The ABCs of Evolutionary Design: Investigating the Evolvability of Embryogenies for Morphogenesis

Sanjeev Kumar

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

S.Kumar@cs.ucl.ac.uk
+44 (0) 171 419 3694

Peter Bentley

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

P.Bentley@cs.ucl.ac.uk
+44 (0) 171 391 1329

Abstract

Previous work investigated three types of embryogeny: External, Explicit, and Implicit. This paper focuses on the two most interesting embryogenies: explicit and implicit, investigating the evolvability and scalability of both embryogenies for morphogenesis. The problem set is that of evolving certain predefined shapes - letters of the alphabet. The results show that both embryogenies are good at defining different morphologies, but significantly, the implicit embryogeny incurs no increase in genotype size as the problem is scaled.

1 INTRODUCTION

Evolutionary computation (EC) has been a very successful area of computer science for some time. EC has grown from several types of evolutionary algorithms (EAs) which take their inspiration from nature: Genetic Algorithms, Genetic Programming, Evolutionary Strategies and Evolutionary Programming (Bentley, 1999). The EAs differ from each other in various ways, for example, in the use of differing genetic operators and underlying representations.

An important difference is the distinction between the genotype (coded parameters and values) and the phenotype (representation of solutions). GP practitioners often regard the genotype as the phenotype, as do ES and EP practitioners. The genetic algorithm (GA) is the only one of the four EAs that makes the distinction. It is the omission of this crucial genotype to phenotype mapping process, known in biology as an *embryogeny*, which denies the non-GA practitioner the advantages that embryogenies bring.

This paper investigates the evolvability of two of the most interesting types of computational embryogeny, *explicit*, and *implicit*. The paper is organised as follows: section two gives an extended background, comprising reviews on both biology and computer science. Section three introduces the two embryogenies, section four describes a series of experiments followed by section five which gives an analysis of the results. Section six provides conclusions and the paper ends with a synopsis depicting further work.

2 BACKGROUND

This section gives a background on the related biology and computer science for the research described in this paper. The sub-section relating to biology introduces the important concepts upon which this paper is based, namely: embryogenies and morphogenesis. The sub-section on computer science briefly introduces the reader to the types of computational translation of the biology in the literature.

2.1 BIOLOGICAL BACKGROUND

The way in which embryos grow to become babies and then further, into adults, has been the subject of great debate since as early as Aristotelian times. It was Aristotle who first suggested that animal embryos actually grew (*epigenesis*), rather than being pre-formed (*preformationism*).

Biology has moved on considerably since Aristotle, and an entire new field now known as Developmental Biology has emerged. In particular, we now know that Aristotle's notion of epigenesis is much closer to the

truth. Developmental Biology and in particular, Embryology has fast become an exciting and fashionable subject for research.

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

- *morphogenesis* - which involves the emergence and change of form (Bard, 1990).
- *regional specification* - in which compartmentalisation of the embryo into specific regions occurs (Slack, 1991).
- *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, and in stages according to a 'recipe' known as an *embryogeny*. Embryogenies have evolved in nature to describe how an animal should be grown, rather than contain an overall description of an animal.

Embryology like other subjects has its own set of problems that need answering. One such problem is that of positional information i.e. how cells 'know' where to grow? This was addressed by the eminent embryologist Lewis Wolpert who put forth *positional information theory*.

Positional information theory states that cells glean their positional information from the free diffusion of a chemical, known as a *morphogen*, relative to a boundary. The information is coded in the form of the concentration value of the diffusing morphogen. This diffusion sets up a *chemical gradient*, thus allowing cells to position themselves relative to the boundary whereupon they can, if need be *differentiate*, i.e., become a specialised type of cell (Wolpert, 1998). This idea is used in the implicit computational embryogeny described later.

2.2 COMPUTER SCIENCE BACKGROUND

Turing first advocated the use of Morphogenesis for computer science as early as 1952. Since then morphogenesis has featured in a number of works, such as in the evolution of neural network morphologies (Jakobi, 1995), and evolvable hardware (Koza et al. 1998).

However, to date computer science has used crude approximations of the natural embryological processes, paying little attention to the intricate subtleties. For example, the use of morphogenesis for neural network topology design (Jakobi, 1995).

Regional specification, which plays an important part in the early stages of embryogenesis has to the authors knowledge not been investigated, despite its obvious advantages for the creation of form, such as compartmentalisation (which can be viewed as reducing the problem down to smaller units).

