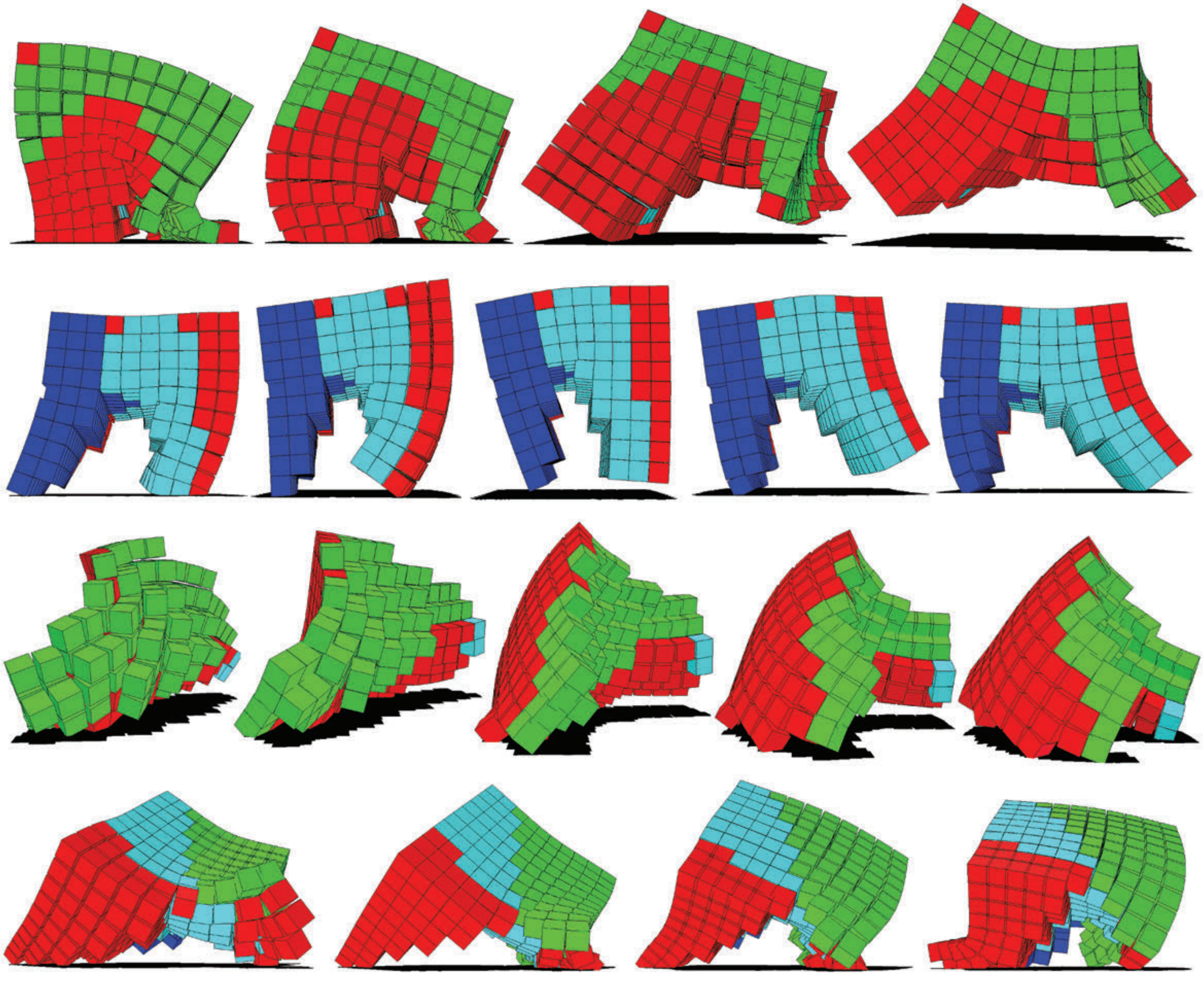
SIGEVOlution

newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation

Volume 7
Issue 1



in this issue

Visualising Evolutionary Search Spaces

James McDermott

Unshackling Evolution

Nick Cheney, Robert MacCurdy

Jeff Clune, and Hod Lipson

The BEACON Center

new theses

calls & calendar

Editorial

GECCO-2014 was great as usual and Vancouver was an amazing venue. Great hotel, excellent nearby restaurants. And what about the weather huh? Sunny, clear sky, hot but not too hot. In other words, perfect. Dirk Arnold, Christian Igel, and their team did an amazing job. They surely raised the bar for the next GECCO organizers!

Before leaving for vacation, I am very happy to bring you the first issue of the new volume. As I anticipated during the SIGEVO meeting, the newsletter is four issues behind but we will try to catch up by April 2015 which would put us right on schedule. But to succeed we need your help. So, if you have EC applications that you wish to showcase to a broader audience, if you just published new Master or PhD theses, if you want to let our community know how your workshop or your conference turned out, if you have new EC software available, if you have a lab (like BEACON) that you wish to advertise, just drop me an email.

In this issue we have a new article by James McDermott about an approach to visualize evolution and an article by the winners of the Virtual Creatures Competition, Nick Cheney, Robert MacCurdy, Jeff Clune and Hod Lipson, who — as you may guess — provided the images for the nice cover. The issue continues with the presentation of the BEACON Center for the Study of Evolution in Action and William Raffe's new PhD thesis. At the end, the usual rich list of forthcoming events.

As always, due thanks to the people who made this possible: James McDermott, Nick Cheney, Robert MacCurdy, Jeff Clune, Hod Lipson, William L. Raffe, Daniele Loiacono, Cristiana Bolchini, Viola Schiaffonati, and Francesco Amigoni.

Have a great holiday. See you in September!

Pier Luca
August 4, 2014

SIGEVOlution Volume 7, Issue 1

Newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation.

SIGEVO Officers

Wolfgang Banzhaf, Chair
Una-May O'Reilly, Vice Chair
Marc Schoenauer, Secretary
Franz Rothlauf, Treasurer

SIGEVOlution Editor in Chief

Pier Luca Lanzi

Contributors to this Issue

James McDermott
Nick Cheney
Robert MacCurdy
Jeff Clune
Hod Lipson
Erik Goodman
William Raffe

Contents

Visualizing Evolution **2**
  James McDermott
Unshackling Evolution **11**
  Nick Cheney
  Robert MacCurdy
  Jeff Clune
  Hod Lipson
The BEACON Center **25**
New Theses **27**
Calls and Calendar **20**
About the Newsletter **24**

Visualising Evolutionary Search Spaces

James McDermott (jmmcd@jmmcd.net)
Management Information Systems, Lochlann Quinn School of Business — Natural Computing Research and Applications Group, Complex and Adaptive Systems Lab, University College Dublin, Ireland

Understanding the structure of search spaces can help us to design better search algorithms, and it is natural to try to understand search spaces by visualising them. For typical evolutionary search spaces, like the space of genetic programming trees, visualising them directly is impossible, because of their large dimensionality. However, we can use the idea of *distances* on search spaces to project them into two dimensions, expose their structure, and obtain useful and attractive visualisations.

1 Introduction

We are physical beings, and our thinking is deeply intertwined with our physicality. Perhaps the best examples come from our speech: in English, we say that we put ourselves in someone else's shoes; we move forward, when in fact time is just passing; we climb the ladder of promotion. More specifically, our visual sense is so central to our thinking that it is hard to carry on a conversation without appealing to visual metaphors. We see the other person's point of view; we look into the past; we see things in our minds' eye; we see the way out of a problem.

When we use visualisations to help us understand data – whether on paper, on a computer screen, or even using some 3D technology – we are "using vision to think" [3]. Visualisations have become central in making abstract and complex situations more concrete and understandable.

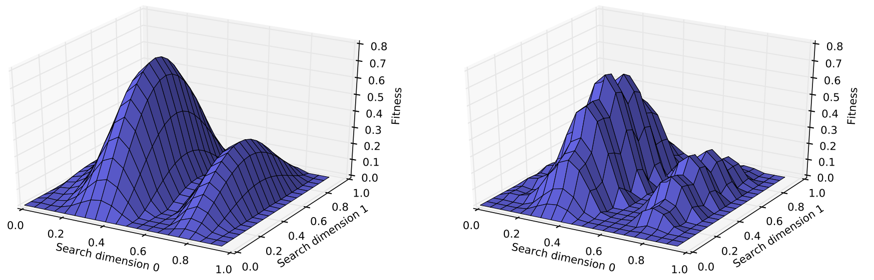


Fig. 1: Fitness landscapes representing search in two dimensions, with fitness as height.

Few everyday problems present as much abstraction and complexity as those which occur in our field of evolutionary computation (EC). In particular, the physical metaphor of the *fitness landscape* is central when we discuss how EC works. The fitness landscape is often pictured as something like a mountain range, with two dimensions representing the search space and its connectivity, and height representing fitness. As a metaphor, the fitness landscape allows us to imagine search from the inside: we place ourselves *in the search space*, and look around, and imagine the properties of the landscape which might lead to difficulties for search (see Figure 1).

(a) Pseudo-3D view

(b) “Flattened” view with graph layout algorithm

Fig. 2: The four-dimensional hypercube representing bitstrings of length 4 with bitflip mutation.

As a metaphor, the fitness landscape is invaluable. However, as a data visualisation, it is rather limited: it works well only when the search space is a subset of $\mathbb{R}^2$. Realistic search spaces are of higher dimensionality, or are not real-valued (e.g. spaces of bitstrings or permutations), or don't have an obvious number of dimensions (e.g. spaces of trees). Visualising such spaces is not easy.

Even when omitting fitness information, and visualising just the connectivity of the space, the results can be unsatisfactory. A search space and a mutation operator together constitute a directed graph, which has a natural visualisation. Individuals or points in the space are nodes; there is a directed edge from one node to another if mutation from one to the other is possible. However, for larger spaces, drawing such a graph tends to lead to complicated, tangled pictures. Even with bitstrings of length 4, the visualisation becomes complicated. Graph layout algorithms can be used to disentangle the edges as much as possible, but the results can still be hard to interpret (see Figure 2).

We approach the problem of visualising EC spaces by defining the *distance* between pairs of individuals. Depending on the EC representation, distance can reflect genotypic or phenotypic dissimilarity, or it can be based on mutation transition probabilities or related quantities, as described in Section 2. We then use *multi-dimensional scaling* (MDS), described in Section 3, to produce a two-dimensional layout for the individuals in a space. This can be plotted directly, omitting edges, in contrast to graph layout algorithms. The results, shown in Section 4, allow us to visualise the distribution of individuals and obtain hints as to the structure of several (very small) search spaces: bitstrings, permutations, and especially program trees. In Section 5 we consider some related work, and Section 6 concludes.

2 Defining distances on search spaces

A distance is a real-valued function of two arguments which abstracts the notion of physical distance. For a pair of objects, a distance function gives a numerical value reflecting the dissimilarity, dissociation, or physical distance *per se* between them: small values mean they are similar, associated, or nearby; larger values mean the opposite. A distance which obeys certain axioms is called a *metric*, but the axioms are not essential to our task.

In the context of EC, distances between individuals arise in several ways, depending on the representation. Genotypic, phenotypic, and mutation operator properties can be used to define distances.

We will consider three (very small) search spaces in different representations. For each, we define one or more distances between pairs of individuals. Since we will also visualise fitness, we also define simple fitness functions for each.

For **bitstrings**, as used in genetic algorithms, the Hamming distance is well-known. We consider the 1024 bitstrings of length 10. Fitness is on the one-max problem.

For **permutations**, as used in travelling salesman problems (TSP) and similar, a suitable distance is the Kendall τ distance [14, 24]. We consider the 360 distinct permutations of length 7. Fitness is on a random 7-city TSP instance.

We will focus particularly on **program trees**, as used in genetic programming. We consider the trees composed of the four functions $+$, $-$, $*$, $/$ (with $/$ functioning as the analytic quotient of [17] to avoid division by zero) and the two variables x and y, with a maximum depth of two, leading to a maximum number of nodes of 7. The number of trees in this space is 1298. We will use subtree mutation as the mutation operator, and a two-variable symbolic regression problem from [20] as the fitness function. We can define multiple genotypic (syntactic) and phenotypic (semantic) distances. We will use four syntactic distances:

TED Tree-edit distance [19] counts the number of atomic edits (node additions, deletions, or re-labellings) which must be carried out to transform one tree into another.

TAD Tree-alignment distance [30] aligns trees, allowing for empty nodes in the alignment, and counts the number of locations in which they differ. We use a variation in which differences at deeper locations are de-weighted.

OVD Overlap distance [25] calculates all subtrees in both trees and the number of subtrees they have in common.

FVD A distance based on the Euclidean distance of the trees in a simple feature space where the features are number of nodes, minimum and maximum depth, fan-out, and symmetry.

We will also define semantic distance on real-valued program trees:

SEMD Semantic distance is based on the Euclidean distance between the two trees' vectors of outputs, when they are executed on the vector of fitness cases. We take the log of the Euclidean distance, since Euclidean distances between GP individuals tend to include some very large outliers.

For all EC representations, we can also define distances based on the action of genetic operators – in the simplest case, mutation. We will use two mutation-based distances:

$\mathrm{SD}_{\mathrm{TP}}$ The symmetric transition probability distance $\mathrm{SD}_{\mathrm{TP}}(u,v)$ is based on the transition probability $\mathrm{TP}(u,v)$, i.e. the probability that u will transition to v in a single step of the mutation operator. We take the negative log of this, so that larger values mean more inaccessible: $\mathrm{D}_{\mathrm{TP}}(u,v) = -\log(\mathrm{TP}(u,v))$. We define $\mathrm{D}_{\mathrm{TP}}(u,u) = 0$. Finally we symmetrize by taking $\mathrm{SD}_{\mathrm{TP}}\ (u,v) = (\mathrm{D}_{\mathrm{TP}}(u,v) + \mathrm{D}_{\mathrm{TP}}(v,u))/2$. Symmetry is necessary for this application since the visualisation technique we use fails with asymmetric distances (see Section 3, next).

