

Evolutionary Benefits of Evolvable Component Integration

David Malkin

University College London
11-43 Bath Street
London EC1V 9EL
0044 (0)2076084052
d.malkin@cs.ucl.ac.uk

R. Beau Lotto

University College London
11-43 Bath Street
London EC1V 9EL
0044 (0)2076084052
lotto@ucl.ac.uk

ABSTRACT

A system composed of multiple interacting components is capable of responding to contextual information and producing a higher range of non-linear responses to stimuli compared to modular systems with a low degree of component interaction. However, the fitness landscape of highly integrated systems is more rugged indicating that such systems are likely to be less evolvable. In this work we use an artificial life simulation to investigate whether the evolvability of highly integrated systems can be improved if the level of integration between the system's components is under evolutionary control. When evolving our multi-component system we discover that the level of integration very quickly falls to virtually zero reducing the ruggedness of the landscape and making it nearly neutral. This allows the evolving population to explore the genome space without getting stuck on local optima. The components then integrate and the evolving population settles on the global optimum. This work is unique because the presented problem requires the evolving system to be fully integrated in order to solve it and as such the decreased ruggedness and near neutrality are not a permanent feature of the landscape but rather a property which is temporarily manipulated and exploited by the evolving population.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence – *Multigent systems*.

General Terms

Theory

Keywords

Near neutrality, ruggedness, evolvability, artificial life, context dependence.

1. INTRODUCTION

What makes a population evolvable is one of the central points of discussion in biology [22] and increasingly computer science, where algorithms based on evolution are used to find solutions to problems, [2], [11]. In very simple terms evolvability means that an evolving population can easily produce new phenotype that are

of higher fitness, put another way, evolvability means that a particular fitness functions can easily be optimized using some form of an evolutionary search algorithm. High evolvability is associated with a high level of neutrality or near neutrality [12], [13], [8], [5], [19] and a low ruggedness [18], [9] of the fitness landscape. High levels of neutrality or near neutrality allow an evolving population to explore large parts of the genome space because there is a minimal change in fitness, allowing the population to move closer towards the global optima [21]. Low levels of ruggedness, usually measured using autocorrelation estimates, indicate that the fitness landscape has few local optima the evolving population can potentially get stuck on [10],[17].

The level of fitness landscape ruggedness is highly dependent on the amount of interactions between the components of the system in question. Modular systems (i.e. low interaction intensity between constituent components) are associated with a low number of local optima and hence a smoother fitness landscape. On the other hand, a system where the constituent components are highly dependent on the state of other components have many local optima and therefore a very rugged fitness landscape. Modularity is therefore often championed as one of the most important mechanisms which increases evolvability [14], [1],

Neutrality of the fitness landscape can be caused by redundancy in the genotype phenotype mapping which means that several different genotypes have the same phenotype and hence the same fitness [16], [7]. Neutrality allows an evolving population to drift through neutral networks of the genome space until a new genome that increases its fitness is found.

Near neutrality is different from absolute neutrality in the mechanisms that give rise to it but its effects can be similar. Near neutrality, just like low ruggedness can be caused by sparse connections between a system's components because in a sparsely connected system the overall changes to fitness due to a single mutation are likely to be small but non-zero. If the selection pressure is relatively low an evolving population will be able to move through the fitness landscape as if tough the fitness landscape was neutral, with all the evolutionary benefits this entails.

This work investigates the relationship between the amount of component interaction and fitness landscape ruggedness, neutrality and evolvability. We created an artificial life simulation consisting of 100 non-interacting agents each composed of eight components. The agents were presented with a 2D navigation problem but since the agent's components could only obtain limited, local and ambiguous information the components had to interact in order to solve the problem.

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-698-1/07/0007...\$5.00.

We find that the evolvability of the artificial life agents is increased if, in addition to the evolution of the components, the intensity of component interaction also evolves. Under such a case the intensity of interaction falls to virtually zero making the fitness landscape nearly neutral and less rugged. Then, after a variable period of time the level of component interaction increases and the population settles onto the global optima. When the intensity of component interaction is not under evolutionary control it is very difficult for the evolving population to find the global optimum. While the explicit task is contrived the results are of general relevance. Specifically, this work differs from other works on neutrality and ruggedness in two respects, first: the level of component integration is under explicit evolutionary control as opposed to it being imposed and second: the ideal solution requires complete component integration hence the near neutrality and low ruggedness are not isomorphic features of the landscape but rather just describe certain portions of it.