Previous work (Bentley & Kumar, 1999), categorized the notion of embryogenies into three different types. To summarise, there are three types of computational embryogeny, namely: external, explicit, and implicit.

Most *external* embryogenies are hand-designed and are defined globally and externally to genotypes. For example, Evolutionary Art systems often use embryogenies defined by fixed, non-evolvable structures which specify how phenotypes should be constructed using the genes in the genotypes (Bentley, 1999). Similarly, Richard Dawkins' Blind Watchmaker program (Dawkins, 1987), used a simple external embryogeny, to create biomorphs, using the eye-of-the-beholder as a fitness function, and the genetic operator mutation along with selection. Dawkins suggests a simple 1-to-1 mapping as an embryogeny and concludes that this sort of embryogeny is a 'naive' way of achieving the aims. He proposes improvements via constraints and thus introduces the notion of a constrained embryogeny. Dawkins (Dawkins, 1987) also points out advantages of using embryogenies, why they are so important, and perhaps most interestingly, that different embryogenies can lead to different results. For example, different embryogenies allow different areas of 'solution space' to be searched and thus in doing so constrain themselves to different types of shapes or designs (not necessarily a bad thing as Dawkins shows, relative to what he calls naïve pixel-peppering).

An *explicit* embryogeny specifies each step of the growth process in the form of explicit instructions. In computer science, an explicit embryogeny can be viewed as a tree containing a single growth instruction at each node. Genetic Programming (GP) uses tree structures to represent its genotypes. GP therefore, offers a simple and concise way to evolve explicit embryogenies. Typically, the genotype and the embryogeny are combined and both are allowed to evolve simultaneously. Examples of explicit embryogenies include Coates (1997) who uses Lindenmayer systems to evolve architectural form. Koza et al (1999) use an explicit embryogeny in the form of cellular encoding for the evolution of analogue circuits. Sims (1999), uses an explicit embryogeny with the idea of directed graphs to specify the nervous systems (neural networks), and morphologies of virtual creatures.

An *implicit* embryogeny does not explicitly specify each step of the growth process. Instead, the growth process is implicitly specified by a set of rules or instructions, similar to a 'recipe', which govern the growth of a shape. For example, de Garis (1999) describes an implicit embryogeny to evolve convex and concave shapes using a cellular automata approach along with the 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).

3 EVOLVING EMBRYOGENIES

Previous work (Bentley & Kumar, 1999), compared the performance and scalability of different evolved computational embryogenies for the representation of tessellating tiles. Subsequent experiments have shown that significant questions remain concerning the evolvability of different embryogenies. Specifically, evolved implicit and explicit embryogenies show inconsistent abilities to define specific morphologies. Consequently, it was decided that further investigation was necessary to help explore and understand these issues of evolvability.

3.1 EVOLVING PREDEFINED SHAPES

In order to assess the change in performance of the two embryogenies, a number of fixed shapes were specified as targets for evolution. Since shapes with distinct and useful characteristics were desired (e.g. convex, concave, solid, hollow, curved and linear), a subset of the alphabet was selected. Six letters were chosen: C, E, G, L, O, and R, as shown in figure 1. (The fact that these letters also allow us to spell the word 'GECCO' is purely incidental.)

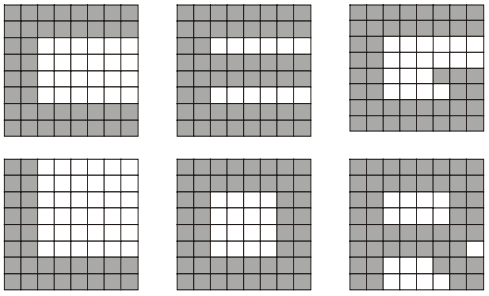

Figure 1: The pre-defined six target shapes.

These 6 letters were selected based upon how much of the alphabet they were representative of. For example, the diagonal in the letter R forming the bottom right portion of the letter is characteristic of the letters M, N, W, X, Y, Z. Likewise, the semi-circle is characteristic of the letter P,R,B. The Letters E and L with their up-right stem are characteristic of P,T,D,F.

To judge how closely each evolving shape matched the targets, a fitness function based on the number of incorrectly filled squares was employed. The fitness score is incremented by one whenever an element in the evolving shape differs from the corresponding element in the current target, see figure 2. To assess scalability, three different phenotype grid sizes of 4x4, 8x8 and 16x16 cells were used.

