

Comparing the evolvability of generative encoding schemes

Danesh Tarapore and Jean-Baptiste Mouret

Sorbonne Universités, UPMC Univ Paris 06, UMR 722, ISIR, F-75005, Paris, France

CNRS, UMR 7222, ISIR, F-75005, Paris, France

danesh.tarapore@gmail.com

Abstract

The evolvability of a system is the ability to generate heritable, novel and non-lethal phenotypes, from random genetic mutations. However, most evolutionary computation studies estimate evolvability either as, (i) the proportion of mutations beneficial to an individual's performance, irrespective of the phenotypic diversity of the mutants, or (ii) the range and diverseness of mutated phenotypes, without taking into account the viability of the genetic change. This paper reports a novel approach to measure the evolvability provided by an encoding, by characterizing both the quality of the mutations and the quantity of phenotypic variation. We evolved controllers for hexapod robot locomotion using a parameterized direct encoding, and the generative encoding of artificial neural networks (similar to HyperNEAT) and single-unit pattern generators (SUPGs). Our results reveal that the performance of an encoding is not always a good assessment of evolvability. Although both the generative encodings evaluated had individuals with high performance gaits, there were apparent differences in their measured evolvability. A direct and predictive relationship is indicated between our measure of evolvability, and the number of generators required by amputated individuals to recover an effective gait.

Introduction

The capacity of a population to rapidly adapt to novel environments, called evolvability, is critical in the evolutionary processes of natural organisms and artificial systems (Hu and Banzhaf, 2010). In the natural world, the high evolvability of biological systems is hypothesized to be responsible for the rich diversity of the millions of species creatively adapting to diverse niches, all arising from a combination of random mutations and natural selection (Pavlicev and Wagner, 2012). Evolvability is also of interest in evolutionary computation (EC), wherein highly evolvable solutions are considered to optimize faster and achieve a greater performance than non-evolvable solutions (e.g., Cheney et al. (2013)), and may be capable of generalizing previously learned information and adapting it to new environments (e.g., Reisinger et al. (2005); Clune et al. (2013)), desirable characteristics from an engineering perspective.

The developmental mechanisms translating genetic change into phenotypic change, that is the genotype-

phenotype map, is commonly understood to play a key role in the evolvability of evolutionary systems (Kirschner and Gerhart, 1998; Wagner and Altenberg, 1996). Mappings facilitating evolvability, confer on the individual a robustness to lethal mutations, and exhibit a modular architecture wherein genes preferably only affect traits with the same function (Pavlicev and Wagner, 2012). Inspired by the evolvability of biological systems, researchers in EC have abstracted the underlying developmental processes, to formulate generative genotype-phenotype maps for artificial systems (e.g., Stanley (2007)). The resultant generative encodings frequently outperform traditional direct encodings for various application problems such as, designing 3D objects (e.g., Hornby (2005)), game playing (e.g., Reisinger and Miikkulainen (2007); Gauci and Stanley (2010)), pattern matching (e.g., Clune et al. (2011)), and robot locomotion (e.g., Hornby and Pollack (2002); Seys and Beer (2007)). Furthermore, the higher evolvability provided by generative encodings is often considered as the reason for the observed differences in performance, consequent to their capability to reuse parts of the genotype to affect different phenotypes, scale well to large phenotypic spaces, and generate modular architectures (Stanley and Miikkulainen, 2003).

In the field of EC, there are two main approaches to estimate the evolvability of artificial developmental systems. Most studies estimate evolvability either as, (i) the proportion of genetic mutations that are beneficial to an individual, irrespective of the phenotypic novelty of the resultant offspring (e.g., Hornby et al. (2003); Reisinger and Miikkulainen (2007)), or as (ii) the range and diversity of the phenotypic variants resulting from genetic change (Lehman and Stanley, 2011; Reisinger et al., 2005; Lehman and Stanley, 2013), usually without considering the deleteriousness of the change. Importantly, both these estimates when considered alone do not discount for mutations that, (i) generate very diverse phenotypes but prove lethal to an organism, and (ii) result in minor improvements to a phenotype, but are unable to generate novelty. According to the theory of facilitated variation (Gerhart and Kirschner, 2007), the capacity of an

individual to evolve is a function of its ability to curtail lethal mutations, and simultaneously to decrease the number of mutations necessary to evolve diverse or novel phenotypes.

In this study we present a novel approach to visualize evolvability provided by direct and generative encodings, featuring both the quality and quantity of phenotypic variation consequent to genetic change, for hexapod robot locomotion. Evolvability is analyzed for a parametrized direct encoding, and the generative encoding of artificial neural networks (similar to HyperNEAT, Stanley et al. (2009)) and single-unit pattern generators (Morse et al., 2013), in three independent experiments. The significance of our measure of evolvability is analyzed by the ability of the robot to adapt to previously unencountered changes in its morphology.

Measuring evolvability

The evolvability provided by the direct and generative encodings is measured by computing the effect of genetic mutations on, (i) the viability of the mutated individual, and (ii) the diversity of phenotypes generated. The two resultant effects are treated separately instead of being combined into a single quantitative measure of evolvability, to consider the trade-offs between them in their individual influence on evolvability.

