M. Fujita. Evolving robust gaits with aibo. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 3, pages 3040 –3045 vol.3, 2000.

[10] G. S. Hornby, H. Lipson, and J. B. Pollack. Generative representations for the automated design of modular physical robots. *Robotics and Automation, IEEE Transactions on*, 19(4):703–719, 2003.

[11] M. Komosinski and A. Rotaru-Varga. Comparison of different genotype encodings for simulated three-dimensional agents. *Artificial Life*, 7(4):395–418, 2001.

[12] S. Koos, A. Cully, and J.-B. Mouret. Fast damage recovery in robotics with the t-resilience algorithm. *International Journal of Robotics Research*, 32(14):1700–1723, 2013.

[13] C. Leger. *Automated synthesis and optimization of robot configurations: an evolutionary approach*. PhD thesis, Carnegie Mellon University, 1999.

[14] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on Artificial Life (ALIFE XI)*, pages 329–336, 2008.

[15] J. Lehman and K. O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218. ACM, 2011.

[16] H. Lipson and J. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978, 2000.

[17] J.-B. Mouret. Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics*, pages 139–154. Springer, 2011.

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

[19] J.-B. Mouret and S. Doncieux. Using behavioral exploration objectives to solve deceptive problems in neuro-evolution. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 627–634. ACM, 2009.

[20] S. Nolfi, D. Floreano, and D. Floreano. *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*, volume 26. MIT press Cambridge, 2000.

[21] S. Revzen, M. Bhoite, A. Macasieb, and M. Yim. Structure synthesis on-the-fly in a modular robot. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4797–4802. IEEE, 2011.

[22] E. Samuelsen, K. Glette, and J. Torresen. A hox gene inspired generative approach to evolving robot morphology. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 751–758. ACM, 2013.

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

[24] K. Sims. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 15–22, New York, NY, USA, 1994. ACM.

[25] K. Zhang, J. T.-L. Wang, D. Shasha, et al. On the editing distance between undirected acyclic graphs. *International Journal of Foundations of Computer Science*, 7(1):43–58, 1996.