2. METHODS

A 2D artificial life system was created with a population of 100 agents. The agents were evolved using 2-tournament selection, single point crossover and mutation. Each agent is composed of 8 components. The agents are presented with a 2D navigation task that requires the components to move as close as possible towards 2 orthogonal lines. Each component has an explicit fitness which depends on how close the component is to the lines. The fitness of the agent is simply the sum of the individual component's fitness. Each component is 'aware' of the normal Euclidian distance between itself and one of the two lines. Which one of the two lines a component is 'aware' of depends on the agent's genome. The best location for a component to move towards is the intersection between the two lines because then it will be closest to both lines.

In order to ensure the components do not 'memorise' the exact location of the line's intersection the orientation of the lines is rotated and the initial position of the agents is altered several times per each generation. (see § Appendix for exact line and component coordinates).

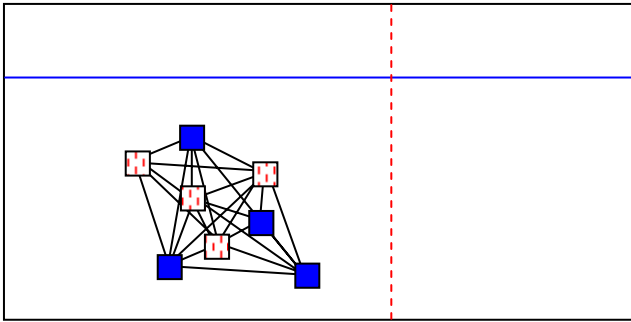


Figure 1. shows a schematic illustration of an agent in a given instance of an environment. (not to scale). The agent is composed of 'red' and 'blue' components that sense the distance to the red and blue line respectively. The agents have to move towards the point of intersection of the two lines.

Since each component is only given the absolute distance from one of the lines and since the components have neither memory nor can exchange information, no single component can independently determine which is a suitable direction to move in. The components can only find the intersection of the two lines if they interact with each other.

The direction and speed of motion of any component at any given time step is characterised by three different motion vectors. The first vector, referred to as the internal motion vector, depends on the component's genes and the distance between the component's current location and one of the given lines. The second motion vector, referred to as the external motion vector, is a scaled sum of all the other component's internal motion vectors. Finally, there is the absolute motion vector which is in fact the sum of the internal and external motion vectors. The relationship between the different vectors is given by

$$\overline{A}_i = \overline{I}_i + \overline{E}_i \quad (1)$$

Where \overline{A}_i is the absolute motion vector, \overline{I}_i is the internal motion vector and \overline{E}_i is the external motion vector for a component i .

The external motion vector is given by

$$\overline{E}_i = S \sum_{\substack{j=1 \\ j \neq i}}^N \overline{I}_j \quad (2)$$

where S is the intensity of component interaction and \overline{I}_j is the internal motion vector of component j . The value S is encoded in the agent's genome by 1 or more bits, depending on the experiment run (see below). The internal motion vector of any component i is given by:

$$\overline{I}_i = \overline{d} x \quad (3)$$

where \overline{d} (represented by two bits in the genome) can be any of the following 2D vectors $[1,0],[0,-1],[0,1],[1,0]$ representing a one unit motion of up, down, left or right respectively. x is a Bernoulli variable:

$$p(x; p) = \begin{cases} p \rightarrow x = 1 \\ 1 - p \rightarrow x = 0 \end{cases} \quad (4)$$

while the value p in equation (4) is calculated according to the following relationship:

$$p = \begin{cases} t(r - (a - 1)) + (1 - t)(b - (a - 1)) \rightarrow r, b \leq a \\ t((a + 1) - r) + (1 - t)((a + 1) - r) \rightarrow r, b > a \end{cases} \quad (5)$$

This rather cumbersome expression is actually rather straightforward. The variables t and a are both controlled by the genes of a given component. The 1 bit binary variable $t = \{0,1\}$ controls which of the two lines, 'blue' or 'red', are used for the distance measure, when $t=0$ the distance from the red line is measured, when $t = 1$ the distance to the blue line is measured. The 4 bit variable a represents any of the following values $\{0, 1/31, 2/31, \dots, 30/31, 1\}$. r and b are given by (6)

$$r = (1 - (|d_r|/100)) \quad (6)$$

$$b = (1 - (|d_b|/100))$$

where d_r is the distance between the component and the red line, d_b is the distance between the component and the blue line. If r or $b < 0$ then $r, b = 0$. Expression (5) essentially links the distance between a component and one of the lines with the probability of motion according to the evolved value a so that $|p| \propto \{|r, b\} - a|$.