CT The commute time is the expected number of steps required to transition from one individual to another and back. It is naturally symmetric. It can be derived using Markov chain methods from the transition probabilities.

	p_0	p_1	p_2	p_3
p_0	0	1	$\sqrt{2}$	1
p_1	1	0	1	$\sqrt{2}$
p_2	$\sqrt{2}$	1	0	1
p_3	1	$\sqrt{2}$	1	0

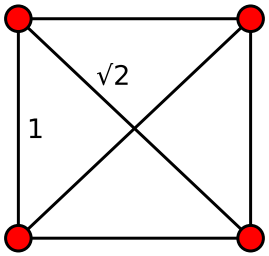


Fig. 3: The matrix of distances between points (left) are enough for MDS to determine their layout in 2D (right).

3 Multi-dimensional scaling

The technique we will use to project the search space into two dimensions is *multi-dimensional scaling*, or MDS [15, 4]. It is commonly used as a dimensionality-reduction technique: given p points distributed in $\mathbb{R}^n$, the goal is to find p points in $\mathbb{R}^m$ $(m < n)$ whose inter-point distances approximate the original inter-point distances. It works by scattering points randomly in $\mathbb{R}^m$, then adjusting them iteratively in order to approximate the original distances. The algorithm is guaranteed to produce a result, though there is no guarantee that the distances between the points produced will closely approximate the original distances.

In fact MDS is somewhat more general: the input points need not be given explicitly, but instead just their inter-point distances need be given: see Figure 3. They need not correspond to Euclidean distances: they might arise, for example, as entries of a confusion matrix, or (inverted) correlation coefficients [15].

They can also arise in several natural ways between pairs of individuals in EC search spaces, as explained in Section 2. We can therefore calculate inter-point distances for all pairs of individuals and use MDS with $m = 2$ to produce a two-dimensional visualisation of the space.

MDS can be seen as a generalisation of the graph layout algorithm used in our Figure 2b. That algorithm is due to [13] and is available as *neato*[1]. It uses a *spring model*. It works by trying to push the nodes to positions such that the Euclidean distance in the plane between each pair of nodes approximates the shortest path distance between them in the graph. It then adds edges as appropriate. MDS is a generalisation of this because it can use any distance matrix, rather than just the shortest path distance arising in a graph.

4 Visualisations

We use MDS to obtain a layout of the individuals and add fitness information as colour. We will use different colour maps in the different images purely for aesthetic effect, but regardless of the colour map, darker colours indicate better fitness. Results for the three search spaces are shown in Figure 4.

Both the bitstring and the permutation spaces are highly symmetric, as expected, and the tree space less so. The permutation and tree images correctly show that the fitness landscape is rather rugged, with good fitness values scattered among bad ones. The bitstring image shows a smoother landscape. However it fails to show a smooth upward gradient from every individual towards the optimum. Such a gradient does exist in the search space for the one-max problem because every individual other than the optimum can be improved by a single bit-flip mutation. Although it would be possible to lay the individuals out to show this gradient, it would disimprove the layout with respect to individuals other than the optimum.

In the tree space we also use larger markers to represent larger trees (measured as number of nodes). The tree space under TED is shown as consisting of clumps of trees, arranged with some regularity. The two one-node trees x and y are marked with very small markers, outliers from the main patterns. The larger trees (of which there are many) tend to clump together, but in multiple clumps. The four-fold structure of the space is visible, caused by the four non-terminal choices for the root node of the tree.

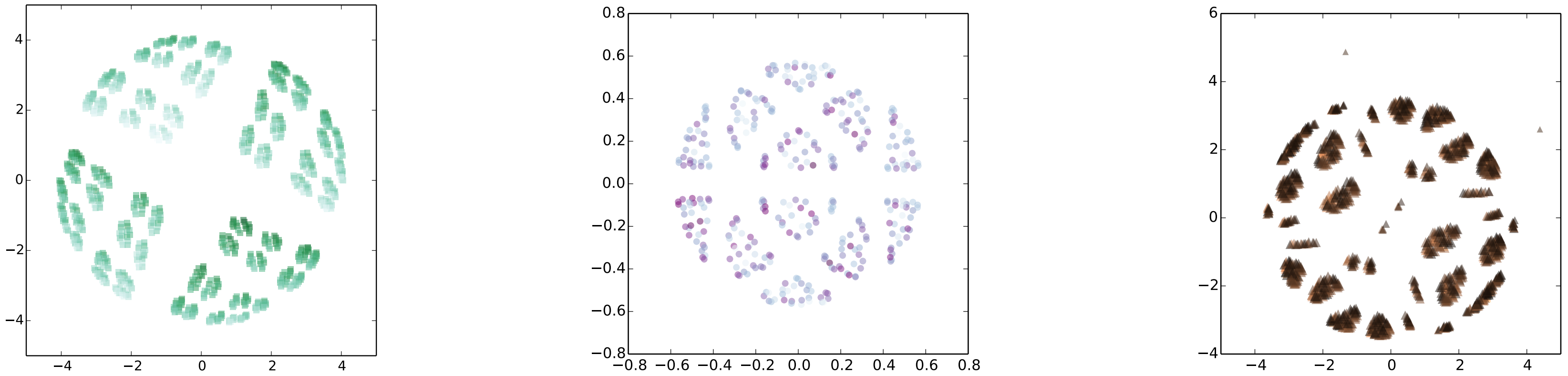
Next, we will compare the results of several different distances on the tree space. In Figure 5, we find contrasting views. TAD and OVD give similar overall behaviour to TED. OVD, however, has a two-fold rather than four-fold structure. FVD, in contrast, produces a highly asymmetric space. TAD puts the two one-node trees close together (the small markers in the centre-left of Figure 5a), since just a single node relabelling is needed to transform between them. In contrast, OVD puts them as far away from each other as possible (bottom-left and top-right of Figure 5b), since they share no subtrees. They are outliers according to FVD (top-right of Figure 5c) since they differ in several features from all other nodes. All three (TAD, OVD, and FVD) continue to exhibit some asymmetry.

In Figure 6, two mutation-based distances are shown, and give very similar results. $\mathrm{SD}_{\mathrm{TP}}$ produces a core of trees surrounded by an “asteroid field”; CT produces a tighter version of the same thing. We can see these images as visualisations of the *fitness distance correlation* (FDC) statistic [12]. In a space with perfect FDC, we would find colours changing uniformly with distance from the optimum. These images therefore illustrate the fact that GP symbolic regression does not give a perfect FDC.

The semantic distance used in Figure 7 is interesting because when individuals are laid out according to semantic distance, the space exhibits a smooth gradient leading towards the optimum (darkest points). Here, FDC is indeed perfect. Both images use SEMD, hence the layout of points is the same. However, for the second image we have introduced a different target function $1+x^2$ (still using two variables and the same fitness cases as before), so the smooth gradient leads to a different area of the space.

---

[1] Graphviz, http://www.graphviz.org



(a) Bitstrings (Hamming)

(b) Permutations (Kendall τ)

(c) Trees (TED)

Fig. 4: Small spaces of bitstrings, permutations, and trees. Darker colours indicate better fitness.

5 Related work

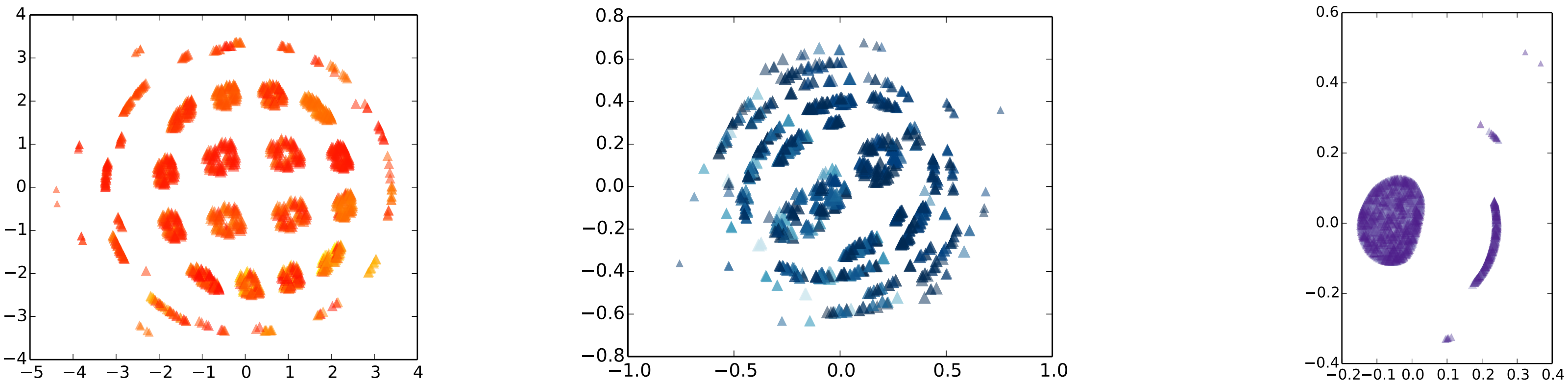
Visual analytics are common for visualising EC data. Many EC papers include graphs representing changes in solution quality over time, for example. But the study of more advanced visualisations is an important developing sub-field within EC. The VizGEC workshop has been running for five years at the time of writing[2]. Franken [8] visualised the behaviour of evolutionary algorithms under different hyperparameters. [16] used several types of visual analytics to help understand the behaviour of interactive EC. [2] and [9] visualised individuals' genetic heritage, while [18] did the same for island-model algorithms.

Several authors have used MDS and related ideas to produce low-dimensional visualisations of EC populations and behaviour. [10] used self-organising maps to visualise in two dimensions the current population of an interactive EC run. This helps the user to interact with the population. The self-organising map is a type of neural network which can be used as an alternative to MDS, though the results have somewhat different properties [6]. Stan-Bishop et al. [26] used a custom force-directed layout algorithm to visualise in two dimensions the current population of a non-interactive EC run. Collins & O'Neill [5] used principal components analysis, a linear analogue of MDS, to create visualisations of GA search behaviour, averaged over multiple runs. Pohlheim [23] used *Sammon mapping*, a variant of MDS, to compare the paths through the search space taken by multiple runs of an evolutionary algorithm. Valdés et al. [28] also used Sammon mapping to project the Pareto fronts for several multi-objective problems into three dimensions. The problem of visualising Pareto fronts was also tackled in a line of research by [31]. In our work, it is the entire space, rather than the current population or Pareto front, which is visualised. Dybowski [7] used Sammon mapping to visualise the entire space of bitstrings of length 5, similar to our Figure 4a (where bitstrings are of length 10).

Outside the field of EC, [27] used the idea of tree edit distance to expose the structure of a space of trees. This was not a search space: instead, the trees were an internal representation used for image matching in a computer vision application. Principal components analysis was used to present visualisations similar to ours. Visualisations of tree spaces have been used to help understand phylogenies in evolutionary biology [1, 11]. Again, tree edit distance provides a distance between pairs of trees, and MDS provides a visualisation. Again, the main difference in our work is that we are visualising a search space.

---

[2] http://www.vizgec.ex.ac.uk/



(a) TAD

(b) OVD

(c) FVD

Fig. 5: Trees of maximum depth 2, visualised using MDS under three syntactic distances. Darker colours indicate better fitness. Marker size represents number of nodes in the tree.

6 Conclusions

We have produced two-dimensional visualisations of EC search spaces, in which Euclidean distances between pairs of individuals approximate the syntactic, semantic, or mutation-based distances between them. These allow us to compare the structure of different spaces, see differences in the behaviour of distances, understand the interplay between syntactic distance, semantic distance, and fitness, and see the differences between different fitness functions.

In future work, we will consider more modern methods of projecting the space into two dimensions, such as *t-distributed stochastic neighbor embedding* [29]. We will also seek to visualise much larger search spaces using sampling. We will also consider 3D visualisations.

Our code uses the MDS implementation of Scikit-learn [22] and the tree-edit distance implementation of [21], and is available at `https://github.com/jmmcd/GPDistance`.

References

[1] Amenta, N. and Klingner, J. (2002). Case study: Visualizing sets of evolutionary trees. In *INFOVIS 2002: IEEE symposium on information visualization*, pages 71–74. IEEE.

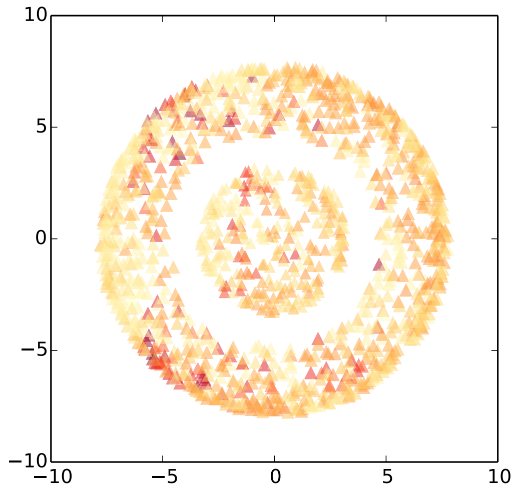
[2] Burlacu, B., Affenzeller, M., Kommenda, M., Winkler, S., and Kronberger, G. (2013). Visualization of genetic lineages and inheritance information in genetic programming. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation (conference companion)*, pages 1351–1358. ACM.

[3] Card, S. K., Mackinlay, J. D., and Shneiderman, B. (1999). *Readings in information visualization: using vision to think*. Morgan Kaufmann.

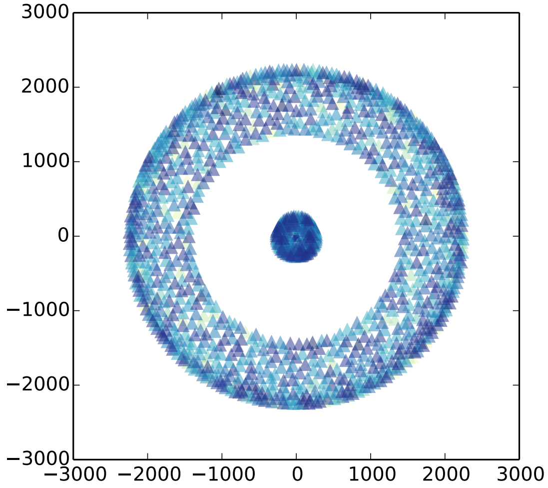
[4] Carroll, J. D. and Arabie, P. (1980). Multidimensional scaling. *Annual Review of Psychology*, 31:607–649.