Feature 1: Deleteriousness of mutations. The first feature in our measure of evolvability is computed as the proportion decrement in the fitness of a mutated individual.

For an individual i and the mutant i' , we have,

$$f_1 = (F'_i - F_i)/F_i \quad (1)$$

where F_i and F'_i , are the fitness values before and after the application of a random genetic mutation, respectively.

The feature f_1 reflects the phenotype quality following beneficial ($f_1 > 0$), neutral ($f_1 \approx 0$), and deleterious ($f_1 < 0$) genetic change.

Feature 2: Diversity of phenotypes. The second feature in our measure of evolvability is quantified as the difference in the phenotype, resulting from genetic mutation. There are many available diversity metrics to measure behavioral differences, ranging from the classical euclidean distance, to correlation coefficients, and various information theoretic measures (reviewed in Mouret and Doncieux (2012)). The mutual information metric, in comparison with other measures such as correlation, provides a more general criterion to investigate statistical dependences between behaviors, and is applicable to numerical and symbolic phenotypic representations.

We compute the phenotype diversity as the normalized mutual information (Cover and Thomas, 1991) between behaviors of an individual, before and after its genome is mutated. Assuming that the behavior of an individual i can be represented as a discrete vector B_i (details in Mouret and Doncieux (2012)), for the behaviors B_i and B'_i , of individual i and mutant i' , we have:

$$H(B_i) = - \sum_{b_i \in B_i} p(b_i) \log p(b_i) \quad (2a)$$

$$H(B_i, B'_i) = - \sum_{b_i \in B_i} \sum_{b'_i \in B'_i} p(b_i, b'_i) \log p(b_i, b'_i) \quad (2b)$$

$$f_2 = 1 - \frac{H(B_i) + H(B'_i) - H(B_i, B'_i)}{\max(H(B_i), H(B'_i))} \quad (2c)$$

where $H(B_i)$ is the entropy of the behavior B_i comprising the individual states b_i with probability $p(b_i)$, $H(B_i, B'_i)$ is the joint entropy between behaviors B_i and B'_i with joint probability density function $p(b_i, b'_i)$, and f_2 denotes the inverse of the normalized mutual information between the two behaviors.

The feature f_2 in our measure of evolvability represents the quantity of phenotypic variation following genetic change, and is indicative of the ability of the evolutionary system to produce novel phenotypes.

Hexapod locomotion problem

The evolution of locomotion gaits for multilegged robots is a classical problem in EC, addressed in many prior studies utilizing both direct and generative encodings (e.g., Liu and Iba (2004); Clune et al. (2011); Valsalam and Miikkulainen (2008)). In the large majority of these studies, the performance of evolved individuals is analyzed solely by the walking speed of the robot and the required number of generations of evolution. The rate of evolution and evolved performance has also been linked to evolvability provided by the encoding scheme, wherein controllers achieving a higher task fitness and requiring fewer generations to evolve are considered more evolvable (e.g., see Hornby et al. (2003); Komosiński and Rotaru-Varga (2001)). While these approaches provide interesting insights on the characteristic of the underlying genotype-to-phenotype mapping, they largely ignore its capabilities to generate viable phenotypic variations (diverse gaits in case of legged robots). However, the diversity of evolved walking gaits is important for the legged robot to recover rapidly from faults such as, the loss of one or more limbs, or motor malfunctions (Koos et al., 2013), and for the robot to adapt to previously unencountered environmental changes. Furthermore, an efficient recovery is particular relevant for hexapedal legged robots, wherein the probability of component failure is high, consequent to the large number of moving parts.

Hexapod platform details: The hexapod robot is simulated on a flat, horizontal surface (Fig. 1a), with the Open Dynamics Engine¹ (ODE) physics simulator. The robot has 18 Degrees of Freedom (DOF), 3 for each leg (Fig. 1b), and each DOF is actuated by a single servo. The first servo on each leg (s_1) actuates the horizontal orientation of the leg within range $[-\pi/8, \pi/8]$ radians. The second (s_2) and third

¹<http://www.ode.org>

(s_3) servos control the leg elevation and extension, respectively, each within the range of $[-\pi/4, \pi/4]$ radians.

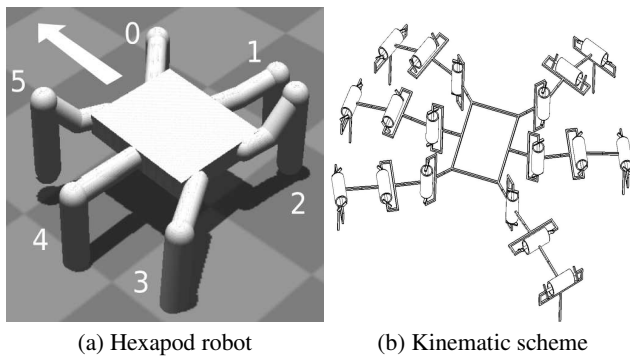


Figure 1: (a) Snapshot of an 18-DOF simulated hexapod robot walking on a horizontal surface, with contacts simulated. (b) Kinematic scheme of the robot, with cylinders representing actuated pivot joints. The three servos on each leg, s_1 , s_2 and s_3 , are labeled in increasing order of distance to robot torso.

