Implicit Evolvability:
An Investigation into the Evolvability of an Embryogeny

Sanjeev Kumar

Department of Computer Science,
University College London,
Gower Street, London WC1E 6BT, UK.

S.Kumar@cs.ucl.ac.uk
+44 (0) 20 7419 2878

Peter Bentley

Department of Computer Science,
University College London,
Gower Street, London WC1E 6BT, UK.

P.Bentley@cs.ucl.ac.uk
+44 (0) 20 7391 1329

Abstract

This paper investigates the evolvability of an implicit embryogeny-based representation for the evolution of 3-D morphologies. Previous results using this representation have shown that this particular incarnation of an implicit embryogeny does not lend itself well to evolution. Two different experiments are described, the results of which suggest that a many-to-one genotype-to-phenotype mapping is not sufficient to ensure evolvability. The paper concludes by suggesting attributes that a better representation should have.

1 INTRODUCTION

The Genetic Algorithm (GA) (Holland, 1975; Goldberg, 1989) has been around since the 1970s and is based on evolution in nature. GAs require a coded representation of the solution known as a *genotype.* A population of genotypes (coded candidate solution) is then created and maintained. Genetic operators such as recombination and mutation are then applied to the genotypes. A fitness function encapsulating the essence of the problem is then applied to evaluate the performance of each genotype's corresponding *phenotype* (or candidate solution).

The genetic algorithm is the only type of Evolutionary Algorithm (EA) that makes an explicit distinction between genotype and phenotype. Nature has exploited this distinction by evolving a complex mapping between genotype and phenotype, enabling the evolution of organisms far more complex than anything our EAs have managed to evolve. Despite this source of inspiration, little work has been done into the nature of the mapping between genotype and phenotype. A complex mapping is not alone sufficient for the evolution of complex solutions. We also need a deeper understanding of the nature of evolvability. Such an understanding would permit us to evolve and identify complex solutions in solution space.

This paper looks at the evolvability of an instance of a special type of genotype-to-phenotype mapping: an implicit embryogeny (Bentley and Kumar, 1999). The following section introduces the areas of natural and computational embryology. Section three describes evolvability and briefly summarises some work in the field. Section four details the implicit embryogeny-based system used and outlines a set of experiments. Sections five and six provide results and analysis, respectively. Conclusions are presented in section seven, with a brief section on further work.

2 EMBRYOLOGY AND EMBRYOGENY

Embryology is essentially the study of the formation and development of animal and plant embryos. It comprises three fundamental processes:

- *morphogenesis* — which involves the emergence and change of form (Bard, 1990).
- *pattern formation* — the generation of ordered spatial patterns of cell activities, through processes such as cellular differentiation (Wolpert, 1998).
- *cellular differentiation* — in which cells become specialised for particular functions (Wolpert, 1998).

These three processes operate together in different parts of the embryo at different times, in stages defined

by a 'recipe' known as an *embryogeny*. Embryogenies have evolved in nature to describe how an animal should be grown (epigenesis). This contrasts with the preformationist idea, where a complete organism was thought to be present from the earliest stages of development and simply increased in size (rev. Wolpert, 1998; Kumar & Bentley, 2000).

The distinction between genotype and phenotype in biology is a relatively recent one, even in comparison to the age of embryology as a discipline. The genotype-phenotype distinction was officially recognised in 1909 by the Danish Botanist Wilhelm Johannsen (Wolpert, 1998) and has been instrumental in helping to link the fields of genetics and embryology.

2.1 COMPUTATIONAL EMBRYOLOGY

Biology has clearly evolved designs of impressive complexity. This is due in part to the underlying representation (DNA) and the complex mapping from genotype-to-phenotype. Nature does not use a set of step-by-step, explicit instructions or encodings; instead instructions are implicitly encoded within the representation. Structure within a design can then emerge due to the complex dynamics of interaction between multiple implicitly encoded instructions. It therefore seems likely that one way of evolving complex solutions is to move away from step-by-step, explicit instructions and encodings, towards implicitly encoded instructions and representations that are specifications for the construction of complex phenotypes from relatively simple genotypes.