[5] Collins, J. and O'Neill, M. (1999). Visualisation of evolutionary algorithms using principal components analysis. In *GECCO Workshop Program*, pages 99–100. Citeseer.

[6] Duch, W. and Naud, A. (1996). Multidimensional scaling and Kohonen's self-organizing maps. In *Proc. 2nd Conf. Neural Networks and their Applications, Szczyrk, Poland*, volume 1, pages 138–143. `http://www.fizyka.umk.pl/publications/kmk/96-sommds.pdf`.

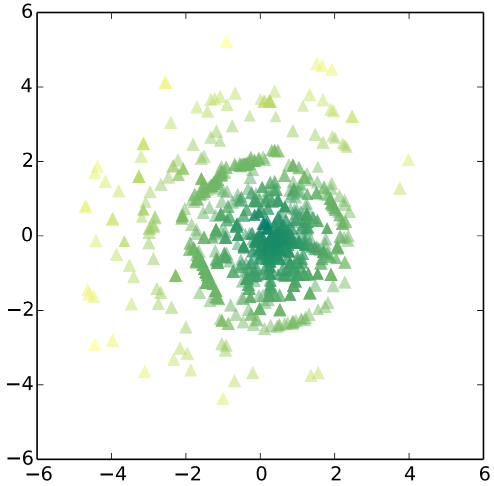


$\mathrm{SD}_{\mathrm{TP}}$

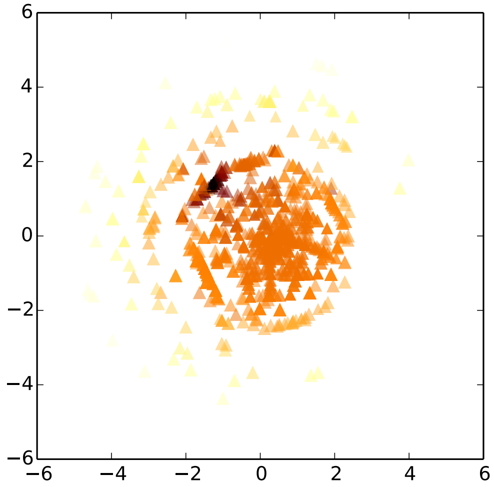


CT

Fig. 6: Trees of maximum depth 2: mutation-based distances.



SEMD with fitness on target function $1/(1+x^{-4})+1/(1+y^{-4})$ [20]



SEMD with fitness on target function $1+x^2$

Fig. 7: Trees of maximum depth 2: semantic distance. Darker colours indicate better fitness.

[7] Dybowski, R., Collins, T. D., and Weller, P. (1996). Visualization of binary string convergence by Sammon mapping. In Fogel, L., Angeline, P., and Bäck, T., editors, *The Proceedings of the Fifth Annual Conference on Evolutionary Programming (EP96)*, pages 377–383, San Diego, CA, USA. MIT Press.

[8] Franken, N. (2009). Visual exploration of algorithm parameter space. In *CEC'09: IEEE Congress on Evolutionary Computation*, pages 389–398. IEEE.

[9] Hart, E. and Ross, P. (2001). GAVEL—a new tool for genetic algorithm visualization. *Transactions on Evolutionary Computation*, 5(4):335–348.

[10] Hayashida, N. and Takagi, H. (2000). Visualized IEC: Interactive evolutionary computation with multidimensional data visualization. In *IECON 2000: 26th Annual Conference of the IEEE Industrial Electronics Society*, volume 4, pages 2738–2743. IEEE.

[11] Hillis, D. M., Heath, T. A., and John, K. S. (2005). Analysis and visualization of tree space. *Systematic Biology*, 54(3):471–482.

[12] Jones, T. and Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[13] Kamada, T. and Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15.

[14] Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

[15] Kruskal, J. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

[16] Llorà, X., Sastry, K., Alías, F., Goldberg, D. E., and Welge, M. (2006). Analyzing active interactive genetic algorithms using visual analytics. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1417–1418. ACM.

[17] Ni, J., Drieberg, R. H., and Rockett, P. I. (2013). The use of an analytic quotient operator in genetic programming. *Transactions On Evolutionary Computation*, 17(1).

[18] O'Neill, M., Brabazon, A., and Hemberg, E. (2011). Tracer spectrum: a visualisation method for distributed evolutionary computation. *Genetic Programming and Evolvable Machines*, 12(2):161–171.

[19] O'Reilly, U.-M. (1997). Using a distance metric on genetic programs to understand genetic operators. In *IEEE International Conference on Systems, Man, and Cybernetics: Computational Cybernetics and Simulation*, volume 5.

[20] Pagie, L. and Hogeweg, P. (1997). Evolutionary consequences of coevolving targets. *Evolutionary Computation*, 5:401–418.

[21] Pawlik, M. and Augsten, N. (2011). RTED: a robust algorithm for the tree edit distance. *Proceedings of the VLDB Endowment*, 5(4):334–345.

[22] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[23] Pohlheim, H. (2006). Multidimensional scaling for evolutionary algorithms—visualization of the path through search space and solution space using Sammon mapping. *Artificial life*, 12(2):203–209.

[24] Sevaux, M., Sörensen, K., et al. (2005). Permutation distance measures for memetic algorithms with population management. In *Proceedings of the 6th Metaheuristics International Conference (MIC'05)*, Vienna, Austria.

[25] Spector, L. (2011). Overlap. `http://push.i3ci.hampshire.edu/2011/06/07/overlap/`. Blog post last viewed 11 October 2013.

[26] Stan-Bishop, C., Barone, L., and While, R. L. (2009). Visualisation of building blocks in evolutionary algorithms. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 2431–2438. IEEE.

[27] Torsello, A., Robles-Kelly, A., and Hancock, E. R. (2007). Discovering shape classes using tree edit-distance and pairwise clustering. *International Journal of Computer Vision*, 72(3):259–285.

[28] Valdés, J., Barton, A., and Orchard, R. (2007). Virtual reality high dimensional objective spaces for multi-objective optimization: An improved representation. In *CEC'07: IEEE Congress on Evolutionary Computation*. IEEE.

[29] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85.

[30] Vanneschi, L. (2004). *Theory and Practice for Efficient Genetic Programming*. PhD thesis, Université de Lausanne.

[31] Walker, D. J., Everson, R. M., and Fieldsend, J. E. (2013). Visualizing mutually nondominating solution sets in many-objective optimization. *Transactions on Evolutionary Computation*, 17(2):165–184.

About the author

James McDermott is a Lecturer in Business Analytics in University College Dublin, Ireland, and a member of the University College Dublin Centre for Business Analytics and Complex and Adaptive Systems Lab. His research interests are in evolutionary computation for program synthesis, music, graphics, design, and analytics. He holds a BSc in computer science and mathematics from NUI Galway, Ireland (2000) and a PhD in evolutionary computation and computer music from the University of Limerick, Ireland (2008). His postdoctoral research was in evolutionary computation, evolutionary design, and genetic programming, in University College Dublin and Massachusetts Institute of Technology, as an IRCSET/Marie Curie Fellow. He was the co-chair of EvoMUSART 2013 and 2014 and publication chair of EuroGP 2015.

Homepage: http://www.jmmcd.net
Email: jmmcd@jmmcd.net

Unshackling Evolution

Evolving Soft Robots with Multiple Materials and a Powerful Generative Encoding

Nick Cheney (nac93@cornell.edu), Robert MacCurdy (rbm7@cornell.edu),
Jeff Clune[†] (jeffclune@uwyo.edu) & Hod Lipson (hod.lipson@cornell.edu)
Creative Machines Lab, Cornell University, Ithaca, New York
[†]Evolving AI Lab, University of Wyoming, Laramie, Wyoming

In 1994, Karl Sims' evolved virtual creatures showed the potential of evolutionary algorithms to produce natural, complex morphologies and behaviors [30]. One might assume that nearly 20 years of improvements in computational speed and evolutionary algorithms would produce far more impressive organisms, yet the creatures evolved in the field of artificial life today are not obviously more complex, natural, or intelligent. Fig. 2 demonstrates an example of similar complexity in robots evolved 17 years apart.

One hypothesis for why there has not been a clear increase in evolved complexity is that most studies follow Sims in evolving morphologies with a limited set of rigid elements [21, 4, 3, 16, 22]. Nature, in contrast, composes organisms with a vast array of different materials, from soft tissue to hard bone, and uses these materials to create sub-components of arbitrary shapes. The ability to construct morphologies with heterogeneous materials enables nature to produce more complex, agile, high-performing bodies [35].

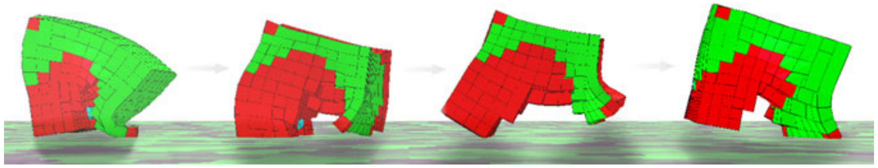


Fig. 1: An example of a natural looking morphology and behavior evolved by combining a generative encoding with voxel-resolution soft, actuatable materials. The soft robot gallops from left to right across the image with a dog-like gait.

An open question is whether computational evolution will produce more natural, complex forms if it is able to create organisms out of many material types. Here we test that hypothesis by evolving morphologies composed of voxels of different materials. They can be hard or soft, analogous to bone or soft tissue, and inert or expandable, analogous to supportive tissue or muscle. Contiguous patches of homogeneous voxels can be thought of as different tissue structures.

Another hypothesis is that the encodings used in previous work limited the design space. Direct encodings lack the regularity and evolvability necessary to consistently produce regular morphologies and coordinated behaviors [9, 6, 34, 16], and overly regular indirect encodings constrict the design space by disallowing complex regularities with variation [16, 31, 34]. We test this hypothesis by evolving morphologies with the CPPN-NEAT encoding [31], which has been shown to create complex regularities such as symmetry and repetition, both with and without variation (Fig. 3).

Reprinted with permissions. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. GECCO'13, July 6–10, 2013, Amsterdam, The Netherlands. Copyright 2013 ACM 978-1-4503-1963-8/13/07 ...$15.00.

CPPN-NEAT has shown these abilities in 2D images [29] and 3D objects [7] and morphologies [4]. To test the impact of the CPPN encoding, we compare it to a direct encoding. Overall, we find that evolution does utilize additional materials made available to it; their availability led to a significant amount of diverse, interesting, complex morphologies and locomotion behaviors without hindering performance. Furthermore, the generative encoding produced regular patterns of voxel 'tissue', leading to fast, effective locomotion. In contrast, the direct encoding produced no phenotypic regularity and led to poor performance.

Because it is notoriously difficult to quantify attributes such as "impressiveness" and "complexity", we make no effort to do so here. Instead, we attempt to visually represent the interesting diversity of morphologies and behaviors that evolved once evolution was provided with more materials and a sophisticated encoding. We also demonstrate the ability for this system to scale to higher resolutions and greater material diversity without hindering performance.

Finally, we investigate the effects of different fitness functions, revealing that evolution with this encoding and material palette can create different bodies and behaviors in response to different environmental and selective pressures.

1 Background

There are many Evolutionary Robotics papers with rigid-body robots [25]. However, few attempts have been made to evolve robots composed of soft materials [27], and most of those attempts are limited to only a few components. This paucity is due largely to the computational costs of simulating flexible materials and because many genetic encodings do not scale to large parameter spaces [5, 18].

The CPPN encoding abstracts how developmental biology builds natural complexity, and has been shown to produce complex, natural-appearing images and objects (Fig. 3) [29, 7, 31]. Auerbach and Bongard used this generative encoding to evolve robotic structures at finer resolutions than previous work. The systems evolved demonstrated the ability to take advantage of geometric coordinates to inform the evolution of complex bodies. However, this work was limited to rigid building blocks which were actuated by a large number of hinge joints [1, 4, 3], or had no actuation at all [2].

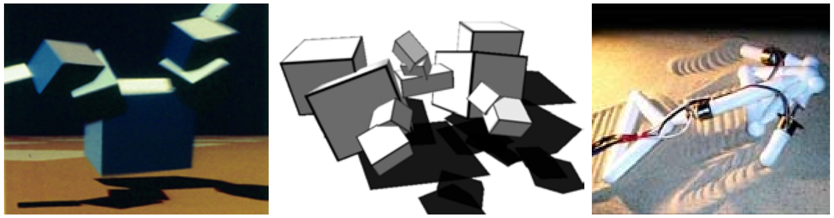


Fig. 2: (left) The scale and resolution of robots evolved by Sims in 1994; (middle) The scale and resolution at which evolutionary robotics commonly occurs today (from Lehman and Stanley in 2011); (right) The scale and resolution of robot fabrication techniques (from Lipson and Pollack, 2000).

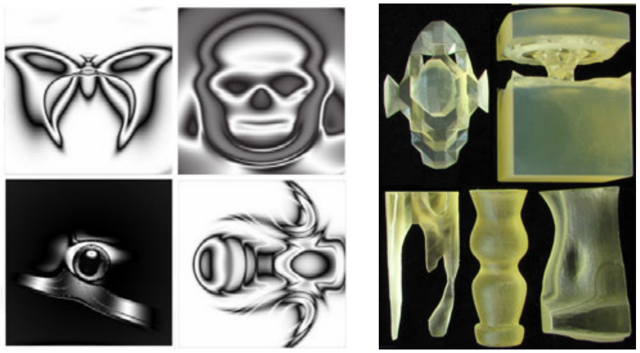


Fig. 3: (left) Examples of high resolution, complex, natural-looking images evolved with CPPN-NEAT that contain symmetry, repetition, and interesting variation; (right) Examples of CPPN-encoded 3D shapes with these same properties from (J. Clune and H. Lipson 2011).