Angular positions are sent to the 18 servos once every 15 ms. Furthermore, in order to maintain the last subsegment of each leg vertical (for enhanced stability), the control signal for the third servo (s_3) is always in antiphase to that of the second servo (s_2). Consequently, the robot is reduced to a 12 DOF system, despite being actuated by 18 motors.

Hexapod gait representation: The phenotypic diversity in our measurement of evolvability corresponds to the inter-gait diversity in the hexapod robot locomotion problem. For this measurement, a hexapod gait is represented using a gait diagram (Kajita and Espiau, 2008, p. 379), comprising a binary matrix C of leg-surface contacts:

$$C_{it} = \begin{cases} 1 & \text{if leg } i \text{ makes surface contact at time-step } t, \\ 0 & \text{otherwise.} \end{cases}$$

where $t \in \{0 \dots T\}$, the gait is evaluated for T time-steps, and the hexapod legs $l \in \{0 \dots 5\}$.

The hexapod gait for an individual i is represented by binary vector B_i , comprising the contacts in C concatenated in row-major order, $B_i = [C_{00}, C_{10} \dots C_{T5}]$. The difference between two gaits is measured as the normalized mutual information between the corresponding gait vectors (eq. 2c).

Encoding schemes

The generative encodings for evolving hexapod locomotion controllers are based on compositional pattern producing networks (CPPNs) (Stanley, 2007). CPPNs abstract the processes of embryonic development by determining the attributes of phenotypic components as a function of their geometric location in the individual, instead of simulating complex inter-cellular interactions.

The CPPN genome is represented as a directed graph, comprising a set of *Sine*, *Gaussian*, *Sigmoid*, and *Linear* type of nodes, connected by weighted links. The node type indicates the activation function applied to the sum of its weighted inputs, to compute the node output. Selected activation functions can succinctly encode a wide variety of phenotypic patterns, such as symmetry (e.g., a Gaussian function) and repetition (e.g., a Sine function), that evolution can exploit. Mutations to the CPPN genome can change the connection weights and node type, and add or remove nodes from the graph. Consequently, the topology of the CPPN is unconstrained, and can represent any possible relationship between the input coordinates of the phenotypic component and its output attributes (see details in Stanley (2007)).

In this study, the CPPN genotype is mapped to two very different phenotypes, the conventional Artificial neural networks (ANNs), and the Single-unit pattern generators (SUPGs). The SUPG is a new type of macro-neuron introduced by Morse et al. (2013) to genetically encode spatio-temporal oscillatory patterns.

Encoding ANNs with CPPNs (minimal HyperNEAT):

The first generative encoding scheme evaluated is a simplified version of HyperNEAT indirect encoding² (Stanley et al., 2009; Clune et al., 2011; Yosinski et al., 2011). The CPPNs encode the weights of a fixed topology, single-layer feedforward ANN, featuring 2-D cartesian grids of inputs, hidden and output neurons (Fig. 2). Neurons are placed in a geometric space termed the *substrate*, so that each neuron in a layer has a distinct (x, y) coordinate. In addition, the placement of the input and output neurons is meant to reflect the hexapod robot morphology. The CPPN is iteratively input the positions of all source (x_1, y_1) and target (x_2, y_2) neurons in proximal layers, along with a constant bias, and it outputs the corresponding weights of the input-hidden and hidden-output neuron connections (see Fig. 2).

The ANN receives as input the previously requested angles (actual angles unknown) for each of the 12 pivot joints of the robot (s_1 and s_2 , for 6 legs). In addition, sine and cosine waves of frequency 1 Hz are also input to the ANN, to facilitate output periodic oscillations. The output from the ANN at each time-step are 12 numbers (one for each of s_1 and s_2 , on each of 6 legs) in interval $[-1, 1]$, that are scaled to the allowable angular range of the corresponding motors, and indicate the next position of each motor.

In preliminary experiments, this encoding evolved ANNs that exhibited high frequency output oscillations (in excess of 20 Hz). In the resultant gaits, the robot could traverse large distances by vibrating its legs rapidly, and in unison. To resolve this problem, and as suggested by Yosinski et al. (2011), we generated joint angular positions with a time-

²The CPPN is evolved with a simple multi-objective evolutionary algorithm, instead of the NEAT method (details in Tonelli and Mouret (2013)).