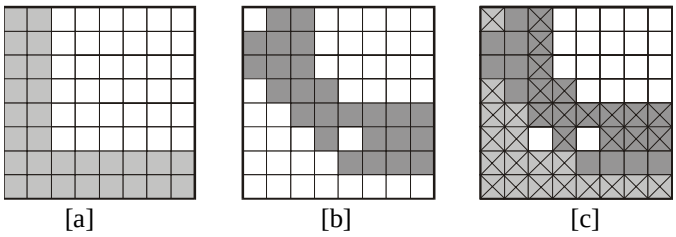

Figure 2: Calculating the fitness of an 8x8 evolving shape. [a] shows the target, [b] shows the shape to be judged, [c] shows the incorrect elements identified by the fitness function.

3.2 EXPLICIT

This first system used genetic programming (GP) (Koza, 1992) to evolve explicit embryogenies in the form of program trees. Beginning at a seed or zygote cell placed in the phenotype grid, the embryogeny defines the direction of growth at every point. Four functions were used: LEFT, RIGHT, UP and DOWN, with each node in the tree allowed up to four branches. Paths of growth were permitted to overlap. Figure 3 shows an example genotype defining the explicit embryogeny. The root node has two parts: *x* and *y* for the co-ordinates of the seed.

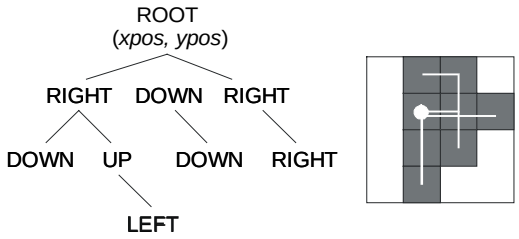

Figure 3: An example explicit embryogeny defined by a tree of nine nodes, and its corresponding 4x4 phenotype.

The GP system used steady-state selection and a crossover designed to minimise disruption by crossing parents at points of similarity in the two trees. Further details of this system and crossover operator can be found in (Mallinson & Bentley, 1999 and Bentley & Wakefield, 1996). As with all GP systems, bloat occurred, so an additional fitness function penalised genotypes with more nodes. This system differed from the one presented in (Bentley & Kumar, 1999), in that it evolved the co-ordinates of the seed.

3.3 IMPLICIT

The second system was an advanced variable length chromosome GA that evolved implicit embryogenies. Each genotype comprised a variable number of rules (usually between four and eight). Each rule had a precondition and an action. Each precondition had six fields: *LEFT, RIGHT, UP, DOWN, X, Y*. A specific rule can take the following values for each precondition field (where # is *don't care*, 0 is *empty*, 1 is *filled*, **0**,**1**,**2**,**3,4,5,6,7** are gradient zones):

LEFT	RIGHT	UP	DOWN	X	Y
0,1,#	0,1,#	0,1,#	0,1,#	**0-7**,#	**0-7**,#

For a rule to be fired, values in at least four of the six fields in the precondition must be matched. (This provides the equivalent of disjunction for rule preconditions.) The action of a rule can be: DIE, UPDATE, or grow LEFT, RIGHT, UP or DOWN.

Growth takes place in a phenotype grid, which as usual can be 4x4, 8x8 or 16x16 elements. In order to permit evolution of specialised rules that can provide detail in specific areas of the phenotype, the grid has two 'gradients' - one in the x direction, one in the y direction. In a similar way to the gradients used to provide positional information in eggs and wombs of nature (Wolpert, 1998, Slack, 1991, Lawrence, 1995), the gradients divide the grid into 8 zones per axis (as opposed to 4 in previous work), regardless of the number of elements in the phenotype grid.

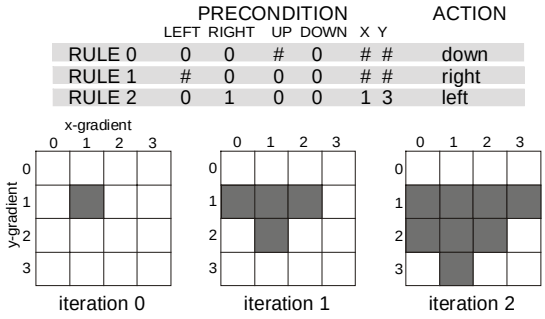

Figure 4: Example of a three-rule implicit embryogeny and its corresponding phenotype after two iterations.