Rigid structures limit the ability of robots to interact with their environments, especially when compared to the complex movements of structures in biology composed of muscle and connective tissue. These structures, called muscular hydrostats, often display incredible flexibility and strength; examples from biology include octopus arms or elephant trunks [35]. While soft robots can be designed that provide outstanding mobility, strength and reliability, the design process is complicated by multiple competing and difficult-to-define objectives [35]. Evolutionary algorithms excel at such problems, but have historically not been able to scale to larger robotic designs. To demonstrate that evolution can design complex, soft-bodied robots, Hiller and Lipson created a soft-voxel simulator (called VoxCAD) [11].

They showed a preliminary result that CPPNs can produce interesting locomotion morphologies, and that such designs can transfer to the real world (Fig. 4) [13]. However, this work did not take advantage of the NEAT algorithm, with its historical markings, speciation, crossover, and complexification over time - which have been shown to greatly improve the search process [33]. Additionally, these preliminary results consisted of only three trials per treatment. Here we conduct a more in-depth exploration of the capabilities of CPPNs when evolving soft robots in VoxCad.

2 METHODS

2.1 CPPN-NEAT

CPPN-NEAT has been repeatedly described in detail [31, 9, 7, 10], so we only briefly summarize it here. A compositional pattern-producing network (CPPN) is similar to a neural network, but its nodes contain multiple math functions (in this paper: sine, sigmoid, Gaussian, and linear). CPPNs evolve according to the NEAT algorithm [31]. The CPPN produces geometric output patterns that are built up from the functions of these nodes. Because the nodes have regular mathematical functions, the output patterns tend to be regular (e.g. a Gaussian function can create symmetry and a sine function can create repetition).

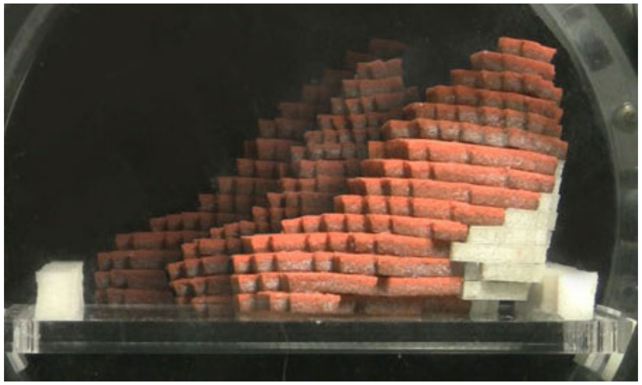


Fig. 4: A time-series example of a fabricated soft robot, which actuates with cyclic 20% volumetric actuation in a pressure chamber (J. D. Hiller and H. Lipson 2012). This proof-of-concept shows that evolved, soft-bodied robots can be physically realized. Current work is investigating soft robot actuation outside of a pressure chamber.

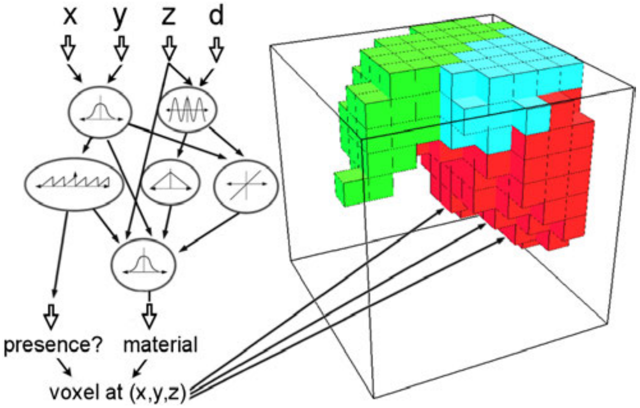


Fig. 5: A CPPN is iteratively queried for each voxel within a bounding area and produces output values as a function of the coordinates of that voxel. These outputs determine the presence of voxels and their material properties to specify a soft robot.

In this paper, each voxel has an x, y, and z coordinate that is input into the network, along with the voxel's distance from center (d). One output of the network specifies whether any material is present, while the maximum value of the 4 remaining output nodes (each representing an individual material) specifies the type of material present at that location (Fig. 5). This method of separating the presence of a phenotypic component and its parameters into separate CPPN outputs has been shown to improve performance [36]. Robots can be produced at any desired resolution. If there are multiple disconnected patches, only the most central patch is considered when producing the robot morphology.

2.2 VoxCAD

Fitness evaluations are performed in the VoxCAD soft-body simulator, which is described in detail in Hiller and Lipson 2012 [14]. The simulator efficiently models the statics, dynamics, and non-linear deformation of heterogeneous soft bodies. It also provides support for volumetric actuation of individual voxels (analogous to expanding and contracting muscles) or passive materials of varying stiffness (much like soft support tissue or rigid bone). For visualization, we display each voxel, although a smooth surface mesh could be added via the Marching Cubes algorithm [23, 7].

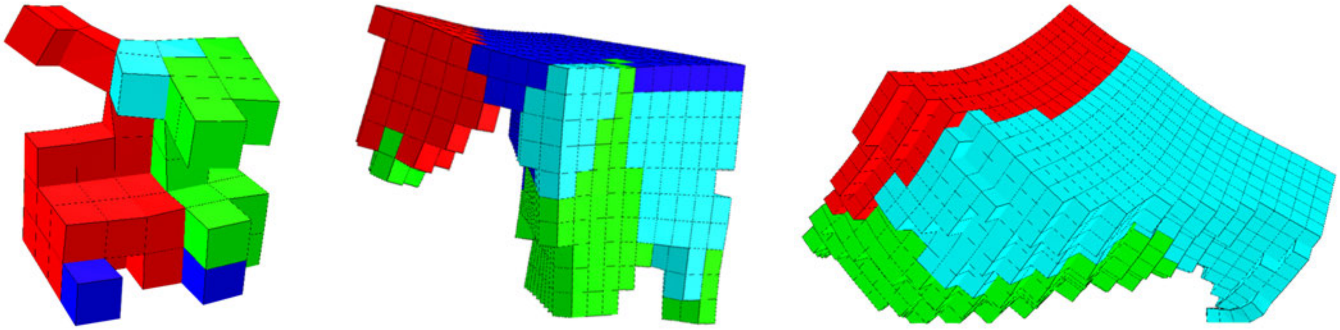


Fig. 6: CPPN-NEAT-encoded soft robots can scale to any resolution. Pictured here are soft robots sampled at voxel resolutions of $5 \times 5 \times 5$ (left), $10 \times 10 \times 10$ (center), and $20 \times 20 \times 20$ (right).

2.2.1 Materials

Following [12], there are two types of voxels: those that expand and contract at a pre-specified frequency, and passive voxels with no intrinsic actuation, which are either soft or hard. We expand upon [12] to include multiple phases of actuation. Unless otherwise noted, four materials are used: Green voxels undergo periodic volumetric actuations of 20%. Light blue voxels are soft and passive, having no intrinsic actuation, with their deformation caused solely by nearby voxels. Red voxels behave similarly to green ones, but with counter-phase actuations. Dark blue voxels are also passive, but are more stiff and resistant to deformation than light blue voxels. In treatments with less than 4 materials, voxels are added in the order above (e.g. two material treatments consist of green and light blue voxels).

2.3 GAlib

The direct encoding is from GAlib—fully described in [37]—a popular off-the-shelf genetic algorithm library from MIT. In the direct encoding genome, each voxel has its own independent values representing its presence and material outputs. The first value is binary, indicating whether a voxel at that position exists. If the voxel exists, the highest of the material property values determines the type of voxel. Thus, a $10 \times 10 \times 10$ (“10^3”) voxel soft robot with 4 possible materials would have a genome size of $10^3 \times 5 = 5000$ values.

2.4 Experimental Details

Treatments consist of 35 runs, each with a population size of 30, evolved for 1000 generations. Unless otherwise noted, fitness is the difference in the center of mass of the soft robot between initialization and the end of 10 actuation cycles. If any fitness penalties are assessed, they consist of multiplying the above fitness metric by: $1 - \frac{\text{penalty metric}}{\text{maximum penalty metric}}$. For example, if the penalty metric is the number of voxels, an organism with 400 non-empty voxels out of a possible 1000 would have its displacement multiplied by $1 - \frac{400}{1000} = 0.6$ to produce its final fitness value. Other CPPN-NEAT parameters are the same as in Clune and Lipson 2011 [7].

3 Results

Quantitative and qualitative analyses reveal that evolution in this system is able to produce effective and interesting locomoting soft robots at different voxel resolutions and using different materials. We also discover that imposing different environmental challenges in the form of penalty functions provides an increased diversity of forms, suggesting the capability to adapt to various selective pressures.

Videos of soft robot locomotion are available at `http://tinyurl.com/EvolvingSoftRobots`. So the reader may verify our subjective, qualitative assessments, we have permanently archived all evolved organisms, data, source code, and parameter settings at the Dryad Digital Repository.

3.1 Direct vs. Generative Encoding

The CPPN-NEAT generative encoding far outperforms the direct encoding (Figure 8), which is consistent with previous findings [9, 6]. The most stark difference is in the regularity of the voxel distributions (compare Figs. 1, 6, 12, 13 to Fig. 7). CPPN-NEAT soft robots consist of homogeneous patches of materials akin to tissues (e.g. one large patch of muscle, another patch of bone, etc.). The direct encoding, on the other hand, seems to randomly assign a material to each voxel. These homogeneous tissue structures are beneficial because similar types of voxels can work in a coordinated fashion to achieve the locomotion objective. For example, all the voxels in one large section of green voxels will expand at the same time, functioning as muscle tissue. This global coordination leads to jumping, bounding, stepping, and many other behaviors. In the direct encoding, each voxel works independently from–and often at odds with–its neighboring voxels, preventing coordinated behaviors. Instead, final organisms appear visually similar to those at initialization, and performance barely improves across generations (Figure 8).

Another reason for the success of the CPPN-NEAT encoding is one of the key properties of the NEAT algorithm: it starts with CPPN networks that produce simple geometric voxel patterns and *complexifies* those patterns over time [31].

3.2 Penalty Functions

To explore performance under different selective or environmental pressures, we tested four different penalty regimes. All four require the soft robot to move as far as possible, but have different restrictions. In one environment, the soft robots are penalized for their number of voxels, similar to an animal having to work harder to carry more weight. In another, the soft robots are penalized for their amount of actuatable material, analogous to the cost of expending energy to contract muscles. In a third treatment, a penalty is assessed for the number of connections (adjoining faces between voxels), akin to animals that live in warm environments and overheat if their surface area is small in comparison to their volume. Finally, there is also the baseline treatment in which no penalties are assessed.

While a cost for actuated voxels does perform significantly worse than a setup with no cost ($p = 1.9 \times 10^{-5}$ comparing final fitness values), all treatments tend to perform similarly over evolutionary time (Fig. 9). This rough equivalence suggests that the system has the ability to adapt to different cost requirements without major reductions in performance. However, drastically different types of body-plans and behaviors evolved for the different fitness functions. There are differences in the proportions of each material found in evolved organisms, indicating that evolution utilizes different material distributions to fine tune morphologies to various environments (Fig. 10). For example, when no penalty cost is assessed, more voxels are present ($p < 2 \times 10^{-13}$). When there is a cost for the number of actuated voxels, but not for support tissue, evolution uses more of these inert support materials ($p < 0.02$).

More revealing are the differences in behaviors. Fig. 11 categorizes locomotion strategies into several broad classes, and shows that different task requirements favor different classes of these behaviors. To limit subjectivity in the categorization process, we made clear category definitions, as is common in observational biology, and provide an online archive of all organisms for reader evaluation (see Sec. 3).

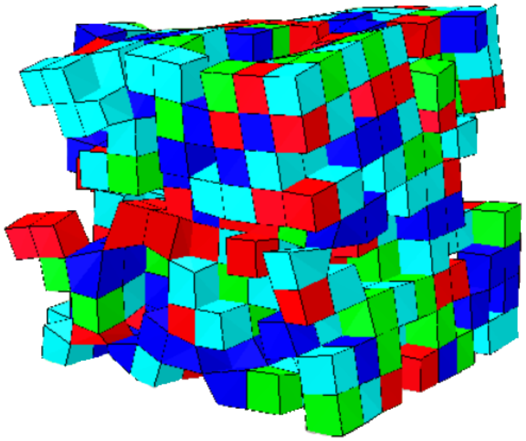


Fig. 7: A representative example of a soft robot evolved with a direct encoding. Note the lack of regularity and organization: there are few contiguous, homogeneous patches of one type of voxel. Instead, the organism appears to be composed of randomly distributed voxels . The resolution is the default 10^3.

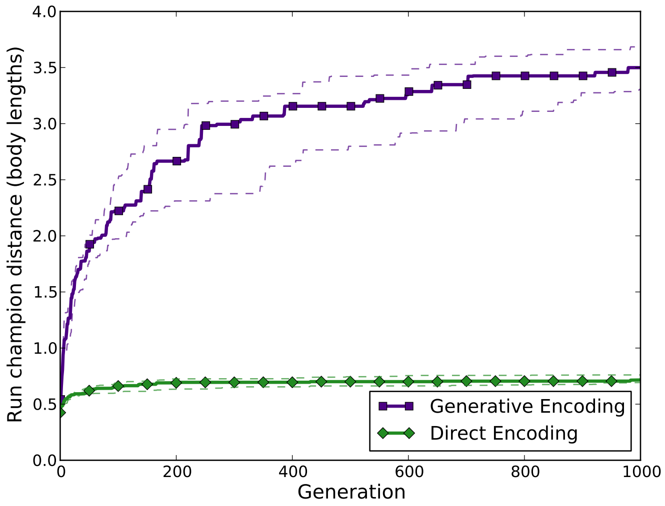


Fig. 8: The best individuals from 35 independent runs with a direct or generative encoding. Note how the generative encoding sees large improvements early in evolution, while it is exploring new locomotion types. It then settles on specific types and gradually improves coordination, timing, etc., to exploit a given strategy. The direct encoding is unable to produce globally coordinated behavior to develop new locomotion strategies, resulting in very minor improvements as it exploits its initial random forms. Here, and in all figures, thick lines are medians $\pm 95\%$ bootstrapped confidence intervals.