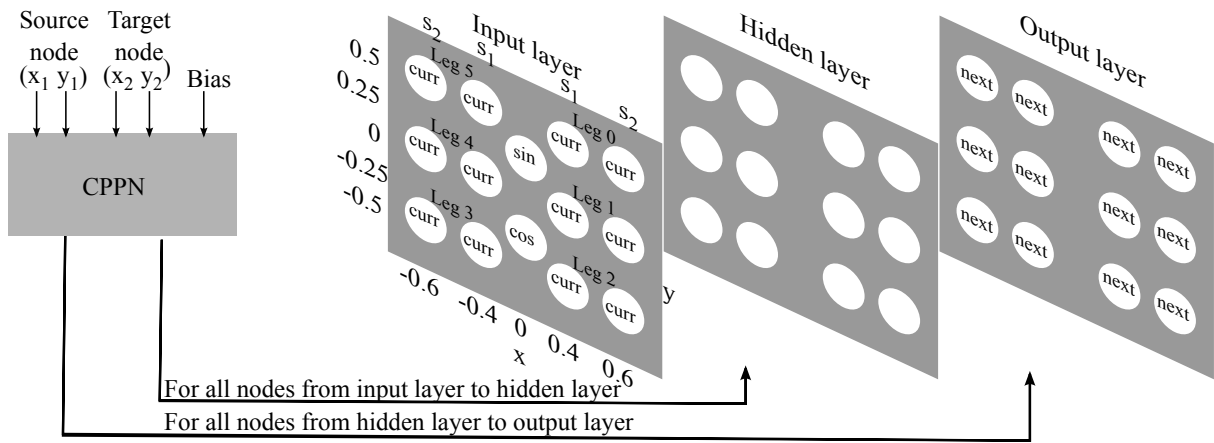


Figure 2: Encoding ANNs with CPPNs (inspired by Clune et al. (2011); Yosinski et al. (2011)). Inter-neuron connection weights are encoded as function of the position source and target neurons of each neural connection. The CPPN outputs the weights of input-hidden and hidden-output neuron connections, for each source (x_1, y_1) and target neuron (x_2, y_2) in proximal layers. The ANN is input the requested angles of the previous time-step for the first two servos (s_1 and s_2) on each leg, and a sine and cosine wave. The output neurons specify the new joint angles for the current time-step.

step of 15 ms, by averaging over four consecutive pseudo-positions generated at 3.75 ms intervals. The number of evolved ANN controllers outputting high frequency oscillation was thus reduced.

Encoding SUPGs with CPPNs: In the second generative encoding scheme evaluated, the CPPN encodes the attributes of a SUPG. The SUPG is a macro-neuron (Fig. 3) that upon being triggered, produces a single cycle of a CPPN encoded activation pattern. Consequently, the repeated triggering of the SUPG results in temporal oscillations, that can be used to drive the motors of a legged robot (Morse et al., 2013).

In this encoding, the CPPN is input the position (x, y) of the SUPG in the substrate, and the elapsed time (in interval $[0, 1]$) since the SUPG was last triggered (Fig. 3a). During the period of the SUPG, its internal timer ramps upwards with each simulation time-step, from an initial value of 0 to a maximum value of 1 (Fig. 3b). Therefore, the resultant activation pattern output by the SUPG is a function of both, its position in the substrate, and the time since the last cycle was initiated.

Applying the SUPGs for hexapod locomotion, the substrate comprises 12 SUPGs positioned to reflect the robot morphology (Fig. 3c). The outputs of the SUPGs at each time-step specify the desired angles for the first and second servos (s_1 and s_2), on each leg of the robot. The two SUPGs on each leg of the robot are triggered by the corresponding foot touching the ground, resulting in a closed-loop control. Finally, to avoid all the SUPGs being triggered simultaneously, the first trigger to each SUPG is delayed by an offset. In this study, the offset output of the CPPN is determined for the s_1 SUPG on each leg by supplying its coordinates as input, and setting the time input to 0. The same offset value

is also applied to the s_2 SUPG on the leg, allowing both the oscillators on each leg to start simultaneously.

Direct encoding: Locomotion controllers evolved with direct encoding are designed to be simple, wherein the actuation along each DOF of the robot is governed by the periodic signal of an uncoupled amplitude controlled phase oscillator. For an oscillator controlling servo s_j on leg l of the robot, we have:

$$\ddot{\alpha}_{ij} = \beta \left(\frac{\beta}{4} (A_{lj} - \alpha_{lj}) - \alpha_{lj} \right) \quad (3a)$$

$$\gamma_{lj} = \alpha_{lj} \cos(2\pi(t + \phi_{lj})) \quad (3b)$$

where α_{lj} and ϕ_{lj} denote the amplitude and phase of the oscillator, A_{lj} represents its intrinsic amplitude, and γ_{lj} represents the control signal with frequency of 1 Hz. The uncoupled oscillator amplitude α_{lj} converges to the intrinsic amplitude A_{lj} at rate determined by positive constant β (set to 10 for rapid convergence). The Euler integration method with a step-size of 20 ms is used to solve the differential equation.

Hexapod leg actuation is governed by the differential equation model (eq. 3) instead of a standard periodic function (e.g., Koos et al. (2013)), to allow for smooth transitions in amplitude and frequency of the control signal between different gaits and to investigate the effects of coupling between oscillators, in our future work.

There are four evolved parameters ($A_{l1}, \phi_{l1}, A_{l2}, \phi_{l2}$) on each leg $l \in \{0 \dots 5\}$ of the robot. Consequently, a directly encoded controller for the hexapod robot is fully represented by 24 parameters.