There are three types of computational embryogeny: external, explicit, and implicit (Bentley & Kumar, 1999). Most *external* embryogenies are hand-designed and are defined globally and externally to genotypes. They are characterised by fixed, non-evolvable structures specifying how phenotypes should be constructed using the genes in the genotype. Richard Dawkins' Blind Watchmaker program (Dawkins, 1987), used a simple *external* embryogeny to create biomorphs. Dawkins' program used the genetic operator mutation to vary biomorph designs. Dawkins assigned fitness values to the biomorphs himself, breeding morphologies to resemble various biological organisms.

An *explicit* embryogeny specifies each step of the growth process in the form of explicit instructions. In computer science explicit embryogenies can be viewed as a tree containing instructions at each node. Typically the genotype and the embryogeny are combined and both are allowed to evolve simultaneously. As an example consider Genetic Programming (GP) (Koza, 1992) which uses tree structures to represent its genotypes. GP therefore, offers a simple and concise way to evolve explicit embryogenies. There are a number of other notable examples of explicit embryogenies. Koza used an explicit embryogeny in the form of cellular encoding for the evolution of analogue circuits (Koza et al, 1999). Sims used an explicit embryogeny with the idea of directed graphs to specify the nervous systems (neural networks) and morphologies of virtual creatures (Sims, 1999).

In contrast, an *implicit* embryogeny does not explicitly specify each step of the growth process. Instead, rules are used to specify a dynamic and emergent process which results in a particular morphology (solution). De Garis describes an implicit embryogeny to evolve convex and non-convex shapes using a cellular automata approach along with one notion of cellular differentiation. He has reported encouraging results, as well as highlighting problems that need to be tackled in order to improve upon them (de Garis, 1999). Jakobi has evolved neural network driven robot controllers, using a biologically inspired encoding scheme – another example of an implicit embryogeny. His work makes use of an environment with diffusable morphogens and protein interactions (Jakobi, 1995). In a similar vein to this work, the focus of this paper is on the evolvability of a particular instance of an implicit embryogeny.

3 EVOLVABILITY

Evolvability is the capacity of a population to evolve and is an important concept in both biology and evolutionary computation (Marrow, 1999). An understanding of evolvability especially in EC would allow us to evolve solutions of greater complexity to problems (Marrow, 1999), and to create better, more evolvable representations for EAs (Bentley, 2000).

Much work has been done into evolvability, however as yet there is still no generally accepted measure. As Bedau points out, "...it is difficult to study evolvability, in part because of the difficulty in objectively and feasibly quantifying evolvability in a general enough way to compare it across different evolving systems", (Bedau, 1999).

Research has identified desirable properties in order to allow the evolution of evolvability. Glickman and Sycara compared mechanisms, operating at two different levels, for the evolvability of a population to itself evolve (Glickman & Sycara, 1999). The first was at the search operator level and involved encoding the per-bit mutation rate for each gene onto the genome – each gene had its own mutation rate, instead of having a global mutation rate. The second mechanism was at the representation level and involved looking at genetic

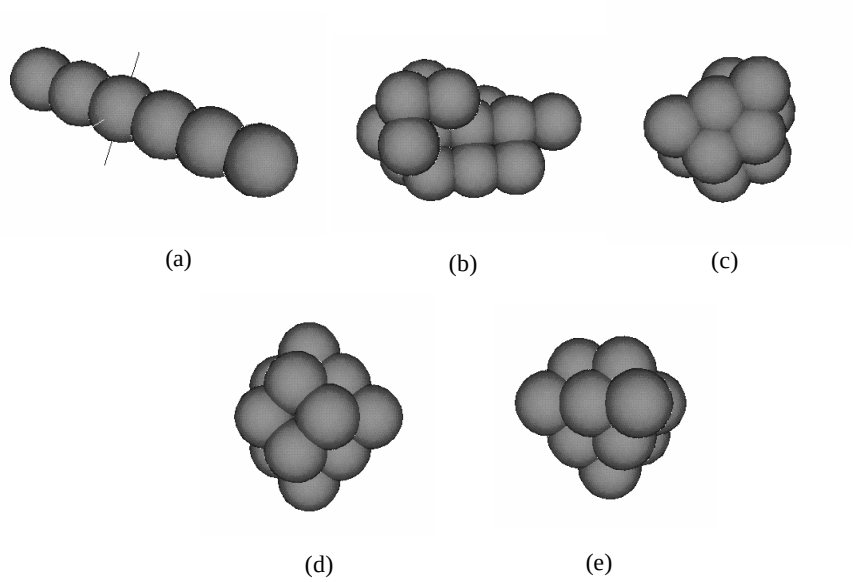


Figure 1. Best of run individuals for each threshold value (a) 0.25, (b) 0.75, (c) 1.5, (d) 2.25, (e) 3.0

programming. Analyses of the results revealed that the following properties were desirable in order to promote evolvability: a many-to-one mapping from genotype-to-phenotype, and non-elitist selection (Glickman & Sycara, 1999).