The 2D surface the agents move on is bounded by a frictionless non-elastic boundary restricting the agent's location to a 200x200 surface area. Two or more components can occupy the same location without interference. The fitness of each agent is given

by equation (7)

$$W = \sum_{i=1}^8 w_i \quad (7)$$

Where w_i is the fitness of component i . The summation is done over all 8 components

$$w_i = \sum_{i=1}^{8000} r + b \quad (8)$$

w_i is given in terms of r , b which are defined in (6). The summation is done over 8000 turns (this is the number of turns per generation). It should be noted that each component has an explicit fitness, the fitness of an agent is the summation of each component's fitness. Essentially, the components of the agents are initialised in a given location with a particular line orientation and then their fitness is assessed. After 1000 turns the orientation of the lines are changed and the coordinates of the agents are re-set (see appendix). After this process has occurred 8 times (hence 8000 additions), 2 agent tournament selection with a 75% chance of the best agent winning is applied to the population of agents followed by single point crossover. Once this is completed 10% of the new agents have one of their bits mutated. Each experiment is run for 3000 generations. The population consists of 10 agents.

3. EXPERIMENTS AND RESULTS

To investigate what effects evolvable component integration has on evolvability we conducted 4 experiments following the methods described above. In the first experiment the intensity of component interaction was set to 1, i.e. $S = 1$ in equation (2), for the entire duration of evolution. In the second experiment the S gene was represented using 1 bit so that the components were either fully integrated or fully independent. In the third experiment the S gene was represented by 2 bits so that the level of integration could vary between $\{0, 0.33, 0.66, 1\}$ while in the last experiment the S gene was represented by 5 bits so that the level of integration between components was a value in the following set $\{0, 1/31, 2/31, \dots, 1\}$, (remembering that $S = 0$ indicates zero intensity component interaction while $S = 1$ indicates maximum intensity component interaction).

The fitness illustrated in figure 2 was normalised so that 1 represents the maximum possible score (an absolute fitness value of 14000) and 0 represents the worst maximum score obtained in one of the runs (an absolute fitness value of 11235). When the elements are fully integrated ($S = 1$) in experiment 1, the agents are unable to evolve the optimal behaviour. This is despite the fact that the optimal behaviour requires complete integration between the elements (see figure 2 Experiment 1). Instead, the average maximum score reached by agents that were de facto fully integrated was only 45% of the maximum score. When, however, the level of integration between the agent's components could evolve, the agents were more likely to evolve the optimal behaviour. Thus, when the level of integration could evolve, but only coarsely (i.e. the ability to alternate between full unit independence and full unit integration: $S = 0$ and $S = 1$, respectively) in Experiment 2, the agent's level of fitness increased to 55% of the maximum fitness (see Figure 2, Experiment 2). Increasing the resolution of component integration intensity still further enabled the agents to attain a normalized fitness of more than 85% to 95% of the maximum (see Figure 2, Experiment 3 and 4). As a control the agents were also evolved when the level of integration was $S = 0$ for the entire duration of the experiment (not shown in figure 2) and the maximum absolute

fitness attained was 7850, while the maximum possible score is 14000, clearly illustrating the requirement for the components to be integrated in order solve the presented problem.

Given the explicit nature of the task, agents in experiment 3 and 4 evolve so that the components are fully integrated ($S = 1$), like agents in experiment 1. However, unlike agents in experiment 1, being able to alter their level of integration during evolution enabled agents in experiment 3 and 4 to evolve the optimal solution. Indeed, consistent with our hypothesis, the level of integration between the components in experiment 3 and 4 varied significantly during evolution, the intensity of integration (S) always drops from an initial random value of 0.5 to virtually 0 (Figures 3a and 3b). After a variable period of component independence, the parameter S then increased until in the end the components are fully integrated. And yet when the elements are fully integrated from the outset (as is the case in experiment 1), the ideal behaviour is not evolved, despite the fact that setting $S = 1$ reduces the size of the search space to a region still containing the ideal solution.