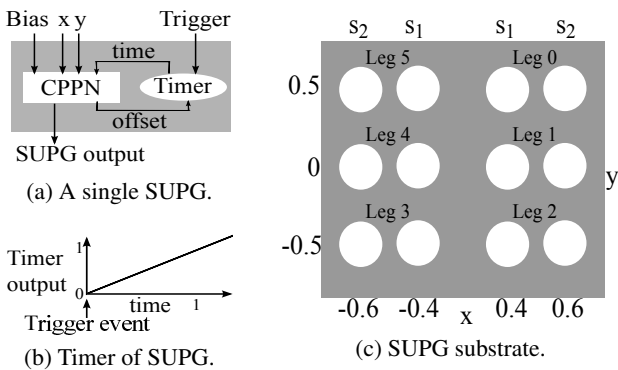


Figure 3: Encoding SUPGs with CPPNs (inspired by Morse et al. (2013)). (a) The SUPG output is a function of its coordinates (x, y) in the substrate, and the elapsed time since last trigger (output of Timer). The time of first trigger is determined by an offset. (b) Once triggered, the SUPG timer ramps upward linearly from 0 to 1 and stays there, until it is re-triggered. (c) Positions of the 12 SUPGs in the substrate, outputting the desired angles for the first two servos (s_1 and s_2), on each leg of the hexapod.

Experiments

We conducted 8000 generations of artificial selection in populations consisting of 100 individuals. Our aim was to evolve controllers for the hexapod robot to walk forward, evaluated for a period of 5 s (334 time-steps). The nondominated sorting genetic algorithm II (NSGA-II) was used to simultaneously optimize the following three objectives:

$$\text{Maximize } \begin{cases} -F_i \\ -|\Theta_i| \\ \frac{1}{N} \sum_{j=0}^{N-1} D(B_i, B_j) \end{cases} \quad (4)$$

where for individual i in the population, F_i is the fitness computed as the distance between the final position of i and a goal located 25 m directly in front of the robot's initial position, Θ_i denotes the angle of the robot's trajectory with respect to the normal forward walking direction, $D(B_i, B_j)$ is the hamming distance between the binary gait vectors of individual i and j , and N is the size of the population.

In eq. 4, the first and second objectives reward individuals to walk forward large distances towards a goal, unattainable by the robot within the experiment evaluation time. The third objective is introduced to facilitate the exploration of diverse solutions and avoid premature convergence (Mouret and Doncieux, 2012).

Artificial selection was conducted in 20 independent replicates, for each of the three encodings. Performance and evolvability analysis are reported for the best individual of each replicate, selected to have the highest fitness in the population, and with an angle of trajectory in the range

of $\pm 1^\circ$ (simulation source code can be downloaded from http://pages.isir.upmc.fr/evorob_db.)

Individual performance

In all the encodings, the performance of the best individuals rapidly increased with a quasi-stable equilibrium being reached with less than 5000 generations of selection (Fig. 4a). Additionally, the directly encoded individuals converged more rapidly as compared to those encoded with ANN and SUPG schemes (Fig. 6a). After 8000 generations, the performance of the Direct, ANN and SUPG encodings, was 1.94 ± 0.18 , 2.93 ± 1.60 and 2.78 ± 1.43 meters, respectively (Median \pm IQR, see Fig. 4b). Amongst the three encoding schemes, the directly encoded individuals received the lowest performance (Mann-Whitney test, $d.f. = 38$, all $p < 0.001$), while no significant difference in performance was detected between the two generative encoding schemes.

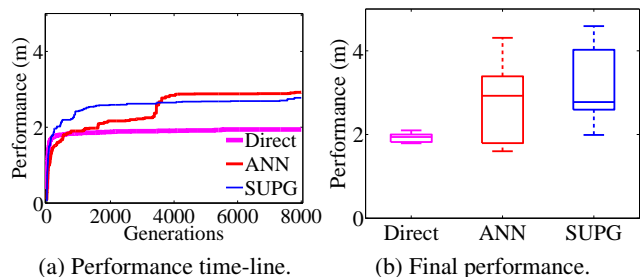


Figure 4: Performance in forward displacement for the Direct, ANN, and SUPG encodings: (a) Median performance for the 8000 generations of evolution; and (b) Performance of encodings at generation 8000.

Importantly, the frequency of the directly encoded oscillations governing leg actuation was prefixed at 1 Hz (eq. 3b). By contrast, the individuals evolved with generative encodings were capable of expressing higher frequency oscillations (2.56 ± 1.25 Hz for ANN, and 3.81 ± 0.73 Hz for SUPG), and the frequency of the gait may itself be under selection. Consequently, an assessment of the three encodings solely on the basis of the performance is biased, and other measures are needed to compare encodings.