Through evolving morphologies under artificial selection, using his Blind-Watchmaker program, Dawkins (1989) has suggested that some lineages are more evolvable, and capable of generating more new forms than others. He attributes this to the use of inheritable replicators and in particular an embryology able to convert a simple genome into a relatively complex phenotype. In addition, a many-to-one genotype-to-phenotype mapping has been identified by numerous researchers as an important property for evolvability (Altenberg, 1995; Glickman & Sycara, 1999; Turney, 1999; Wagner, 1999).

4 SYSTEM & EXPERIMENTS

This section describes both the current implicit embryogeny system and two sets of experiments used to investigate the evolvability of a relatively simple instance of an implicit embryogeny.

Phenotypes were displayed in an *isospatial grid,* which uses isospatial co-ordinates as opposed to standard cartesian co-ordinates. It was developed by Frazer (1995) who saw the cartesian co-ordinate system as containing strong biases towards linear shapes, and as he points out nature does not exhibit linear shapes. A point in isospatial space is termed a *mote,* and is defined by six axes yielding 12 directions. Although no system is free of biases, the isospatial system removes orthogonal biases, thus allowing for the generation of more organic morphologies.

4.1 THE IMPLICIT SYSTEM

Within the isospatial grid cells are able to divide and proliferate according to the number of cell divisions.

The system employed in this work used an implicit embryogeny-based representation. Phenotypes are grown using a set of rules. The chromosome length was fixed and consisted of 12 genes/rules in total. Every rule/gene comprised a precondition section and an action section. The precondition section of a gene/rule comprised 24 bits, see figure 1. These 24 bits were grouped into pairs, corresponding to directions in the isospatial grid. Consequently, this grouping of the 24 precondition bits into pairs yielded 12 directions within the grid. The first bit of a pair in the precondition part of gene/rule represents a ‘don't care’ wildcard (depicted in figure 1 as a ‘#’), and the second bit represents the value part (depicted in figure 1 as a ‘V’) for that particular direction. If a value bit is set to 1 this means that in order for the action part of the rule to fire a cell must be present in that particular direction, and if set to 0, no cell should be present in that particular direction. If on the other hand, the don't care bit is set to 1, the system ignores the value part of the pair - meaning the rule does not depend on whether the cell is present or not. However, if the don't care bit is set to 0 then the value bit is taken into account.

The second section of a gene/rule is the action section. This section consists of 4 bits that are decoded to yield a number between 0 and 11, thus giving 12 distinct numbers corresponding to growth in one of 12 distinct directions, as defined by the isospatial grid.

In order for a gene/rule to fire (or be *expressed)* a system of activation was adopted in which every time a precondition was met, the gene/rule's activation energy

was increased by 0.25. Once a gene/rule's activation energy exceeded the specified threshold amount the gene/rule would fire and the action section of the gene/rule decoded and expressed.

0	1	2	3	4	5	6	7	8	9	10	11
#	V	#	V	#	V	#	V	#	V	#	V	#	V	#	V	#	V	#	V	#	V	#	V
1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	0	1	0	1	1	1	0	0	1

Figure 2. The structure of the precondition section of a gene/rule. A '#' denotes a 'don't care' case, and a 'V' denotes the value part of each precondition pair for a particular direction; direction are denoted by the numbers at the top.

4.2 EXPERIMENTS

The evolution of a sphere was the application selected to investigate the evolvability of the implicit embryogeny-based representation. Two sets of experiments were conducted. Both experiments employed a simple generational genetic algorithm (Goldberg, 1989) without elitism (Glickman & Sycara, 1999).

4.2.1 Experiment 1

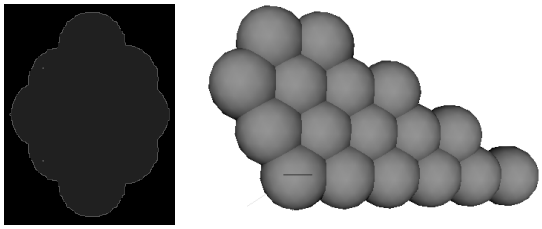
This set of experiments entailed evolving a sphere using 12 genes that encoded rules for the growth of a shape in an isospatial grid.