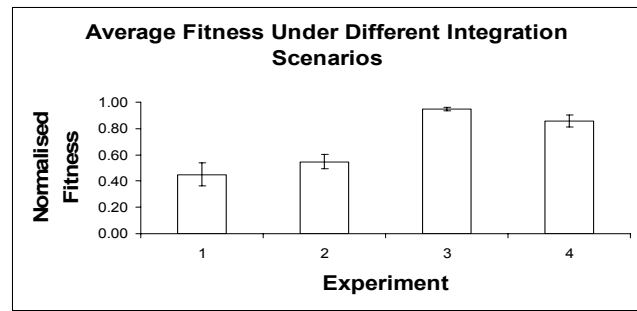


Figure 2. This figure shows the average maximum score for the agents (see text for the normalisation performed) when the components were capable of different levels of integration. In experiment 1 the components were fully integrated, $S = 1$, in experiment 2 $S = 0/1$. In experiment 3 the possible level of component integration was any value in the set $\{0, 0.33, 0.66, 1\}$ while in experiment 4 the level of component integration was any value in the set $\{0, 1/31, \dots, 1\}$.

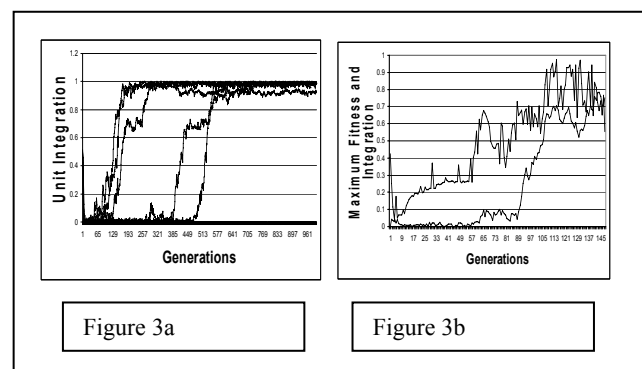


Figure 3a; Shows the average level of unit integration for 5 populations that reached above 95% of the maximum fitness. The number of generations a population of agents remain non-integrated appears random but all do finally integrate, otherwise the problem presented could not be solved. **Figure 3b;** shows the average level of integration (bottom line) and the maximum fitness (upper line) of a typical run.

The results demonstrate a clear relationship between agent success and the ability to evolve component integration. To formalise these observations we measured the level of fitness landscape ruggedness when the agent's components were integrated and when the agent's components were independent. A common method for measuring ruggedness of a fitness landscape is to analyze the autocorrelations of fitness value series [6];[20], [18]. A fitness value series is obtained by randomly initializing the genome of an agent and evaluating its fitness. The genome of the agent is then modified in some way and the fitness of the new agent is evaluated. This process is repeated until one has a series of fitness values, which describe aspects of the fitness landscape topology. An autocorrelation is then performed on this series. The autocorrelation is the expected value of the product of a series with a shifted version of itself as given by (5):

$$R(k) = \frac{E[(X_i - \mu)(X_{i+k} - \mu)]}{\sigma^2} \quad (8)$$

where X_i is the series in question, k is the number of steps by which the series is shifted, μ is the mean and σ^2 the variance. If the variance is not zero or infinity, the function is well defined and lies in the following range $[-1,1]$: 1 indicates complete correlation between the time series and a shifted version of itself, while -1 indicates complete anti-correlation.

In this study a fitness value series was created by mutating a single bit in the genome or by using single point crossover with randomly created genomes [15]. Both methods were applied to agents with fully integrated and fully independent components, resulting in four fitness value series in total. The higher the autocorrelation for a given value of lag k the smoother the fitness landscape, assuming fitness landscape statistical isomorphism. It should be noted that the autocorrelation measure does not capture neutrality in the fitness landscape, see for example fitness landscape portraits proposed by [18].

Results in figure 4 show the average of ten autocorrelation estimates for the four series under different values of lag k . The top two lines are autocorrelation estimates for agents with independent components (where $S = 0$). The bottom two lines are autocorrelation estimates for fully integrated agents (where $S = 1$). Clearly, when $S = 0$, the autocorrelation is higher than when $S = 1$, demonstrating that component independence is associated with a smooth fitness landscape, and component integration with a more rugged landscape. This is similar to autocorrelation estimates of the Kauffman NK landscape, where ruggedness increases as the number of epistatic interactions K increases [18].