Fig. 12 displays the common locomotion strategies and Fig. 11 shows how frequently they evolved. They are described in order of appearance in Fig. 12. The L-Walker is named after the “L" shape its rectangular body forms, and is distinguished by its blocky form and hinge-like pivot point in the bend of the L. The Incher is named after its inchworm like behavior, in which it pulls its back leg up to its front legs by arching its back, then stretches out to flatten itself and reach its front legs forward. Its morphology is distinguished by its sharp spine and diagonal separation between actuatable materials. The Push-Pull is a fairly wide class of behaviors and is tied together by the soft robot’s powerful push with its (often large) hind leg to propel itself forward, which is usually coupled with a twisting or tipping of its front limb/head to pull itself forward between pushes. The head shape and thinner neck region are surprisingly common features. Next, the Jitter (or Bouncer) moves by bouncing its (often large) back section up and down, which pushes the creature forward. It is distinguished by its long body and is often composed mainly of a single actuatable material. The Jumper is similar in that it is often comprised of a single actuatable material, but locomotes in an upright position, springing up into the air and using its weight to angle its jumping and falling in a controlled fashion to move forward. The Wings is distinguished by its unique vertical axis of rotation. It brings its arms (or wings) in front of it, then pushes them down and out to the sides, propelling its body forward with each flapping-like motion. Fig. 13 demonstrates other, less-common behaviors that evolved.

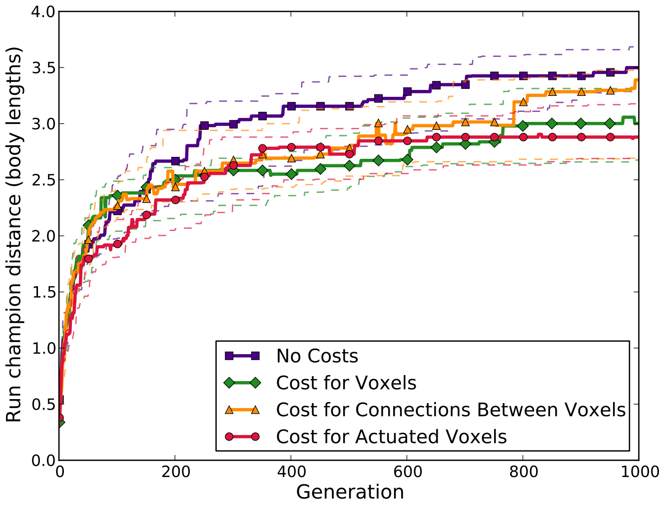


Fig. 9: Performance is mostly unaffected by different selection pressures (i.e. fitness functions).

These example locomotion strategies display the system’s ability to produce a diverse set of morphologies and behaviors, which likely stems from its access to multiple types of materials. Our results suggest that with even more materials, computational evolution could produce even more sophisticated morphologies and behaviors. Note that different behaviors show up more frequently for different task settings (Fig. 11), suggesting the ability of the system to fine tune to adapt to different selective pressures.

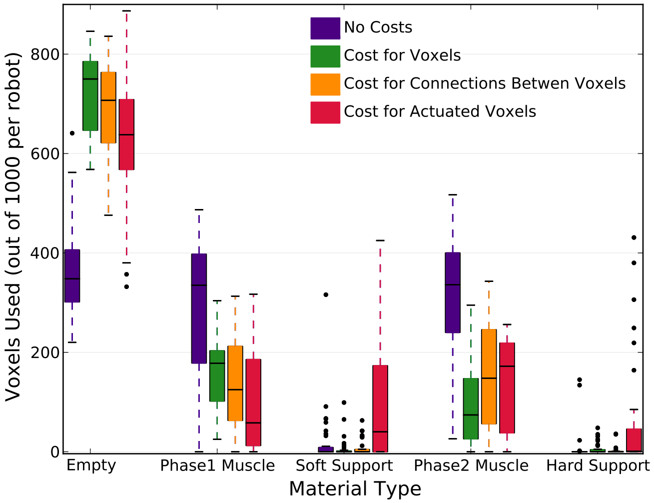


Fig. 10: The amount of each material that evolved for different cost functions, revealing the system's ability to adapt material distributions to different environments. For example, without a cost, evolution used more voxels to produce actuation ($p < 2 \times 10^{-13}$). With a cost for actuated voxels, evolution tends to use more inert support tissue ($p < 0.02$).

3.3 Material Types

To meet its full potential, this system must scale to arbitrarily large numbers of materials and resolutions. We first explore its ability to compose soft robots out of a range of materials by separately evolving soft robots with increasing numbers of materials (in the order outlined in Sec. 2.2.1). Adding a second, and then a third, material significantly improved performance (Fig. 14, $p < 2 \times 10^{-6}$), and adding a further hard, inert material did not significantly hurt performance (Fig. 14, $p = 0.68$). This improved performance suggests that CPPN-NEAT is capable of taking advantage of the increase in morphological and behavioral options. This result is interesting, as one might have expected a drop in performance associated with the need to search in a higher dimensional space and coordinate more materials.

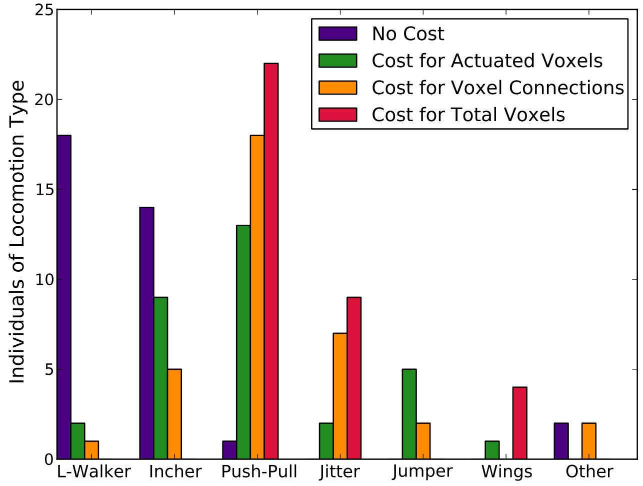


Fig. 11: Common behaviors evolved under different cost functions, summed across all runs. These behaviors are described in Sec. 3.2 and visualized in Fig. 12. Some behaviors occur more frequently under certain selective regimes. For example, the L-Walker is more common without a voxel cost, while Jitter, Jumper, and Wings do not evolve in any of the no cost runs.

3.4 Resolution

This system also is capable of scaling to higher resolution renderings of soft robots, involving increasing numbers of voxels. Fig. 6 shows example morphologies evolved at each resolution. The generative encoding tended to perform roughly the same regardless of resolution, although the computational expense of simulating large numbers of voxels prevented a rigorous investigation of the effect of resolution on performance. Faster computers will enable such research and the evolution of higher-resolution soft robots.

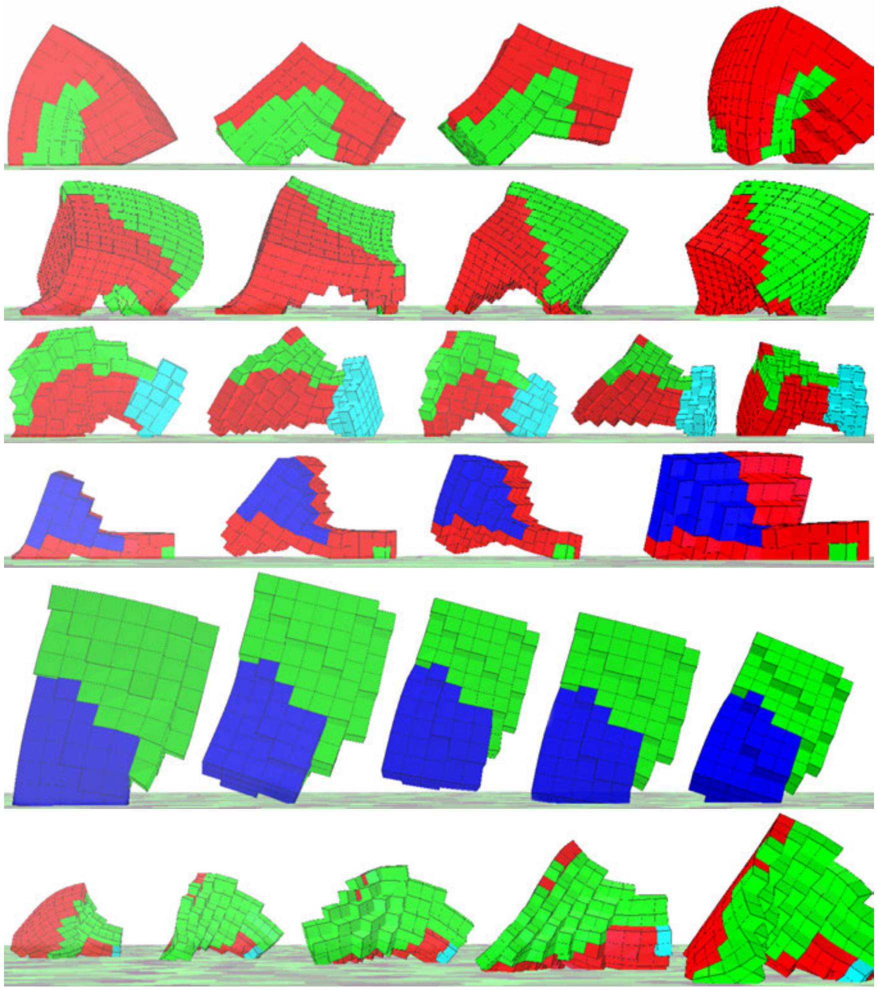


Fig. 12: Time series of common soft robot behaviors as they move from left to right across the image. From top to bottom, we refer to them as L-Walker, Incher, Push-Pull, Jitter, Jumper, and Wings. Fig. 11 reports how frequently they evolved.

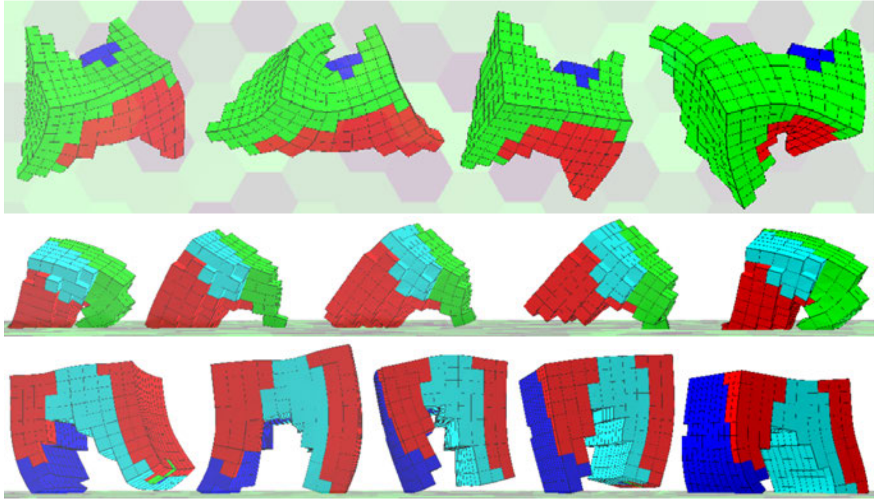


Fig. 13: Time series of other evolved strategies. (top) Opposite leg stepping creates a traditional animal walk or trot. (middle) A trunk-like appendage on the front of the robot helps to pull it forward. (bottom) A trot, quite reminiscent of a galloping horse, demonstrates the inclusion of stiff material to create bone-like support in longer appendages.

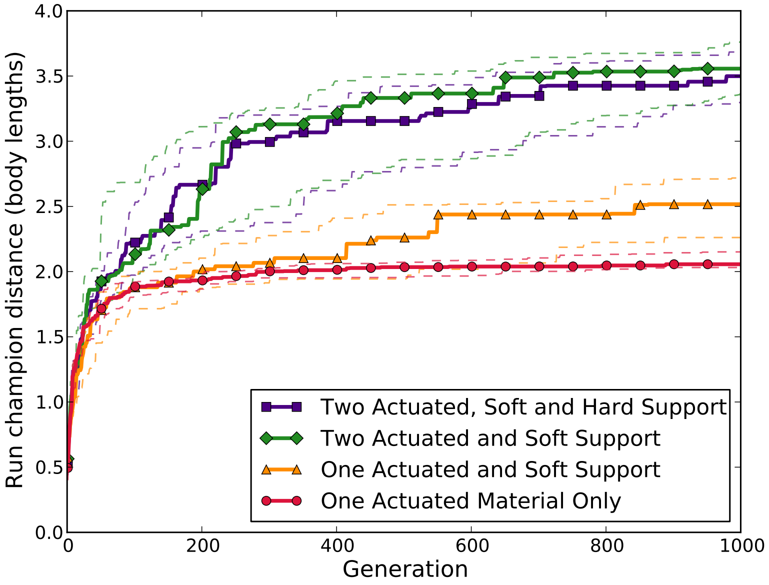


Fig. 14: The number of materials also affects performance. With only one, only simple behaviors like Jumping or Bouncing are possible, so performance peaks early and fails to discover new gaits over time. Upon adding a second material, more complex jumping and L-Walker behavior develops. When a second actuatable material is added, most behavior strategies from Fig. 12 become possible. Adding a stiff support material broadens the range of possible gaits, but is only rarely taken advantage of (such as in the bottom gallop of Fig. 13) and thus has a minimal impact on overall performance. These observational assessments may be verified, as all evolved organisms are available online (Sec. 3)

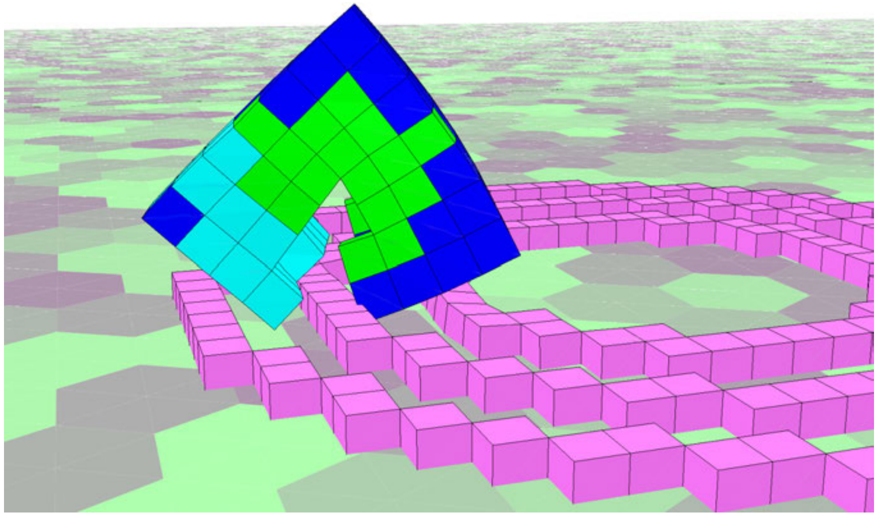


Fig. 15: An example of a soft robot that has evolved “teeth" to hook onto the obstacle rings in its environment and propel itself across them.