At iteration zero, a seed cell is placed in the phenotype grid. To model biological cell growth, the rules are then applied for a fixed number of iterations to each *filled* element in the current embryonic phenotype grid. (This is unlike traditional cellular automata, where rules are applied to empty or filled grid elements.) Depending on whether the neighbouring elements of the current element exist or not, and on the strength of the two gradients at that point, the rules may be activated, causing growth or cell death in the phenotype. Rules are applied 'in parallel' so that the results of applying the rules to each filled element only take effect at the end of each iteration step. However, a rule which performs the UPDATE action causes all activated rules in the current iteration to be applied. By prematurely placing cells in the phenotype grid in this way, evolution can increase the number of rules applied in each iteration and provide extra growth where needed. This new type of rule action was added to the embryogeny because during the development of the system, the number of iterations was found to be overly critical. Finally, the system was also given the ability to evolve the seed co-ordinates. Figure 4 shows the growth of a target shape, defined by three rules.

4 EXPERIMENTS

4.1 OBJECTIVES AND PARAMETERS

The experimental objectives were three-fold: firstly to investigate the use of both embryogenies for efficiency of search in terms of fitness. Secondly, to investigate the scalability of both embryogenies for evolving different morphologies, and finally, to see how the evolution of the two embryogenies differs in defining different morphologies.

A total of 50 runs were performed with each target shape (letters C, E, G, L, O and R) for each grid size. Population sizes of 100 and a total of 100 generations were used for each run. The explicit embryogeny system used an initial tree depth of 4 for the 4x4, 5 for the 8x8 and 6 for the 16x16 grids. All trees were created randomly.

The implicit system used random rule initialisation for both the initial population and for each new rule evolved[1]. The variant on the cellular automata presented in this work used iteration values of 4 for the 4x4, 8 for the 8x8, and 14 for the 16x16 grids. Both systems used random crossover for offspring creation. The explicit system employed a mutation probability rate of 0.001 per bit, whereas preliminary experiments revealed that the implicit system required an increased bit mutation rate of 0.05, along with a rule mutate rate of 0.5.

4.2 RESULTS

A summary of the results from the experiments is given in Table 1[2]. As shown in the table, both embryogenies attained good fitnesses for all 4x4 target letters. However, relative performances between the two approaches were inconsistent. For example, the explicit embryogeny outperformed the implicit for the letters C and L, whilst the reverse was true for the other targets.

[1] This is as opposed to copying an existing rule as in previous work (Bentley & Kumar 1999).

[2] Because of time constraints, average values given for the 16x16 targets using the implicit embryogeny were based on only 10 experiments per target.

Table 1: Results for the target shapes. Values in *italics* denote the results for the implicit embryogeny. Solution sizes are measured in tree nodes for the explicit, and rules for the implicit embryogeny.

	4x4		8x8		16x16	
Shape	**Mean Soln. Size**	**Mean Fitness**	**Mean Soln. Size**	**Mean Fitness**	**Mean Soln. Size**	**Mean Fitness**
C	14.28 *12.70*	0.92 *1.64*	57.70 *11.47*	13.20 *12.82*	309.40 *10.00*	84.1 *53.7*
E	24.22 *11.42*	1.28 *0.32*	168.44 *11.96*	9.54 *5.89*	693.58 *6.700*	81.40 *49.40*
G	18.88 *13.86*	1.2 *0.78*	59.52 *10.98*	12.72 *12.84*	302.12 *6.000*	76.34 *52.90*
L	9.52 *11.22*	0.26 *0.58*	71.04 *9.02*	3.56 *6.38*	235.46 *8.200*	39.46 *38.40*
O	20.20 *11.60*	1.31 *0.18*	81.29 *13.29*	15.76 *9.00*	293.00 *6.400*	104.33 *48.70*
R	18.78 *12.98*	1.35 *0.60*	121.53 *12.33*	7.88 *9.92*	513.43 *6.900*	76.16 *55.80*

For the 8x8 grid, fitness scores were reduced, on average for both methods. For example, the explicit embryogeny managed 3.55 at best for the letter ‘L’ and at worst 15.76 for the ‘O’. The implicit faired similarly on the 8x8 targets achieving a fitness of 5.89 for the 'E' and only 12.84 for the 'G'. Again, relative performances varied, this time with each embryogeny providing better scores for 3 of the letters.