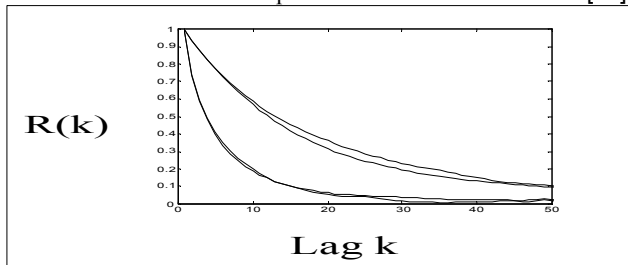


Figure 4; the top two lines show the estimated autocorrelations when the component are independent while the bottom two lines show the estimated autocorrelations when the components are integrated. There are two lines per case since one fitness series was generated by mutating a single bit while the other fitness series was generated through single point crossover with a random genome.

The ruggedness results in figure 4 indicate that increasing the level of component integration increases the amount of local optima making it more likely that an evolving population will get stuck on a local optima. In addition to measuring the ruggedness of the fitness landscape under different level of component integration the relationship between component integration and near neutrality were also measured. Absolute neutrality can be measured by recording the number of mutations that produce no change in fitness. However, since near neutrality can not be measured in this manner we looked at the average standard deviation of fitness values produced by a random walks of 400 mutations through the fitness landscape under different levels of integration. More specifically, 10 random agents were mutated 400 times with their fitness evaluated at every mutation. The 10 agents had their level of integration set to 0, 0.25, 0.5, 0.75 and 1 in turn. Figure 5 illustrates the average standard deviation under different levels of integration.

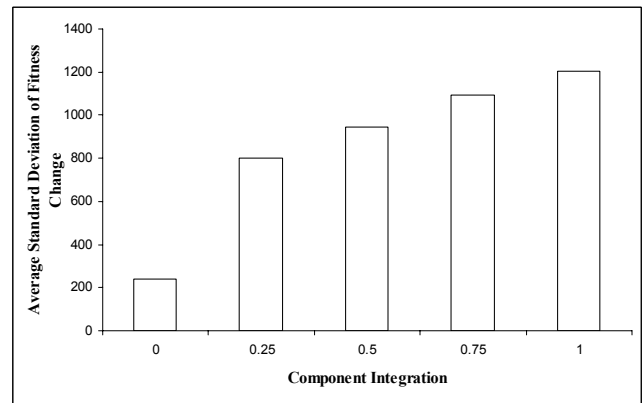


Figure 5. The average of 10 standard deviation values for a random walk through the fitness landscape under different level of component integration.

Figure 5 shows that the level of component integration is inversely proportional the neutrality of the fitness landscape. These results, in combination with results shown in figure 4 indicate that, at least in this model, the single parameter of component integration can reduce the ruggedness and increase neutrality of the fitness landscape.

4. DISCUSSION

For the constructed problem described above the optimum solution purposefully requires the agent's units to be fully integrated, making the optimal solution reside in the relatively rugged and non-neutral, less evolvable part of the fitness landscape [6]. The results in figure 2 Experiment-1 show that a population of agents restricted to this smaller, less evolvable part of the landscape, which nevertheless contains the ideal solution, are unlikely to find it. Rather, figure 2 Experiment -3 -4 agents and Figure 3a show that the optimal solution is more likely to be reached when the level of integration between an agent's elements is under genetic control (i.e. component integration is evolvable). When evolving under these conditions, the initial level of integration always falls from an average of 0.5 to virtually 0, before gradually increasing towards full integration ($S = 1$). Even though the initial population is randomly distributed across the landscape it quickly moves into the smooth and nearly neutral part of the landscape, before moving into the rugged and non-neutral part of the landscape where it finally reaches the global optimum.

In this study, the agents with maximum intensity component interactions are much less likely to evolve the ideal solution compared to agents that can control the intensity of component interaction, in the latter case the intensity of interaction always falls to zero before increasing to a maximum of one. The intensity of component interaction dimension, i.e. S in equation (2), can be viewed as a bypass dimension [3] [4] since its existence is not needed for a 'good' solution, rather, this dimension enlarges the fitness landscape in a non-rugged nearly neutral fashion increasing evolvability.