4 Discussion

The results show that life-like, complex, interesting morphologies and behaviors are possible when we expand the design space of evolutionary robotics to include soft materials that behave similarly to organic tissue or muscle, and search that design space with a powerful generative encoding like CPPN-NEAT. Our preliminary experiments suggest that soft robotics at the voxel resolution will someday provide complex and breathtaking demonstrations of lifelike artificial forms. Soft robotics will also showcase the ability of evolutionary design because human intuitions and engineering fare poorly in such entangled, non-linear design spaces.

We challenged multiple scientists to design fast, locomoting soft robots by hand, using the same resolution and materials. While the sample size is not sufficient to report hard data, all participants (both those with and without engineering backgrounds) were unable to produce organisms that scored higher than the evolved creatures. Participants noted the surprising difficulty of producing efficient walkers with these four materials. This preliminary experiment supports the claim that systems like the CPPN-NEAT generative encoding will increasingly highlight the effectiveness of automated design relative to a human designer.

This work shows that the presence of soft materials alone is not sufficient to provide interesting and efficient locomotion, as soft robots created from the direct encoding performed poorly. Our results are consistent with work evolving rigid-body robots that shows that generative encodings outperform direct encodings for evolutionary robotics [17, 19, 9, 6]. Unfortunately, there have been few attempts to evolve robot morphologies with CPPN-NEAT [2], and there is no consensus in the field of a proper measurement of “complexity", “interestingness", or “natural" appearance, so we cannot directly compare our soft robots to their rigid-body counterparts. However, we hope that the reader will agree about the potential of evolved soft robots upon viewing the creatures in action [`http://tinyurl.com/EvolvingSoftRobots`].

5 Future Work

The ability to evolve complex and intricate forms lends itself naturally to other questions in the field. Auerbach and Bongard have explored the relationship between environment and morphology with rigid robots in highly regular environments [4]. Because our system allows more flexibility in robot morphology and behavior, it may shed additional, or different, light on the relationship between morphology, behavior, and the environment. Preliminary results demonstrate the ability of this system to produce morphologies well suited for obstacles in their environments (Fig. 15).

While our research produced an impressive array of diverse forms, it did use a target-based fitness objective, which can hinder search [38]. Switching to modern techniques for explicitly generating diversity, such as the MOLE algorithm by Mouret and Clune [24, 8] or algorithms by Lehman and Stanley [21], has the potential to create an incredibly complex and diverse set of morphologies and behaviors.

Additionally, we are currently pursuing methods to minimize the need for expensive simulations and to evolve specific material properties instead of having a predefined palette of materials. These avenues are expected to allow increased complexity and diversity in future studies.

The HyperNEAT algorithm [32], which utilizes CPPNs, has been shown to be effective for evolving artificial neural network controllers for robots [9, 20, 6]. The same encoding from this work could thus co-evolve robot controllers and soft robot morphologies. Bongard and Pfeifer have argued that such body-brain co-evolution is critical toward progress in evolutionary robotics and artificial intelligence [26].

Soft robots have shown promise in multiple areas of robotics, such as gripping [15] or human-robot interaction [28]. The scale-invariant encoding and soft actuation from this work has potential in these other areas of soft robotics as well.

In order to compare different approaches, the field would benefit from general, accepted definitions and quantitative measures of complexity, impressiveness, and naturalness. Such metrics will enable more quantitative analyses in future studies like this one.

6 Conclusion

In this work we investigate the difficult-to-address question of why we as a field have failed to substantially improve upon the work of Karl Sims nearly two decades ago. We show that combining a powerful generative encoding based on principles of developmental biology with soft, biologically-inspired materials produces a diverse array of interesting morphologies and behaviors. The evolved organisms are qualitatively different from those evolved in previous research with more traditional rigid materials and either direct, or overly regular, encodings. The CPPN-NEAT encoding produces complex, life-like organisms with properties seen in natural organisms, such as symmetry and repetition, with and without variation. Further, it adapts to increased resolutions, numbers of available materials, and different environmental pressures by tailoring designs to different selective pressures without substantial performance degradation. Our results suggest that investigating soft robotics and modern generative encodings may offer a path towards eventually producing the next generation of impressive, computationally evolved creatures to fill artificial worlds and showcase the power of evolutionary algorithms.

7 Acknowledgments

NSF Postdoctoral Research Fellowship in Biology to Jeff Clune (DBI-1003220), NSF Graduate Research Fellowship (DGE-0707428) to Robert MacCurdy, and DARPA Grant W911NF-12-1-0449. Thanks also to Jon Hiller for help with the VoxCad simulator.

References

[1] J. E. Auerbach and J. C. Bongard. Dynamic resolution in the co-evolution of morphology and control. In *Artificial Life XII: Proc. of the 12th International Conf. on the Simulation and Synthesis of Living Systems*, 2010.

[2] J. E. Auerbach and J. C. Bongard. Evolving CPPNs to grow three-dimensional physical structures. In *Proc. of the Genetic & Evolutionary Computation Conf.*, pages 627–634. ACM, 2010.

[3] J. E. Auerbach and J. C. Bongard. On the relationship between environmental and mechanical complexity in evolved robots. In *Proc. of the Artificial Life Conf.*, pages 309–316, 2012.

[4] J. E. Auerbach and J. C. Bongard. On the relationship between environmental and morphological complexity in evolved robots. In *Proc. of the Genetic & Evolutionary Computation Conf.*, pages 521–528. ACM, 2012.

[5] J. Bongard. Evolving modular genetic regulatory networks. In *Evolutionary Computation, 2002. Proc. of the 2002 Congress on*, volume 2, pages 1872–1877. IEEE, 2002.

[6] J. Clune, B. Beckmann, C. Ofria, and R. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 2764–2771, 2009.

[7] J. Clune and H. Lipson. Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *Proc. of the European Conf. on Artificial Life*, pages 144–148, 2011.

[8] J. Clune, J.-B. Mouret, and H. Lipson. The evolutionary origins of modularity. *Proc. of the Royal Society B*, 280(20122863), 2013.

[9] J. Clune, K. O. Stanley, R. T. Pennock, and C. Ofria. On the performance of indirect encoding across the continuum of regularity. *IEEE Trans. on Evolutionary Computation*, 15(4):346–367, 2011.

[10] J. Gauci and K. O. Stanley. Generating large-scale neural networks through discovering geometric regularities. In *Proc. of the Genetic & Evolutionary Computation Conf.*, pages 997–1004. ACM, 2007.

[11] J. D. Hiller and H. Lipson. Multi-material topological optimization of structures and mechanisms. In *Proc. of the Genetic & Evolutionary Computation Conf.*, pages 1521–1528. ACM, 2009.

[12] J. D. Hiller and H. Lipson. Evolving amorphous robots. *Artificial Life XII*, pages 717–724, 2010.

[13] J. D. Hiller and H. Lipson. Automatic design and manufacture of soft robots. *IEEE Trans. on Robotics*, 28(2):457–466, 2012.

[14] J. D. Hiller and H. Lipson. Dynamic simulation of soft heterogeneous objects. *ArXiv:1212.2845*, 2012.

[15] S. Hirose and Y. Umetani. The development of soft gripper for the versatile robot hand. *Mechanism and Machine Theory*, 13(3):351–359, 1978.

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

[17] G. S. Hornby and J. B. Pollack. Evolving L-systems to generate virtual creatures. *Computers & Graphics*, 25(6):1041–1048, 2001.

[18] M. Joachimczak and B. Wróbel. Co-evolution of morphology and control of soft-bodied multicellular animats. In *Proc. of the Genetic & Evolutionary Computation Conf.*, pages 561–568. ACM, 2012.

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

[20] S. Lee, J. Yosinski, K. Glette, H. Lipson, and J. Clune. Evolving gaits for physical robots with the HyperNEAT generative encoding: the benefits of simulation., 2013.

[21] J. Lehman and K. O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proc. of the Genetic & Evolutionary Computation Conf.*, pages 211–218. ACM, 2011.

[22] H. Lipson and J. B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978, 2000.

[23] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM Siggraph*, volume 21, pages 163–169, 1987.

[24] J.-B. Mouret and J. Clune. An algorithm to create phenotype-fitness maps. *Proc. of the Artificial Life Conf.*, pages 593–594, 2012.

[25] S. Nolfi, D. Floreano, O. Miglino, and F. Mondada. How to evolve autonomous robots: Different approaches in evolutionary robotics. In *Artificial Life IV*, pages 190–197. Cambridge, MA: MIT Press, 1994.

[26] R. Pfeifer and J. C. Bongard. *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

[27] J. Rieffel, F. Saunders, S. Nadimpalli, H. Zhou, S. Hassoun, J. Rife, and B. Trimmer. Evolving soft robotic locomotion in PhysX. In *Proc. of the Genetic & Evolutionary Computation Conf.*, pages 2499–2504. ACM, 2009.

[28] S. Sanan, J. Moidel, and C. G. Atkeson. A continuum approach to safe robots for physical human interaction. In *Int'l Symposium on Quality of Life Technology*, 2011.

[29] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley. Picbreeder: evolving pictures collaboratively online. In *Proc. of the 26th SIGCHI Conf. on Human Factors in Computing Systems*, pages 1759–1768. ACM, 2008.

[30] K. Sims. Evolving virtual creatures. In *Proc. of the 21st Annual Conf. on Computer Graphics and Interactive Techniques*, pages 15–22. ACM, 1994.

[31] K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162, 2007.

[32] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

[33] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[34] K. O. Stanley and R. Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, 2003.

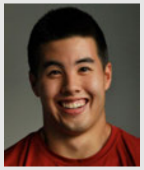
[35] D. Trivedi, C. Rahn, W. Kier, and I. Walker. Soft robotics: Biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics*, 5(3):99–117, 2008.

[36] P. Verbancsics and K. O. Stanley. Constraining connectivity to encourage modularity in HyperNEAT. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1483–1490. ACM, 2011.

[37] M. Wall. Galib: A c++ library of genetic algorithm components. *Mechanical Engineering Department, Massachusetts Institute of Technology*, 87, 1996.

[38] B. G. Woolley and K. O. Stanley. On the deleterious effects of a priori objectives on evolution and representation. In *Proc. of the Genetic and Evolutionary Computation Conf.*, pages 957–964. ACM, 2011.

About the authors



Nick Cheney is a Computational Biology PhD student in the Creative Machine Lab at Cornell University, where he is advised by Professor Hod Lipson. He is also a NASA Space Technology Research Fellow, spending time in the Adaptive Control and Evolvable Systems Group in the Intelligent Systems Division of the NASA Ames Research Center under the mentorship of Greg Hornby. He holds a Bachelor of Science in Applied Mathematics with a concentration in Complex Systems from the University of Vermont, advised by Professor Josh Bongard. He is generally interested in Complex Adaptive Systems, with specific research interests in the co-evolution of body-brain-environment systems and the study of evolvability within these systems. His work on evolving soft robots won the Virtual Creatures GECCO competition in 2014 and the “Most Entertaining Video” prize at the AAAI Video Competition in 2013. His research has appeared in media outlets such as NBC, Discover Magazine, Discovery News, and Popular Science.

Homepage: http://www.ncheney.com
Email: nac93@cornell.edu

Robert MacCurdy is a PhD candidate in the Sibley School of Mechanical and Aerospace Engineering at Cornell University. He holds Undergraduate degrees in Physics and Electrical Engineering. His research interests include Modular Robotics, Additive Manufacturing, Digital Materials, Heuristic Optimization and Bio-instrumentation. He previously worked at Cornell’s Lab of Ornithology where he designed systems to gather behavioral data on wild animals in their native habitats, and has deployed these systems worldwide.

Email: rbm7@cornell.edu

Jeff Clune is an assistant professor in the Computer Science Department at the University of Wyoming where he directs the Evolving AI lab. Prior to that he was a visiting scientist at Cornell University, funded by a Postdoctoral Research Fellowship from the US National Science Foundation. Jeff has a bachelor’s degree in philosophy from the University of Michigan and a Ph.D. in computer science and a master’s degree in philosophy from Michigan State University. He investigates how to evolve artificially intelligent robots. Specifically, he works to increase the complexity and structural organization of neural networks, making them more regular, modular, and hierarchical. He also investigates how to create evolutionary algorithms that produce a vast diversity of interesting phenotypes. Jeff was the co-chair of the Generative and Developmental Systems track at GECCO (2010 & 2011) and won the best paper award in that track in 2009. His work on evolving soft robots won the Virtual Creatures GECCO competition in 2014 and the “Most Entertaining Video” prize at the AAAI Video Competition in 2013. His research has appeared in news outlets such as Science, NPR, NBC, National Geographic, the BBC, the New Scientist, MIT’s Technology Review, the Daily Telegraph, Slashdot, and U.S. News & World Report.