The following GA parameter settings were used for the first set of experiments: a total of 100 runs were performed with a population size of 100 individuals for 100 generations and a mutation rate, per-bit, of 0.03. Three divisions were allowed, i.e., the initial seed-cell (zygote) was allowed to divide a maximum of three times. On each division each daughter cell inherits the parents division counter minus one. Rules fired when the activation exceeded a threshold value of 0.75 (a threshold of 0.75 means that 9 precondition matches are required), and a total of 12 randomly created rules were used in-order to grow the designs from a single zygote cell. The measure of fitness used was based on the following equation for the radius of a sphere:

$$X^2 + Y^2 + Z^2 = R^2$$

Given this equation, it is possible to determine whether a cell has been placed inside or outside of the desired target shape. For example, if the sum of the cell's X, Y, and Z co-ordinate values squared, are greater than R2, then the cell is out of the desired shape. If less than R2, then the cell is inside the shape, and if equal to R2 the cell is on the boundary itself. Fitness thus became a minimisation of the following function:

fitness = (1 / #cells_inside_shape) + (#cells_outside_shape / 20)



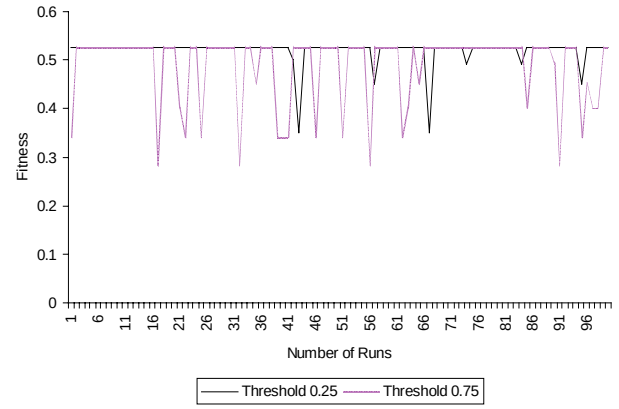
(a) (b)

Figure 3. The system is able to evolve good solutions (a) as well as some rather poor solutions (b). The parameter settings were 5 divisions and threshold values of 3.0 for (a) & 0.25 for (b).

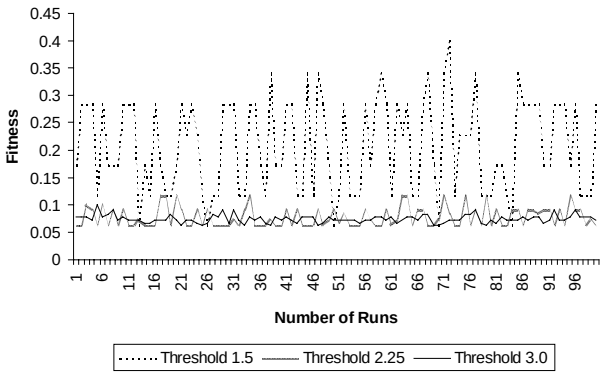
4.2.2 Experiment 2

This set of experiments were carried out to examine the evolvability of the representation. This was done by creating a genome, at random, which was then subjected to 1000 single-point mutations. This was repeated a total of four times starting from different randomly sampled areas of the search-space. The difference in cell number and cell position of the phenotype for each point mutation was then recorded.

show how varying the rule-firing threshold affected fitness when evolving spheres.

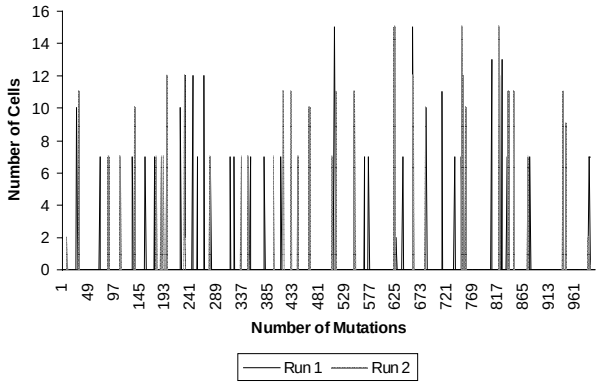


(a)