When the problem was scaled up to the 16x16 grid, the results show that the implicit embryogeny outperformed the explicit, in terms of fitness. The figures are, however, a little deceptive. For these targets, many of the shapes evolved by the implicit embryogeny were solid blocks. Because of the simple nature of the fitness function, such shapes were awarded higher fitness scores compared to the attempts of the explicit. (Nevertheless, it should be noted that the forms generated by the explicit rarely resembled the desired targets, either.)

Execution times were noticeably different for the two techniques. As the scale of the problem was increased, both methods took longer to grow shapes, but of the two, the implicit required the most computation time. For example, evolution time of six hours for one run of the implicit was not uncommon, compared to less than an hour for the explicit.

Perhaps the most significant results shown in Table 1 are the solution sizes. It is clear that the explicit embryogeny required ever-increasing tree sizes as the scale of the target shapes were increased. However, the reverse seems to be true for the implicit embryogeny, where the number of rules actually appears to decrease as the problems are scaled up. This lack of increase of solution size corroborates and confirms the results obtained in previous work which reported similar findings (Bentley & Kumar, 1999).

4.3 ANALYSIS

The results show interesting behaviours of both embryogenies for all grid sizes. For the 4x4 grid, because of the size of the targets, the ability of both methods to find good solutions is not surprising. The explicit embryogeny uses small trees to define its solutions, but the implicit often seems to evolve more rules than are necessary. More specifically, a larger number of rules are evolved than are actually used during the growth process. The reason for the inefficiency for such small targets seems to be to do with the search process - it is very hard for evolution to find the correct rules for specific shapes. Clearly the 'add rule' mutation plays an important, but excessive role for these smaller problems. Rules are added until an appropriate collection exists to define the target, but the unused rules are not removed by mutation, much like bloat in GP.

The same searching mechanism is also evident for larger grid sizes with the implicit embryogeny, with similar levels of redundancy observed. However, because the number of rules did not increase much beyond 12, such redundancy becomes a more acceptable compromise for larger problems.

So how does evolution fine-tune solutions to make them match the target letters? Both types of embryogeny seem to begin by filling a large part (or all) of the phenotype grid, and then 'pruning away' unnecessary elements, see figures 5 and 6. The explicit embryogeny achieves this by pruning branches of its trees; the implicit embryogeny makes use of 'kill' rules to remove elements. Of the two, the implicit embryogeny goes to the furthest

extreme with this technique - often by evolving a completely solid block and then picking out the odd element, see figure 6.

The fitness function may be to blame for the 'carving letters from a solid block' approach - for it awards considerably higher fitnesses for solid shapes than for emptier ones.

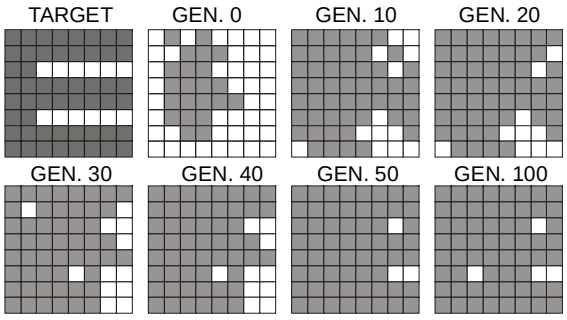

Figure 5: The evolution of an 'E' using the explicit embryogeny. The best new shapes grown in the population are shown every 10 generations (except where no change occurred). The final shape has a fitness of 8 and a soln. size of 251 nodes.

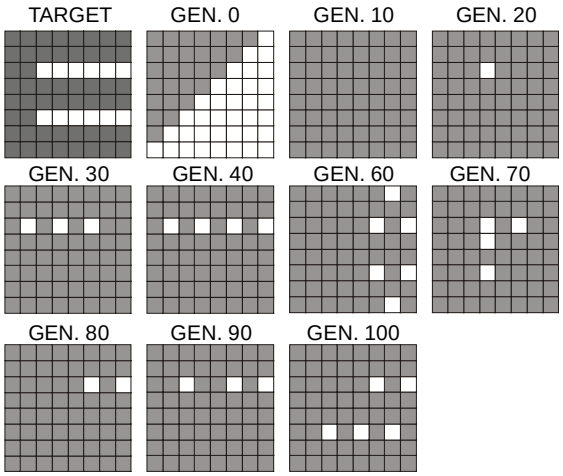

Figure 6: The evolution of an 'E' using the implicit embryogeny. The best new shapes grown in the population are shown every 10 generations (except where no change occurred). The final shape has a fitness of 7 and a solution size of 11 rules.