Evolvability analysis

The evolvability provided by the encoding schemes is analyzed by mutating the best individual of each replicate at generation 8000, and reporting the following: (i) The proportion decrease in performance consequent to the mutation (eq. 1); and (ii) The gait diversity, computed as the mutual information between gait vectors of the original and mutated individual (eq. 2c). The individual is mutated 1000 times, in separate, independent instances. Finally, a kernel density estimation (Scott, 2009), is used to visualize the re-

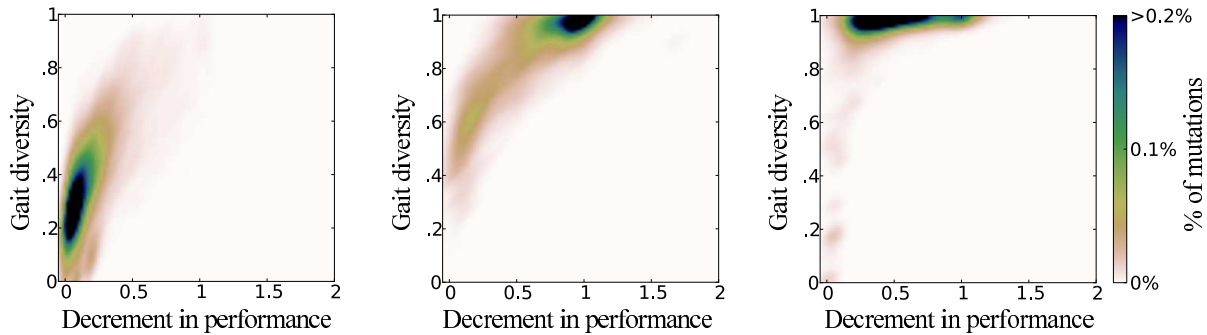


Figure 5: Gait diversity and the proportion decrease in performance, consequent to each of 20000 independent mutations, for the Direct (left), ANN (middle), and SUPG (right) encoding schemes, pooled from all 20 replicates.

sultant landscape of 20000 data points (1000 mutations \times 20 replicates), pooled together from all replicates.³

Across the three encoding schemes, the SUPG approach revealed the highest evolvability, with the capability to explore very different but viable gaits (see Fig. 5). In evolvability analysis with Direct encoding, a conservative exploration of the phenotype, limited to solutions close to the unmutated individuals was found (0.33 and 10.5%, median gait diversity, and drop in performance, respectively). By contrast, the generative encodings were aggressive in the exploration of the phenotypic landscape, with the gait diversity of mutated individuals at 0.95 for ANN, and 0.99 for the SUPG encodings. However, differences existed in the severity of negative effects of mutations amongst the two encoding schemes. The ANN encoded individuals were sensitive to the effects of deleterious and lethal mutations⁴, resulting in a 78.9% drop in performance. In comparison, individuals evolved with the SUPG encoding were much more resilient to the negative effects of mutations, with a smaller decrement of 43.1% in performance following mutation.

Adaptation to faults

The significance of our measure of evolvability of the Direct, ANN and SUPG encodings was investigated by analyzing the adaptation of the evolved robot's gait, following the removal of one of its legs. We expect that for the encodings registering a higher measure of evolvability, the corresponding evolved individuals would require fewer generations to recover an effective walking gait.

In these experiments, the new populations comprised 100 mutated individuals of the best individual of each replicate at generation 8000. We conducted a further 10000 generations of artificial selection on the populations of amputee

³Bivariate density estimation, with Gaussian type kernels over a grid of 100×100 equidistant points.

⁴Lethal mutations result in performance drops in excess of 100%, corresponding to the failure of any forward movement by the robot.

hexapods (leg 1 removed). The number of generations required to regain an effective gait and the proportion of the original performance recovered, was analyzed.

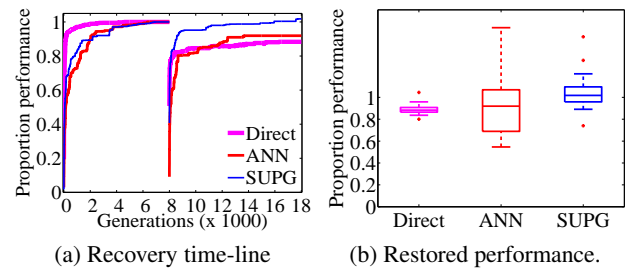


Figure 6: Recovered performance in forward displacement with the removal of leg 1 (legs numbered in Fig. 1a), for the Direct, ANN, and SUPG encodings: (a) Median proportion of performance before (left), and after (right) leg removal; and (b) Proportion of performance recovered 10000 generations after leg removal.

In the 10000 generations of selection, all the three encodings were capable of recovering the majority of their original performance (see Fig. 6a). After 10000 generations post leg removal, the recovered proportion of performance of the Direct, ANN and SUPG encodings, was 0.88 ± 0.04 , 0.92 ± 0.38 , and 1.02 ± 0.14 , respectively (Median \pm IQR, see Fig. 6b). Furthermore, amongst the three encoding schemes, the SUPG encoded individuals recovered the most of their original performance (Mann-Whitney test, *d.f.* = 38, all $p < 0.001$), while no significant difference in recovered performance was detected between the Direct and ANN schemes.