The central more general point of this work is the proposition that fully connected multi-component systems can become more evolvable if the intensity of interactions between the components is under evolutionary control. This allows an evolving system to reduce the intensity of component interaction hence potentially reducing the amount of local optima. Furthermore, the few local optima that remain will be less difficult to escape from since the landscape will be more neutral.

Figure 6 is a schematic illustration of a fitness landscape showing the evolutionary benefits of variable component integration. The rugged part of the fitness landscape is associated with a high intensity of component interaction while the smoother part of the fitness landscape is associated with a low intensity of component interaction.

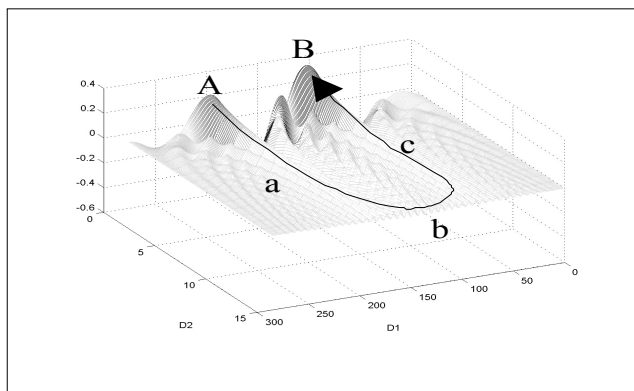


Figure 6. This hypothetical fitness landscape illustrates how reducing the intensity of interaction between components reduces the ruggedness of the landscape. The D1 dimension corresponds to different phenotypic parameters while the D2 dimension corresponds to the intensity of interaction between components.

If a population is trapped on the local optimum A it can still reach the global optimum B if a) the intensity of interaction is reduced b) the genome space is explored with minimal change in fitness c) the intensity of interaction increases. However, the 'intensity of component interaction' dimension only improve evolvability of an evolving population if the fitness values associated with the smooth part of the fitness landscape are sufficiently similar to the local optima being bypassed, otherwise the evolving population will never reach the smooth part of the landscape and so will be unable to explore the genome space. This implies that each component comprising the system must be relatively fit, even when independent of other components. This condition will only be met by some fitness functions.

Not only does this study inform work on evolutionary search algorithms employed by computer scientists to optimise different fitness functions but adds insight into studies of biological systems, in particular the study of evolution and trait integration.

5. CONCLUSION

Fully connected systems provide the necessary substrate for context-dependent, non-linear behavioural responses to environmental stimuli, thus potentially improving the system's overall fitness. On the other hand, the 'evolvability' of these systems is likely to be low, since altering any component of the system alters the function of all other components and hence is likely to result in lots of local optima and less fitness neutrality. Here we tested the hypothesis that the evolvability of integrated systems can be improved if the intensity of component interaction is under genetic control.

We tested this hypothesis using an artificial life simulation where the problem to be solved required the agent's components to exhibit a particular behaviour while fully integrated. Since the intensity of component interaction for the most successful agents always fell to virtually 0 and remained at this value for a number of generations before finally increasing to 1 the hypothesis was shown to be correct at least for this example fitness function and simulation.

We explained our findings in terms of decreased ruggedness which reduces the number of local optima in the fitness landscape and near neutrality which makes the escape from local optima more achievable.

We concluded our discussion by noting that variable intensity of component interaction will only improve evolvability in the way described in this study if the constituent components of the evolving system have an inherent and relatively high fitness even when independent of other components. This will clearly only be true for a certain class of fitness functions.

6. ACKNOWLEDGEMENTS

We would like to thank Peter Bentley for his assistance and conversation during this study. DM was funded by an EPSRC studentship.

7. REFERENCES

- [1] Altenberg, L. Genome growth and the evolution of the genotype-phenotype map. *Evolution and Biocomputation: Computational Models of Evolution*, 889: 205-259.
- [2] Bentley, P. J. & Corne, D. W. 2001. *Creative evolutionary systems*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA
- [3] Bongard, J. C. & Paul, C. Making evolution an offer it can't refuse: Morphology and the extradimensional bypass. *Proceedings of the 6th European Conference on Advances in Artificial Life*, 401-412.
- [4] Cariani, P. A. Extradimensional bypass. *Biosystems*, 64(1-3): 47-53.
- [5] Ebner, M., Langguth, P., Albert, J., Shackleton, M., & Shipman, R. On neutral networks and evolvability. *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, 1*.