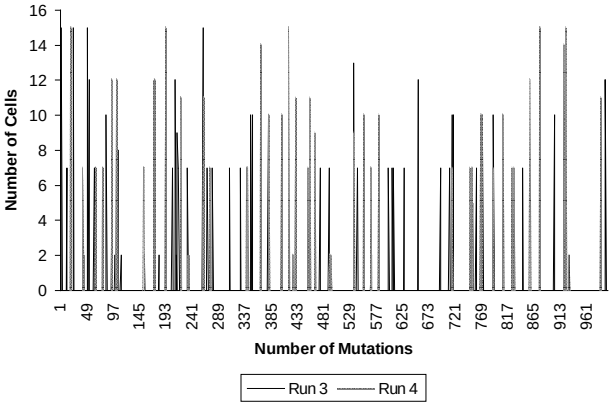


(b)

Figure 4. Graphs of Fitness versus Number of Runs for all five Thresholds. As the Threshold for each rule to fire is increased fitness gets better.



(a)



(b)

Figure 5. Graphs showing difference in phenotype of during a random walk of length 1000, threshold used was 0.25.

5 RESULTS

This section is split into two and provides the results for both experiments.

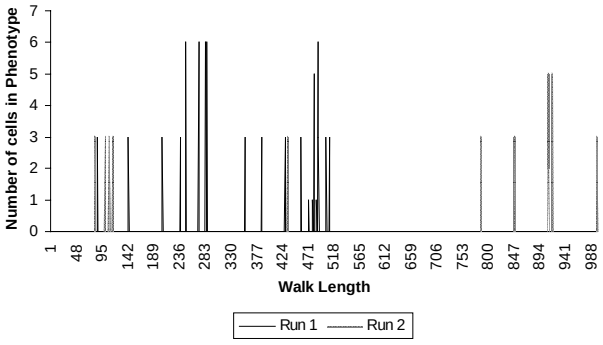
5.1 RESULTS FOR EXPERIMENT 1

Figure 2 shows some of the best individuals evolved after 100 generations for each threshold. They show how fitness improves (morphologies become increasingly more spherical) as the threshold is increased, yet despite the improvements the morphology shown in figure 2e represents the best the system with a threshold of 3.0 and a cell division of 3 can do. The best fitness attained was 0.0625 for threshold values of 2.25 and 3.0 as shown in figures 2d and 2e.

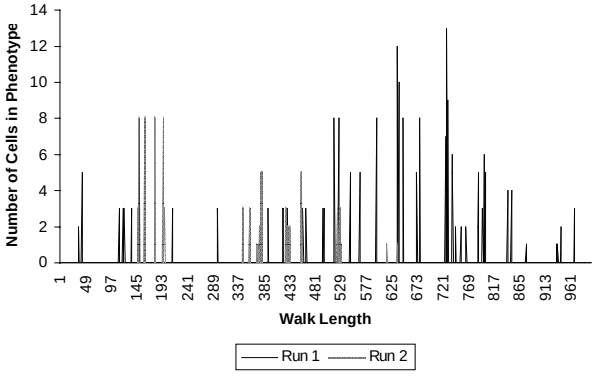
Experiments varying the number of divisions have been performed achieving much better fitness results, such as, 0.027 (figure 3a), for runs using the following parameter settings: threshold 3.0, divisions 5, population size 500, over 100 generations. It was noted however, that an increased number of divisions slowed execution time down. Decreasing the value of the threshold from 3.0 to 0.25 with the number of divisions set to 5, causes the system to produce dramatically worse results, for example, 0.75 (figure 3b). A typical run with these settings had initial fitness values as high as 3.75 and lasted in excess of 25 minutes using a 400MHz Intel Pentium PC.

Figure 4 shows how end of run fitness values get better, i.e., fitness values decrease, as the threshold is increased. Figure 4a shows the results for thresholds 0.25 and 0.75. It can be seen how a threshold of 0.25 is effectively a flat line occasionally plummeting to give a better fitness of 0.35. Increasing the threshold to 0.75 gives only moderately better results, yielding a best fitness of 0.28. Figure 4b shows the results for three

threshold values, namely, 1.5, 2.25, and 3.0. The graph shows how a threshold value of 1.5 gives better fitness values more often than thresholds 0.25, and 0.75. However, the best results were obtained using threshold values of 2.25, and 3.0.