As Dawkins (1987) points out, certain embryogenies are better than others at producing certain morphologies. This observation seems to be echoed in this work too. The explicit embryogeny found morphologies such as C's and O's difficult, whereas the implicit was able to handle these morphologies with relative ease. This can be attributed to the fact that the implicit need only generate a few general growth rules for specific directions, and in doing so, can start from a single seed and grow to encompass all four sides of the grid whilst leaving (or killing), for example in the case of the letter 'O', a hole in the middle. This is difficult for the explicit embryogeny as it must evolve a long and difficult growth path around the edge of the grid, whilst keeping the centre of the shape free of elements.

The way in which evolution attempted to generate morphology indicates two points: firstly that the implicit embryogeny seems to have considerable potential because of its impressive scalability, perhaps more so than the explicit embryogeny. Secondly, the representation used for the implicit embryogeny is not as amenable to evolution as we would desire. This may be caused by:

- the representation, which allows dissimilar phenotypes to be close together in the solution-space, providing a discontinuous search-space of solutions and thus causing problems for evolution.
- ineffective positional rules. Although the notion of 'zones' improved the quality of solution for the implicit more than without, it is clear that positional information is hard to glean using the current implicit system, making the evolution of specific rules difficult, leading in turn to bad fitness results.

5 CONCLUSIONS

This paper has looked at the evolvability and scalability of two different embryogenies (explicit and implicit) for the problem of evolving shape morphologies.

The behavioural analysis of these two embryogenies has shown that both are good at growing different shape morphologies, and that evolutionary computation can benefit from the use of embryogenies.

In addition, this work highlights some of the problems that require attention with regard to designing evolvable embryogenies. For example, biological concepts such as positional-information, as echoed in this work in the form of zones, can assist in the evolution of shape morphologies, but need to be very carefully designed.

Nature has been successfully evolving complex animals for millions of years. It is the concept of an embryogeny (which itself evolved in nature) that has allowed the evolution of these complex designs. It should be clear that evolutionary computation can benefit a great deal from computational embryology.

Further Work

Work is in progress on developing a new representation for a parallel implicit embryogeny. In which, the fundamental mechanisms behind biological concepts such as, regional specification, and cellular differentiation are to be employed, with a view to studying emergent phenomena.

Acknowledgments

This work is funded by Science Applications International Corporation (SAIC). Our thanks to the members of nUCLEAR for their useful comments and discussions on this work.

References

Bard, J. (1990). *Morphogenesis.* Cambridge University Press, UK.

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

Bentley, P. J. & Kumar, S.. (1999). Three Ways to Grow Designs: A Comparison of Embryogenies for an Evolutionary Design Problem. In Genetic and Evolutionary Computation Conference (GECCO) Orlando, Florida, USA.(to appear)

Bentley, P. J. & Wakefield, J. P. (1996). Hierarchical Crossover in Genetic Algorithms. In *Proceedings of the 1st On-line Workshop on Soft Computing* (WSC1), (pp. 37-42), Nagoya University, Japan.

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. *Proceedings of Artificial Life VI*. 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.

Mallinson, H and Bentley, P. J.. (1999) Evolving Fuzzy Rules for Pattern Classification. In International Conference on Computational Intelligence for Modelling, Control and Automation - CIMCA'99.

N. Jakobi (1995). Harnessing Morphogenesis. *International Conference on Information Processing in Cells and Tissues*, Liverpool, UK.

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

Koza, John R., Bennett III, Forrest H, Andre, David, and Keane, Martin A. (1999). *Genetic Programming III*. San Francisco, CA: Morgan Kaufmann.

Lawrence, P. A. (1995). *The Making of a Fly: The Genetics of Animal Design*. Blackwell Science Ltd, The Alden Press, Oxford, UK.

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

Slack, J. M. (1991). *From Egg to Embryo.* Cambridge University Press.

Turing, A.M. (1952). The chemical basis of morphogenesis. *Phil. Trans. R. Soc*., 237B, 37-72.

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