Homepage: http://EvolvingAI.com
Email: jeffclune@uwyo.edu

Hod Lipson is a professor of engineering at Cornell University in Ithaca, New York, and a co-author of the recent book “Fabricated: The New World of 3D printing”. His work on self-aware and self-replicating robots, food printing, and bio-printing has received widespread media coverage including The New York Times, The Wall Street Journal, Newsweek, Time, CNN, and the National Public Radio. Lipson has co-authored over 200 technical papers and speaks frequently at high-profile venues such as TED and the US National Academies. Hod directs the Creative Machines Lab, which pioneers new ways to make machines that create, and machines that are creative.

Email: hod.lipson@cornell.edu

evo*2015
8-10 April

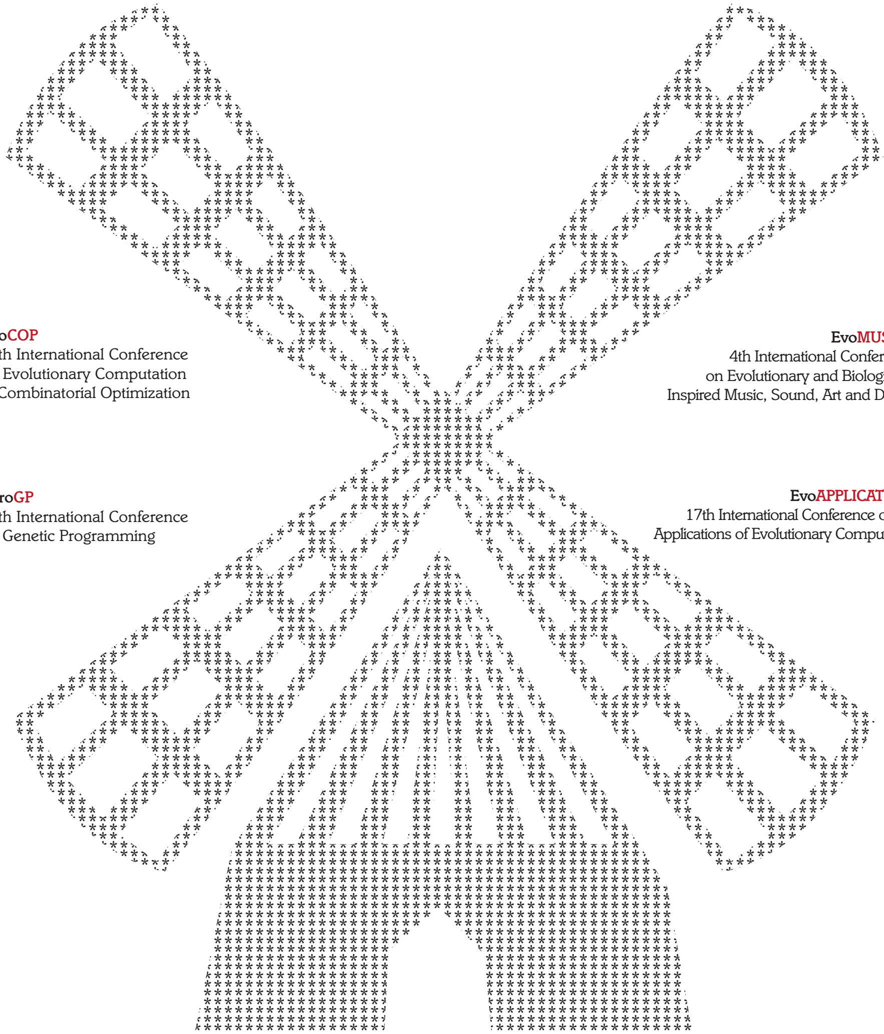
Copenhagen - Denmark

www.evostar.org

Submission deadline
15 November 2014

EVO* coordinator
Jennifer Willies, Edinburgh
Napier University, UK

Local Chairs
Paolo Burelli, Aalborg University
Julian Togelius, IT University of Copenhagen



EvoCOP
15th International Conference
on Evolutionary Computation
in Combinatorial Optimization

EvoMUSART
4th International Conference
on Evolutionary and Biologically
Inspired Music, Sound, Art and Design

EuroGP
18th International Conference
on Genetic Programming

EvoAPPLICATIONS
17th International Conference on the
Applications of Evolutionary Computation

Conferences Chairs: • Penousal Machado • Malcolm Heywood • James McDermott • Gabriela Ochoa • Francisco Chicano • Colin Johnson • Adrián Carballal • João Correia • Antonio Mora

Evo* Publicity Chairs: • Mauro Castelli • Pablo García Sánchez

BEACON Center Teams Biologists with Computer Scientists and Engineers to Study Evolutionary Processes and Solve Problems

In 2010, the U.S. National Science Foundation created BEACON Center for the Study of Evolution in Action, as one of five new Science and Technology Centers funded with 5-year, $25 million grants (renewable for a total of 10 years). In addition to our ”home,” Michigan State University, BEACON includes four partner universities——North Carolina A&T State University, University of Idaho, University of Texas Austin and University of Washington.

BEACON has grown rapidly, now having more than 300 participants, and more than 100 additional affiliates. We conduct many activities designed to bring evolutionary biologists together with computer scientists and with engineers using evolutionary computation to solve real-world problems. Graduate courses for new BEACONites allow the evolutionary biologists to learn computational methods and the computer scientists to learn about evolutionary biology. Then a follow-on course teams them to study a problem of their choice, usually resulting in a publication.

Each week sees at least one BEACON-wide (videoconferenced 5-university) seminar and additional meetings of many research teams, some of them “area” meetings where 15 or 20 graduate students, postdocs and faculty members make small presentations updating the team on progress and getting ideas from the group on interesting questions to pursue. The questions and suggestions are very likely to come from students in other disciplines.

BEACON funds about 50 “seed” projects each year, helping to put together multi-disciplinary teams to initiate projects that may eventually lead to externally funded research. Projects are often across disciplines and across the partner universities. BEACON researchers are making discoveries about evolution that are appearing frequently in such prestigious journals as Science, Nature, and Proceedings of the National Academy of Sciences.

What Can I Do with BEACON—as a Grad Student? Postdoc? Faculty Member?

BEACON funds work within its five member universities, but can also fund BEACON’s activities in projects with other universities in the U.S. or abroad. BEACON has space to host visiting scholars from groups it collaborates with, and can fund travel of BEACONites to work with their partners. BEACON often hosts faculty members on sabbatical from another university to work with someone in BEACON. There are about 40 postdoctoral research associates who are BEACON participants at any time, some funded on BEACON seed projects. While most BEACON graduate students are U.S. citizens or permanent residents, a limited number of international graduate students also receive BEACON support as research assistants. Many graduate students bring national fellowships to BEACON universities or earn them while a part of BEACON.

BEACONites posing at the BEACON booth at GECCO 2014

Search for the John R. Koza Endowed Chair in Genetic Programming

BEACON Center for the Study of Evolution in Action will shortly begin a search to fill the newly created John R. Koza Endowed Chair in Genetic Programming. The gift from John Koza will bring a new faculty member to BEACON and to some department in the College of Engineering (which, at MSU, includes the Department of Computer Science and Engineering). Applications will be sought from faculty members anywhere outside Michigan State University whose research has centered on genetic programming.

Further Information

Interested faculty members should watch the BEACON website, http://beacon-center.org/ for the announcement of the application procedures for the new chair, or may inquire of Erik Goodman, BEACON Director, at goodman@egr.msu.edu

New Theses

Personalized Procedural Map Generation in Games via Evolutionary Algorithms

Doctoral Thesis by William L. Raffe

In digital games, the *map* (sometimes referred to as the *level*) is the virtual environment that outlines the boundaries of play, aids in establishing rule systems, and supports the narrative. It also directly influences the challenges that a player will experience and the pace of gameplay, a property that has previously been linked to a player's enjoyment of a game [1]. In most industry leading games, creating maps is a lengthy manual process conducted by highly trained teams of designers. However, for many decades *procedural content generation* (PCG) techniques have posed as an alternative to provide players with a larger range of experiences than would normally be possible. In recent years, PCG has even been proposed as a means of tailoring game content to meet the preferences and skills of a specific player, in what has been termed *Experience-driven PCG* (EDPCG) [2].

This thesis contributes to the growing EDPCG research field with a focus on personalizing maps. Here, the EDPCG techniques are used within a Search-based PCG (SBPCG) [3] framework, utilizing evolutionary algorithms to search for maps that are appropriate for an individual player. Evolution is a common strategy in SBPCG as it provides a logical means of evaluating map candidates and iteratively improving them over multiple generations of recombination and mutation. Furthermore, this thesis investigates a decomposed approach to map generation, using separate evolutionary cycles, genetic representations, and fitness evaluators for two aspects of a map: the *geometry* and *content layout*. The geometry of a map defines the boundaries of play and the location of static virtual objects. Meanwhile, the content layout describes the location and quantity of interactive game assets, such as enemies and pick-ups. Both of these components affect a player's experience to varying degrees in different game genres but are typically related in that the content layout must be within the geometry.

Finally, as the maps should be appropriate for an individual player, both direct interactive evolutionary computing (IEC) and player preference modeling are investigated as methods of collecting, interpreting, and utilizing knowledge about the player's desires.

The thesis starts with a brief foray into evolutionary terrains. This work was conducted as an initial study into optimizing the most common type of base map geometry and ignoring the content layout completely. Terrains were generated by extracting uniform patches from user-provided sample terrains, recombining them in a grid based genetic representation, and rendering a larger terrain by stitching the patches back together again. The result was a content authoring tool that utilized IEC with both parent selection and gene selection to expedite the terrain creation process for novice designers [4].

However, the centrepiece of the thesis is an unsupervised public experiment on an online map personalization solution. In this solution, the geometry and content layout are linked through hierarchical optimization; first optimizing the geometry of a map and then using that as input to the content layout optimization. The geometry is represented as a custom made fixed n-ary tree that connects pre-made room and corridor templates together into a constrained tree structure and is again evaluated through IEC. The density of content within each room (node) of the geometry tree is then calculated by using the tree coordinates of each node as input to a Compositional Pattern-Producing Network (CPPN) [5] and translating the output as quantities of the various enemies and pick-ups within that room. IEC was deemed inappropriate as a fitness evaluation mechanism here because a content layout candidate could not be as easily visualized to the player. Instead, for each player-selected geometry, roughly 200 generations of NeuroEvolution of Augmenting Topologies (NEAT) [6] (with 50 CPPN candidates in each generation) are evaluated through a learned per-player preference model that is based around the paradigm of model-based and content-based recommender system (RS).

This RS player model requires that players provide a single rating value on how much they enjoyed a map after they play it. This rating is then combined with extracted map features to train a Naive Bayes classifier. That classifier then predicts the probability that a player will enjoy a map that results from a CPPN candidate and thus CPPN-NEAT is being used to reduce the number of RS evaluations need to provide a good recommendation. Further details of the system are presented in an initial report that analyzes the experiences of three sample players [7].

In summary, the primary contributions of this thesis are: 1) the use of decomposed optimization to personalize the geometry and content layout of a map in two separate, yet integrated, evolutionary processes; 2) two geometry generation solutions, one for outdoor terrains and the other of interior spaces, that both function by re-combining pre-made map segments and are controlled via IEC; 3) the application of CPPN-NEAT to the map generation process, specifically for determining the density of content given a map location; 4) an initial exploration into using knowledge from the RS field to create a player preference model to be used during the fitness evaluation of a SBPCG system; 6) an example of rigorous statistical analysis of a personalized PCG system deployed in an unsupervised public experiment; and 5) the introduction of the learning trend metrics, which give a clearer indication of RS performance in environments with noisy user data over traditional accuracy measurements.

References

[1] T. W. Malone, “What makes things fun to learn? a study of intrinsically motivating computer games.” *Pipeline*, vol. 6, no. 2, pp. 50–51, 1981.

[2] G. N. Yannakakis and J. Togelius, “Experience-driven procedural content generation,” *IEEE Transactions on Affective Computing*, vol. 2, no. 3, pp. 147–161, Jul 2011.

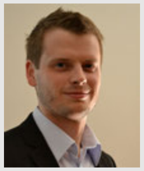
[3] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-based procedural content generation: A taxonomy and survey,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 172–186, Sep 2011.

[4] W. L. Raffe, F. Zambetta, and X. Li, “Evolving patch-based terrains for use in video games,” in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 363–370.

[5] K. O. Stanley, “Compositional pattern producing networks: A novel abstraction of development,” *Genetic Programming and Evolvable Machines*, vol. 8, no. 2, pp. 131–162, Jun 2007.

[6] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[7] W. L. Raffe, F. Zambetta, and X. Li, “Neuroevolution of content layout in the PCG: Angry Bots video game,” in *Evolutionary Computation (CEC), IEEE Congress on*. IEEE, 2013.



William L. Raffe received a B.CompSci, B.CompSci (Hons) in Computer Science in 2008 and 2009, respectively, from the School of Computer Science and Information Technology at RMIT University, Melbourne (Australia). He completed his Ph.D. in Computer Science in 2014, having been supervised by Dr. Fabio Zambetta and Dr. Xiaodong Li, both of RMIT University. William is currently a postdoctoral research fellow at the same institution, investigating player modeling for adaptive virtual-physical play in water environments. He is also a sessional lecturer, a member of the Golden Key International Honor Society, and a regularly attending member of the RMIT Evolutionary Computing and Machine Learning Group. His research interests include personalized procedural content generation, adaptive game A.I., player preference modeling, and exergaming, to all of which he focuses on applying machine learning and metaheuristic optimization techniques.