(a)



(b)

Figure 6. Random walk graphs for four separate runs sampled at random with a threshold of 3.0.

5.2 RESULTS FOR EXPERIMENT 2

Figures 5 and 6 show the results for the second set of experiments. They show the number of changes in the phenotype, both cell number and cell position, during a random walk of length 1000. The ruggedness (number of large changes of cell position and quantity in the phenotype) of figure 5a & 5b show just how discontinuous the solution space is given a threshold of 0.25. Many single-point mutations can be made to the genotype, often in excess of 50 before there is any change in the phenotype.

Figures 6a and 6b show the same graph as for figure 5, except with a threshold value of 3.0. As is clear, figures 6a and 6b are much less rugged than figures 5a and 5b. Figure 6b for example, shows that for run 2 after 148 mutations, cell differences in the phenotype change suddenly to 3. Two further mutations and the change is somewhat more dramatic – a change of 8 cells. A further 10 mutations later (160 mutations altogether at this point), and there is yet another large change of 8 cells.

6 ANALYSIS

When the results for the first set of experiments are analysed it becomes clear that the system is very sensitive to changes in the threshold and division parameters.

As the threshold required to activate a gene/rule is increased the fitness gets better. This indicates that good fitness is dependent on stricter precondition requirements for gene activation (expression).

The reason for this behaviour is that the system must evolve specific rules to promote and control growth. Lower threshold values trigger growth with only few precondition matches, resulting in excessive growth and bad fitness values. Indeed, other experiments have shown a threshold value of zero gives excessive and uncontrolled growth. In contrast, higher thresholds (much stricter precondition matches) get better fitness values as evolution is able to make use of a greater number of more specific rules in order to control growth.

The second set of experiments, the random-walk experiments, show how dissimilar phenotypes are placed close together in solution space, making it difficult to evolve solutions. The graphs in figures 5a & 5b show a very rugged landscape reflecting numerous discontinuities in solution space for a threshold of 0.25. This is consistent with the previous observation that by reducing the value of the threshold parameter the system is more inclined to cause growth.

In contrast, a threshold value of 3.0 as shown in figures 6a and 6b provide a somewhat less rugged landscape, however, not smooth enough to allow progressive evolution. For example, figure 6a, run 1 shows that within a walk limit of 1000, after 511 single-point mutations no further progress is seen, i.e., 489 further mutations resulted in no change. Small changes in the genotype do not correspond to small changes in phenotype; in fact they correspond to large changes in phenotype or no change at all. (As figure 5 shows, this lack of potential evolvability is even worse for the lower threshold.)

The results also show periods of no change (*stasis*) during the course of a random walk. These periods of stasis correspond essentially to different genotypes yielding the same phenotype, due to a small degree of

redundancy in the precondition part of the genome (the 'don't care' bits). There is therefore, a many-to-one mapping from genotype to phenotype. Further examination of identical phenotypes taken from experiment 2 and their corresponding genotypes (which were not identical) confirms that the mapping does indeed possess a many-to-one relationship. Recent literature (e.g., Shipman, 2000) indicates that such relationships may be indicative of neutral networks and hence may increase evolvability. However, irrespective of this property both experiments showed that the representation is not very evolvable. As is apparent from this research, this may be attributed to the fact that dissimilar phenotypes are placed too close together in solution space – a result that is visible in the graphs of figures 5 and 6, showing how periods of stasis are punctuated with greatly dissimilar phenotypes from their neighbours in solution space, making it difficult for gradual evolution to occur.

The implicit embryogeny based representation used in this work has the desirable many-to-one genotype-to-phenotype mapping as advocated in the evolvability literature (Glickman & Sycara, 1999; Turney, 1999; Wagner, 1999; Bedau, 1999; & Altenberg, 1995). Despite having this desirable many-to-one genotype-to-phenotype relationship, the system still does not perform as well as desired. This implies that a many-to-one genotype-to-phenotype mapping, on its own, is not enough to ensure evolvability.

7 CONCLUSIONS

This paper has looked at the evolvability of an implicit embryogeny based representation. The particular instance of an implicit embryogeny used in this work is not as evolvable as one would desire for evolution.

This work has shown that implicit embryogeny-based representations need to be designed with care. The work hints of attributes for a better, new representation:

1. genotypic redundancy to cause many-to-one relationships from genotype to phenotype
2. similar solutions should be placed close together in solution space to allow gradual evolution, rather than having to rely on excessive mutation rates in an attempt to jump over the discontinuities of poor representations.