In order to analyze the recovery rate of amputee hexapods, in Fig. 7 we have plotted the performance lost immediately after leg removal in each of 20 replicates (horizontal axis) and the number of generations required to recover 90% of the original performance in each replicate (vertical

axis). The directly encoded individuals could make a 90% recovery in only 6 of 20 replicates, and suffered a median drop in performance of 60.9% immediately following leg removal. By contrast, the ANN and SUPG generative encoding schemes were better able to recover the loss in performance following amputation. In experiments with ANN encoding, 12 of 20 replicates recovered 90% of their original performance in 4392 generations (median for all replicates), despite a drastic initial loss in performance of 93.9%. The SUPG encoded amputee hexapods exhibited the fastest recovery, with 18 of 20 replicates being able to make the mark in 559 generations (median for all replicates), after incurring a smaller initial drop in performance of 67.3% consequent to leg removal.

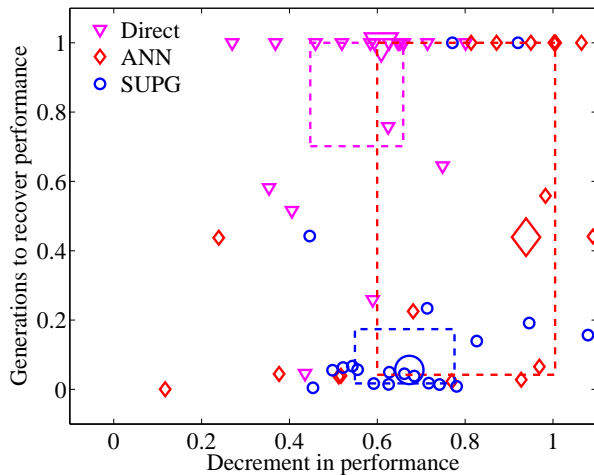


Figure 7: Proportion of performance lost immediately after leg removal and the number of generations (of 10000) required for recovery, in each of the 20 replicates (small markers), and the corresponding medians (large markers), for the Direct, ANN, and SUPG encoding schemes. The bounding boxes extend from first to third quartile.

Discussion

We investigated the evolvability provided by generatively encoded ANNs (Clune et al., 2011; Yosinski et al., 2011) and SUPGs (Morse et al., 2013), and a direct encoding, quantifying both, (i) the robustness to deleterious mutations, and (ii) the variations in the phenotypes after genetic change, for the hexapod locomotion problem. The significance of our measure of evolvability was evaluated by the individual adaptation response to morphological changes, not previously encountered by the hexapod.

Our results revealed a direct relationship between the estimated evolvability, and the capability of an individual to adapt to severe changes in its morphology, simulated by the amputation of one of its legs. Amongst the three encodings evaluated, SUPGs exhibited the highest evolvability,

and were best able to recovery following leg removal. In all but two replicates, the individuals were capable of recovering 90% of their original performance, and did so almost an order of magnitude faster than the other encodings, perhaps consequent to the closed-loop control ingrained in the SUPG encoding scheme. Such a mechanism would attempt to reform deleteriously mutated gaits, and consequently buffer the individual from the negative effects of such mutations. However, the effects of such a self-correction mechanism on evolvability needs further investigation.

The generatively encoded ANN and SUPG individuals achieved high performance gaits, when evolved for the symmetric hexapod robot. However, the performance recovery following leg removal was appreciably different between the two encoding schemes. The longer recovery of ANN individuals may be consequent to the resultant unsymmetrical and irregular hexapod morphology, a region of the problem geometry space wherein ANN generative encodings have been shown to perform poorly (Clune et al., 2011). Consequently, the damage recovery of ANN-encoded individuals would improve in situations wherein changes to the hexapod morphology preserve its symmetry (such as the removal of two middle legs). The outcome of such a scenario needs to be explored further.

For our measure of evolvability, we mutated the individuals with a predetermined mutation rate, tuned to allow a speedy convergence of the evolved solutions. This is a critical consideration, as variations to the mutation rate can affect the viability and gait diversity of generated mutants. Preliminary results suggest that with an increase in mutation rate, the peak of the distribution of mutants shifts towards more diverse gaits. However, the overall shape of the distribution, highlighting desirable regions of the evolvability landscape, remains the same for all three encodings. A rigorous analysis on the effect of variations in mutation rate and size (e.g, small mutations to many genes, or large mutations to a few genes) on the evolvability landscape, is part of our future work.

In this study, we present a novel approach to visualize the evolvability of a genotype-phenotype map, to analyze simultaneously the quality of genetic mutations, and the quantity of phenotypic variation generated. Our measure of evolvability may allow the selection of an encoding capable of producing adaptable individuals, when a inter-encoding performance comparison is no longer sufficient. In our model, the phenotypic variation was associated with the mutual information between hexapod gaits, but the diversity may be similarly computed for bipedal and quadrupedal robots, and for other qualities, such as the final position of a robot in a maze navigation task (Lehman and Stanley, 2011; Mouret and Doncieux, 2012). In this way, we hope our work may serve as a stepping stone for further research on evolvability, applied to a wide range of problems in the field of evolutionary computation.

Supplemental data: Movies of hexapod walking behavior are online at <http://goo.gl/4HC5MY>.

Acknowledgment: This study was supported by the ANR Creadapt project (ANR-12-JS03-0009).