Email: william.raffe@rmit.edu.au
Homepage: WWW
Dissertation: pdf

Calls and Calendar

August 2014

IEEE Conference on Computational Intelligence and Games (CIG-2014)

August 26 - 29, 2014, Dortmund, Germany

Homepage: http://www.cig2014.de

Conference: August 26 - 29, 2014

Games can be used as a challenging scenery for benchmarking methods from computational intelligence since they provide dynamic and competitive elements that are germane to real-world problems. This conference brings together leading researchers and practitioners from academia and industry to discuss recent advances and explore future directions in this field.

The IEEE Conference on Computational Intelligence and Games is the premier annual event for researchers applying computational and artificial intelligence techniques to games. The domain of the conference includes all sorts of CI/AI applied to all sorts of games, including board games, video games and mathematical games. The yearly event series started in 2005 as symposium, and is a conference since 2009. An overview over the past CIG conferences is available at hrefhttp://www.ieee-cig.orgwww.ieee-cig.org, where you also find the proceedings. CIG 2014 will be hosted in the Park Inn hotel in the city center of Dortmund, a vibrant, technology-oriented city in the Ruhr area, Germany's largest metropolitan area with around 5 million people. The conference will consist of a single track of oral presentations, tutorial and workshop/special sessions, and live competitions. The proceedings will be placed in IEEE Xplore, and made freely available on the conference website after the conference.

Topics of interest include, but are not limited to:

- Learning in games
- Procedural content generation
- Player/opponent modeling in games
- Player affective modeling
- Player satisfaction and experience in games
- Computational and articial intelligence based game design
- Intelligent interactive narrative
- Theoretical or experimental analysis of CI techniques for games
- Non-player characters in games
- Comparative studies and game-based benchmarking
- Applications of game theory

General Chairs

Günter Rudolph, TU Dortmund, Germany
Mike Preuss, WWU Münster, Germany

Program Chairs

Mirjam Eladhari, University of Malta
Moshe Sipper, Ben-Gurion University of the Negev, Israel

Tutorials/Special Sessions Chair

Philip Hingston, Edith Cowan University, Perth, Australia

Competition Chair

Simon Lucas, University of Essex, UK

Keynote Chair

Gillian Smith, Northeastern University, Boston, USA

Proceedings Chair

Paolo Burelli, Aalborg University, Copenhagen, Denmark

September 2014

PPSN 2014 – International Conference on Parallel Problem Solving From Nature

September 13-17, 2014, Ljubljana, Slovenia

Homepage: http://ppsn2014.ijs.si

The 13th International Conference on Parallel Problem Solving from Nature (PPSN XIII) will be organized by the Jožef Stefan Institute, Ljubljana, Slovenia, and held at the Ljubljana Exhibition and Convention Centre on September 13-17, 2014. The conference aims to bring together researchers and practitioners in the field of Natural Computing. Natural Computing is the study of computational systems that use ideas and get inspiration from natural systems, including biological, ecological, physical, chemical, and social systems. It is a fast-growing interdisciplinary field in which a range of techniques and methods are studied for dealing with large, complex, and dynamic problems with various sources of potential uncertainties.

Paper Presentation Following the well-established tradition of PPSN conferences, all accepted papers will be presented during poster sessions. Each session will contain several papers, and will begin by a plenary quick overview of all papers in that session by a major researcher in the field. Past experiences have shown that such presentation format led to more interactions between participants and to deeper understanding of the papers.

General Chair

Bogdan Filipič, Jožef Stefan Institute, Slovenia

Honorary Chair

Hans-Paul Schwefel (Tech. Universität Dortmund, DE)

Program Co-Chairs

Thomas Bartz-Beielstein, Cologne University of Applied Sciences, DE
Jürgen Branke, University of Warwick, UK
Jim Smith, University of the West of England, UK

Tutorials Chairs

Shih-Hsi "Alex" Liu, California State University, Fresno, USA
Marjan Mernik, University of Maribor, Slovenia

Workshop Chairs

Evert Haasdijk, VU University Amsterdam, The Netherlands
Tea Tušar, Jožef Stefan Institute, Slovenia

Publication Chair

Jurij Šilc, Jožef Stefan Institute, Slovenia

Local Organizer

Gregor Papa, Jožef Stefan Institute, Slovenia

January 2015

Learning and Intelligent OptimizatioN Conference (LION9)

January 12-16, 2015, Lille, France

Submission deadline: October 10, 2014

Homepage: http://www.lifl.fr/LION9/

The large variety of heuristic algorithms for hard optimization problems raises numerous interesting and challenging issues. Practitioners are confronted with the burden of selecting the most appropriate method, in many cases through an expensive algorithm configuration and parameter tuning process, and subject to a steep learning curve. Scientists seek theoretical insights and demand a sound experimental methodology for evaluating algorithms and assessing strengths and weaknesses. A necessary prerequisite for this effort is a clear separation between the algorithm and the experimenter, who, in too many cases, is "in the loop" as a crucial intelligent learning component. Both issues are related to designing and engineering ways of "learning" about the performance of different techniques, and ways of using past experience about the algorithm behavior to improve performance in the future. Intelligent learning schemes for mining the knowledge obtained from different runs or during a single run can improve the algorithm development and design process and simplify the applications of high-performance optimization methods. Combinations of algorithms can further improve the robustness and performance of the individual components provided that sufficient knowledge of the relationship between problem instance characteristics and algorithm performance is obtained.

This meeting, which continues the successful series of LION events (see LION 5 in Rome, LION 6 in Paris, LION 7 in Catania, and LION 8 in Gainesville), is exploring the intersections and uncharted territories between machine learning, artificial intelligence, mathematical programming and algorithms for hard optimization problems. The main purpose of the event is to bring together experts from these areas to discuss new ideas and methods, challenges and opportunities in various application areas, general trends and specific developments.

Conference Organizers:

Clarisse Dhaenens
Laetitia Jourdan
Marie-Eléonore Marmion

Important Dates

Paper submission:	October 10, 2014
Author Notification:	November 25, 2014
Registration:	December 17, 2014
Camera ready:	January 3, 2015
Conference:	January 12-16, 2015

FOGA XIII – Foundation of Genetic Algorithms

January 17-20, 2015, Aberystwyth, Wales, UK
Homepage: http://foga2015.dcs.aber.ac.uk/
Deadline: August 31, 2014

We invite submissions to the 2015 ACM/SIGEVO Foundations of Genetic Algorithms XIII (FOGA 2015) which will be held 17–20 January 2015 in Aberystwyth, Wales, UK.

FOGA is the premier event on the theoretical foundations of evolutionary computation and all kinds of randomised search heuristics, including but not limited to evolutionary algorithms, ant colony optimisation, artificial immune systems and particle swarm optimisation. Accepted papers will be published in post-conference proceedings by ACM Press.

The goal of FOGA is to advance the theoretical understanding of evolutionary computation and all kinds of randomised search heuristics, promote theoretical work to the wider community and contribute to making randomised search heuristics more useful in practice.

We particularly encourage submissions bridging theory and practice. In addition to strict mathematical investigations, experimental studies contributing towards the theoretical foundations of evolutionary computation methods are also welcome. Topics include but are not limited to runtime analysis; fitness landscapes and problem difficulty; single- and multi-objective optimisation problems; stochastic and dynamic environments; population dynamics; statistical approaches; self-adaptation; black-box complexity; working principles of all kinds of randomised search heuristics.

Organizers

Jun He	Aberystwyth University, Wales, UK
Thomas Jansen	Aberystwyth University, Wales, UK
Gabriela Ochoa	University of Stirling, Scotland, UK
Christine Zarges	University of Birmingham, England, UK

Important dates

Paper submission	31 August, 2014
Author notification	1 November, 2014
Standard registration	30 November, 2014
FOGA 2015	17–20 January, 2015
Post-proceedings deadline	28 February, 2015

April 2015

Evostar 2015 - EuroGP, EvoCOP, EvoBIO and EvoWorkshops

April 8-10, 2015, Copenhagen, Denmark
Submission deadline: November 15, 2014
Homepage: www.evostar.org

EvoStar comprises of five co-located conferences run each spring at different locations throughout Europe. These events arose out of workshops originally developed by EvoNet, the Network of Excellence in Evolutionary Computing, established by the Information Societies Technology Programme of the European Commission, and they represent a continuity of research collaboration stretching back nearly 20 years.

The five conferences include:

- EuroGP 18th European Conference on Genetic Programming
- EvoBIO 12th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Computational Biology
- EvoCOP 15th European Conference on Evolutionary Computation in Combinatorial Optimisation
- EvoMUSART 4rd International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design
- EvoApplications 16th European Conference on the Applications of Evolutionary and bio-inspired Computation including the following tracks
  - EvoCOMNET Application of Nature-inspired Techniques for Communication Networks and other Parallel and Distributed Systems
  - EvoCOMPLEX Applications of algorithms and complex systems
  - EvoENERGY Evolutionary Algorithms in Energy Applications
  - EvoFIN Track on Evolutionary Computation in Finance and Economics
  - EvoGAMES Bio-inspired Algorithms in Games
  - EvoHOT Bio-Inspired Heuristics for Design Automation
  - EvoIASP Evolutionary computation in image analysis, signal processing and pattern recognition
  - EvoINDUSTRY The application of Nature-Inspired Techniques in industrial settings
  - EvoNUM Bio-inspired algorithms for continuous parameter optimisation
  - EvoPAR Parallel and distributed Infrastructures
  - EvoRISK Computational Intelligence for Risk Management, Security and Defense Applications
  - EvoROBOT Evolutionary Computation in Robotics
  - EvoSTOC Evolutionary Algorithms in Stochastic and Dynamic Environments

Featuring the latest in theoretical and applied research, EVO* topics include recent genetic programming challenges, evolutionary and other meta-heuristic approaches for combinatorial optimisation, evolutionary algorithms, machine learning and data mining techniques in the biosciences, in numerical optimisation, in music and art domains, in image analysis and signal processing, in hardware optimisation and in a wide range of applications to scientific, industrial, financial and other real-world problems.

EVO* Poster

You can download the EVO* poster advertisement in PDF format here

EVO* Call for Papers

You can access the call for papers of all the EVO* conferences here.

EVO* Coordinator:

Jennifer Willies, Napier University, United Kingdom
j.willies@napier.ac.uk

General Chairs:

Penousal Machado, Malcom Heywood, James McDermott, Gabriela Ochoa, Francisco Chicano, Colin Johnson, Adrian Carballai, João Correia, Antonio Mora

Local Chair:

Paolo Burelli, Aalborg University
Julian Togelius, IT University of Copenhagen

Publicity Chair:

Mauro Castelli & Paolo García Sánchez

Important Dates

Submission Deadline:	15 November 2014
Notification:	07 January 2015
Camera-ready:	21 January 2015
Conference:	8-10 April 2015

May 2015

2015 IEEE Congress on Evolutionary Computation (CEC 2015)
May 25-28, 2015, Sendai, Japan
Homepage: http://sites.ieee.org/cec2015/
Deadline December 19, 2014

The annual IEEE CEC is one of the leading events in the field of evolutionary computation. It covers all topics in evolutionary computation including: Ant colony optimization, Artificial immune systems, Coevolutionary systems, Cultural algorithms, Differential evolution, Estimation of distribution algorithms, Evolutionary programming, Evolution strategies, Genetic algorithms, Genetic programming, Heuristics, metaheuristics and hyper-heuristics, Interactive evolutionary computation, Learning classifier systems, Memetic, multi-meme and hybrid algorithms, Molecular and quantum computing, Multi-objective evolutionary algorithms, Parallel and distributed algorithms, Particle swarm optimization, Theory and Implementation, Adaptive dynamic programming and reinforcement learning, Coevolution and collective behavior, Convergence, scalability and complexity analysis, Evolutionary computation theory, Representation and operators, Self-adaptation in evolutionary computation, Optimization, Numerical optimization, Discrete and combinatorial optimization, Multiobjective optimization.

IEEE CEC 2015 will feature a world-class conference that aims to bring together researchers and practitioners in the field of evolutionary computation and computational intelligence from all around the globe. Technical exchanges within the research community will encompass keynote lectures, regular and special sessions, tutorials, and competitions as well as poster presentations. In addition, participants will be treated to a series of social functions, receptions, and networking to establish new connections and foster everlasting friendship among fellow counterparts.

Important Dates:

- Competition Proposals Due: September 26, 2014
- Tutorial Proposals Due: January 9, 2015
- Special Session Proposals Due: October 31, 2014
- Paper Submission Due: December 19, 2014

More information can be found at: http://sites.ieee.org/cec2015/.

About the Newsletter

SIGEVOlution is the newsletter of SIGEVO, the ACM Special Interest Group on Genetic and Evolutionary Computation.

To join SIGEVO, please follow this link [WWW]

Contributing to SIGEVOlution

We solicit contributions in the following categories:

Art: Are you working with Evolutionary Art? We are always looking for nice evolutionary art for the cover page of the newsletter.

Short surveys and position papers: We invite short surveys and position papers in EC and EC related areas. We are also interested in applications of EC technologies that have solved interesting and important problems.

Software: Are you are a developer of an EC software and you wish to tell us about it? Then, send us a short summary or a short tutorial of your software.

Lost Gems: Did you read an interesting EC paper that, in your opinion, did not receive enough attention or should be rediscovered? Then send us a page about it.

Dissertations: We invite short summaries, around a page, of theses in EC-related areas that have been recently discussed and are available online.

Meetings Reports: Did you participate to an interesting EC-related event? Would you be willing to tell us about it? Then, send us a short summary, around half a page, about the event.

Forthcoming Events: If you have an EC event you wish to announce, this is the place.

News and Announcements: Is there anything you wish to announce? This is the place.

Letters: If you want to ask or to say something to SIGEVO members, please write us a letter!

Suggestions: If you have a suggestion about how to improve the newsletter, please send us an email.

Contributions will be reviewed by members of the newsletter board.

We accept contributions in $\LaTeX$, MS Word, and plain text.

Enquiries about submissions and contributions can be emailed to editor@sigevolution.org.

All the issues of SIGEVOlution are also available online at www.sigevolution.org.

Notice to Contributing Authors to SIG Newsletters

By submitting your article for distribution in the Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to include the article in the ACM Digital Library
- to allow users to copy and distribute the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.