Further Work

Further work is in progress to develop a new representation more amenable to evolution and benefiting from the research into evolvability.

Acknowledgements

Thanks to Tom Quick and Ian Oszvald for helpful suggestions and criticism. This work is funded by Science Applications International Corporation (SAIC).

References

Altenberg, L. (1995). Genome Growth and the Evolution of the Genotype-Phenotype Map. In *Evolution and Biocomputation: Computational Models of Evolution.* Springer-Verlag, pp. 205-259.

Bard, J. (1990). *Morphogenesis: The cellular and molecular processes of developmental anatomy.* Cambridge University Press, UK.

Bedau, M. A. (1999). Quantifying the Extent and Intensity of Adaptive Evolution. *Proceedings of the 1999 Genetic & Evolutionary Computation Conference Workshop Program*. Orlando, Florida, USA, July 13, 1999.

Bentley, P. J. (2000). Explorative Evolution: Component Based Representations. Plymouth, UK.

Bentley, P. J. (Ed). (1999). *Evolutionary Design by Computers.* Morgan Kaufmann Pub.

Bentley, P. J. & Kumar, S. (1999). Three Ways to Grow Designs: A Comparison of Embryogenies for an Evolutionary Design Problem. *Genetic & Evolutionary Computation Conference*.

Coates, P., (1997) Using Genetic Programming and L-Systems to explore 3D design worlds. *CAAD Futures'97*, R. Junge (Ed), Kluwer Academic Publishers, Munich.

Dawkins, R. (1987). The Evolution of Evolvability. *Artificial Life*. Langton (Ed.) USA.

de Garis, H. (1999) Artificial Embryology and Cellular Differentiation. Ch. 12 in Bentley, P. J. (Ed.) *Evolutionary Design by Computers*. Morgan Kaufman Pub.

Frazer, J. (1995). *An Evolutionary Architecture.* Themes VII, Architectural Association, London, UK.

Goldberg, D. E. (1989). *Genetic Algorithms for Search, Optimization , and Engineering.*

Glickman, M.. & Sycara, K. (1999). Comparing Mechanisms for Evolving Evolvability. *Proceedings of the 1999 Genetic & Evolutionary Computation Conference Workshop Program.* Orlando, Florida, USA, July 13, 1999.

Holland, J. H. (1975). *Adaptation in Artificial and Natural Systems*.

Jakobi, N. (1996) Harnessing Morphogenesis. University of Sussex, Cognitive Science Research Report #429, Brighton, UK.

Koza, John R. (1992). *Genetic Programming I*: On the Means of Natural Selection. San Francisco, CA: Morgan Kaufmann.

Kumar, S. & Bentley, P. J. (1999) The ABCs of Evolutionary Design: Investigating the Evolvability of Embryogenies for Morphogenesis. Genetic & Evolutionary Computation Conference, (GECCO) Orlando, Florida, USA.

Kumar, S. and Bentley, P. J. (2000). "Computational Embryology: Past, Present and Future." To be published as an invited chapter in Ghosh and Tsutsui (Eds) Theory and Application of Evolutionary Computation: Recent Trends. Springer-Verlag (UK).

Marrow, P. (1999). Evolvability: Evolvability, Computation, biology. *Proceedings of the 1999 Genetic & Evolutionary Computation Conference Workshop Program*. Orlando, Florida, USA, July 13, 1999.

Shipman, R. Shackleton, M. et al. (2000). Neutral search spaces for artificial evolution: a lesson from life.

Sims, K. (1999). Evolving three-dimensional Morphology and Behaviour. Ch. 13 in Bentley, P. J. (Ed.) *Evolutionary Design by Computers.* Morgan Kaufman Pub.

Turney, P. D. (1999). Increasing Evolvability Considered as a Large-Scale Trend in Evolution. *Proceedings of the 1999 Genetic & Evolutionary Computation Conference Workshop Program.* Orlando, Florida, USA, July 13, 1999.

Wagner, G. (1999). The Quantitative Genetic Theory of Evolvability. *Proceedings of the 1999 Genetic & Evolutionary Computation Conference Workshop Program*. Orlando, Florida, USA, July 13, 1999.

Wolpert, L. (1998). *Principles of Development.* Oxford University Press, Oxford, UK.