References

- Cheney, N., MacCurdy, R., Clune, J., and Lipson, H. (2013). Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. In *Proc. of GECCO*, pages 167–174. ACM, New York, NY.
- Clune, J., Mouret, J.-B., and Lipson, H. (2013). The evolutionary origins of modularity. *Proceedings of the Royal Society b: Biological sciences*, 280(1755):20122863.
- Clune, J., Stanley, K. O., Pennock, R. T., and Ofria, C. (2011). On the performance of indirect encoding across the continuum of regularity. *IEEE Trans. Evol. Comput.*, 15(3):346–367.
- Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. John Wiley & Sons, Inc.
- Gauci, J. and Stanley, K. O. (2010). Autonomous evolution of topographic regularities in artificial neural networks. *Neural Comput.*, 22(7):1860–1898.
- Gerhart, J. and Kirschner, M. (2007). The theory of facilitated variation. *Proc. Natl. Acad. Sci. U.S.A.*, 104(Suppl 1):8582–8589.
- Hornby, G. S. (2005). Measuring, enabling and comparing modularity, regularity and hierarchy in evolutionary design. In *Proc. of GECCO*, pages 1729–1736. ACM, New York, NY.
- Hornby, G. S., Lipson, H., and Pollack, J. B. (2003). Generative representations for the automated design of modular physical robots. *IEEE Trans. Robot. Automat.*, 19(4):703–719.
- Hornby, G. S. and Pollack, J. B. (2002). Creating high-level components with a generative representation for body-brain evolution. *Artif. Life*, 8(3):223–246.
- Hu, T. and Banzhaf, W. (2010). Evolvability and speed of evolutionary algorithms in light of recent developments in biology. *J. Artif. Evol. App.*, 2010:1:1–1:28.
- Kajita, S. and Espiau, B. (2008). Springer handbook of robotics. In Siciliano, B. and Khatib, O., editors, *Springer handbook of robotics*, chapter 16. Springer Berlin Heidelberg.
- Kirschner, M. and Gerhart, J. (1998). Evolvability. *Proc. Natl. Acad. Sci. U.S.A.*, 95(15):8420–8427.
- Komosiński, M. and Rotaru-Varga, A. (2001). Comparison of different genotype encodings for simulated three-dimensional agents. *Artif. Life*, 7(4):395–418.
- Koos, S., Cully, A., and Mouret, J.-B. (2013). Fast damage recovery in robotics with the T-resilience algorithm. *Int. J. Robot. Res.*, 32(14):1700–1723.
- Lehman, J. and Stanley, K. O. (2011). Improving evolvability through novelty search and self-adaptation. In *Proc. of CEC*, pages 2693–2700. IEEE Press, Piscataway, NJ.
- Lehman, J. and Stanley, K. O. (2013). Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PLoS one*, 8(4):e62186.
- Liu, H. and Iba, H. (2004). A hierarchical approach for adaptive humanoid robot control. In *Proc. of CEC*, volume 2, pages 1546–1553 Vol.2. IEEE Press, Piscataway, NJ.
- Morse, G., Risi, S., Snyder, C. R., and Stanley, K. O. (2013). Single-unit pattern generators for quadruped locomotion. In *Proc. of GECCO*, pages 719–726. ACM, New York, NY.
- Mouret, J.-B. and Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evol. Comput.*, 20(1):91–133.
- Pavlicev, M. and Wagner, G. P. (2012). Coming to grips with evolvability. *Evol. Educ. Outreach*, 5(2):231–244.
- Reisinger, J. and Miikkulainen, R. (2007). Acquiring evolvability through adaptive representations. In *Proc. of GECCO*, pages 1045–1052. ACM, New York, NY.
- Reisinger, J., Stanley, K. O., and Miikkulainen, R. (2005). Towards an empirical measure of evolvability. In *Proc. of GECCO*, pages 257–264. ACM, New York, NY.
- Scott, D. W. (2009). *Multivariate density estimation: theory, practice, and visualization*, volume 383. John Wiley & Sons, Inc.
- Seys, C. W. and Beer, R. D. (2007). Genotype reuse more important than genotype size in evolvability of embodied neural networks. In *Proc. of ECAL*, pages 915–924. Springer Berlin Heidelberg.
- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162.
- Stanley, K. O., D’Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artif. Life*, 15(2):185–212.
- Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artif. Life*, 9(2):93–130.
- Tonelli, P. and Mouret, J.-B. (2013). On the relationships between generative encodings, regularity, and learning abilities when evolving plastic artificial neural networks. *PLoS one*, 8(11):e79138.
- Valsalam, V. K. and Miikkulainen, R. (2008). Modular neuroevolution for multilegged locomotion. In *Proc. of GECCO*, pages 265–272. ACM, New York, NY.
- Wagner, G. P. and Altenberg, L. (1996). Perspective: Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976.
- Yosinski, J., Clune, J., Hidalgo, D., Nguyen, S., Zagal, J., and Lipson, H. (2011). Evolving robot gaits in hardware: the hyperneat generative encoding vs. parameter optimization. In *Proc. of ECAL*, pages 890–897. MIT Press, Cambridge.