[6] Hordijk, W. A Measure of Landscapes. *Evolutionary Computation*, 4(4): 335.

[7] Huynen, M. A. Exploring phenotype space through neutral evolution. *Journal of Molecular Evolution*, 43(3): 165-169.

[8] Izquierdo-Torres, E. The Role of Nearly Neutral Mutations in the Evolution of Dynamical Neural Networks. *Ninth International Conference of the Simulation and Synthesis of Living Systems (Alife 9)*, 322-327.

[9] Jones, T. & Forrest, S. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. *Proceedings of the 6th International Conference on Genetic Algorithms Table of Contents*, 184-192.

[10] Kallel, L., Naudts, B., & Reeves, C. R. Properties of fitness functions and search landscapes. *Natural Computing Series*, 175-206.

[11] Mitchell, M. 1996. *An introduction to genetic algorithms*. MIT Press Cambridge, MA, USA

[12] Newman, M. E. J. Effects of selective neutrality on the evolution of molecular species. *Proceedings: Biological Sciences*, 265(1403): 1333-1338.

[13] Ohta, T. Evolution by nearly-neutral mutations. *Genetica*, 102: 83-90.

[14] Peretz, I. Modularity of music processing. *Nature Neuroscience*, 6(7): 688.

[15] Reidys, C. M. Combinatorial landscapes. *SIAM Review*, 44(1): 3.

[16] Schuster, P., Fontana, W., Stadler, P. F., & Hofacker, I. L. From Sequences to Shapes and Back: A Case Study in RNA Secondary Structures. *Proceedings: Biological Sciences*, 255(1344): 279-284.

[17] Skellett, B., Cairns, B., Geard, N., Tonkes, B., & Wiles, J. Maximally rugged NK landscapes contain the highest peaks. *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, 579-584.

[18] Smith, T. Fitness Landscapes and Evolvability. *Evolutionary Computation*, 10(1): 1.

[19] Smith, T., Philippides, A., Husbands, P., & O'Shea, M. Neutrality and ruggedness in robot landscapes..

[20] Stadler, P. F. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20(1): 1.

[21] Toussaint, M. & Igel, C. Neutrality: A Necessity for Self-Adaptation. *Arxiv Preprint Nlin.AO/0204038*.

[22] Wagner, G. P. Complex adaptations and the evolution of evolvability. *Evolution*, 50(3): 967.

Case1 (85-65, 85-65, 85-35, 85-35, 115-65, 115-65, 115-35, 115-35)

Case2 (85-165, 85-165, 85-135, 85-135, 115-165, 115-165, 115-135, 115-135)

Case3 (85-65, 58-65, 85-35, 85-35, 115-65, 115-65, 115-35, 115-35)

Case4 (85-165, 85-165, 85-135, 85-135, 115-165, 115-165, 115-135, 115-135)

Case5 (85-115, 85-115, 85-85, 85-85, 115-115, 115-115, 115-85, 11-85)

Case6 (135-115, 135-115, 135-85, 135-85, 165-115, 165-115, 165-85, 165-85)

Case7 (85-165, 85-165, 85-135, 85-135, 115-165, 115-165, 115-135, 115-135)

Case8 (135-165, 135-165, 135-135, 135-135, 165-165, 165-165, 165-135, 165-135)

This table shows the coordinates of the blue and red lines for each of the eight cases per generation.

		x	y		x	y
	case 1					
	red	0	100		200	100
	blue	50	0		50	200
	case 2					
	red	0	100		200	100
	blue	50	0		50	200
	case 3					
	red	0	100		200	100
	blue	150	0		150	200
	case 4					
	red	0	100		200	100
	blue	150	0		150	200
	case 5					
	red	100	0		100	200
	blue	0	150		200	150
	case 6					
	red	100	0		100	200
	blue	0	150		200	150
	case 7					
	red	100	0		100	200
	blue	0	50		200	50
	case 8					
	red	100	0		100	200
	blue	0	50		200	50

APPENDIX 1

The initial coordinates of each component for each one of the eight cases is given below in xy pairs.