

Heterochrony and Evolvability in Neural Network Development

Artur Matos, Reiji Suzuki and Takaya Arita

Graduate School of Information Science, Nagoya University

Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

amatos@alife.human.nagoya-u.ac.jp, {reiji, arita}@nagoya-u.jp

Abstract

Recent studies in evolutionary computation have focused on using developmental processes together with genetic algorithms in order to achieve more complex designs. Although several models have been proposed, their growth dynamics, and their interactions with evolutionary algorithms are still poorly understood. One particularly neglected concept in artificial developmental systems is *heterochrony* — how evolution affects development by changing the timing and rate of developmental events. In this paper we attempt to address this issue by analyzing heterochronic changes in a well known artificial developmental model — the cellular encoding model — by using an heterochrony framework by Alberch *et al.* We have conducted experiments by evolving networks to solve a boolean problem, and analyzed heterochronic changes in both successful and unsuccessful runs. Our findings show that the cellular encoding model, due to its properties, strongly affects the developmental dynamics and the heterochronic changes that occur during evolution. Our experiments also show that hypermorphic changes (a kind of heterochronic occurrence) lead to greater evolvability in successful runs.

Introduction

Recently there has been an increasing interest within the Evolutionary Computation (EC) community in simulating developmental processes alongside evolution (Stanley and Mikkulainen, 2003). This approach has already proved to be fruitful, allowing evolutionary algorithms (EAs) to generate more complex designs than traditional approaches, and it has been applied to a wide range of domains, including, among others, neural networks (Gruau, 1994) and artificial creatures (Bongard, 2002). Although several of these models have been proposed, there is still no throughout understanding on how they work, specially on how evolution and development interact. One particularly important area, and that we will address in this paper, is to try to understand how the EA shapes the individuals by rearranging the underlying developmental events.

In evolutionary and developmental biology, this change in the rate, timing and order of developmental events caused by evolution is generally known as *heterochrony* (Klingenberg, 1997). Heterochrony is a well observed phenomena and prevalent in the evolution of species. A well known example

can be found in the Mexican axolotl salamander: most salamander species have two distinct stages of development, a larval and an adult stage. However, the Mexican axolotl does not undergo metamorphosis, and achieves sexual maturity in what would still be considered a larval form. Therefore, from an evolutionary point of view, we can say that evolution shaped the axolotl species by “slowing down” their ancestor’s development. Using more precise terminology, this is an example of *neoteny*. Because examples like this are so common in nature, it is speculated that heterochrony is one of the major factors in the evolution of more complex taxa.

Unfortunately, and despite of this, EC studies on heterochrony are still lacking. A recent review paper on artificial developmental models (Stanley and Mikkulainen, 2003) identified heterochrony as one of the important dimensions to pursue in research, but there are still very few published results on the topic: (Cangelosi, 1999) described heterochronic occurrences in a developmental model based on genetic regulatory networks (GRNs). The author evolved neural networks for solving a food foraging problem, and then compared developmental events between ancestor and descendant networks. By proceeding this way, he was able to identify several different kinds of heterochrony, for instance, occurring in cell division events and during axon growth. (Bongard, 2002) also used a GRN-based model, but for evolving artificial creatures, with coevolved morphologies and neural networks. The author evolved individuals for a locomotion task, and then performed mutation experiments to analyze the evolved GRNs. He then observed that mutation in some individuals caused morphological units to appear earlier or later compared to the original individual.

Although these previous studies show that heterochrony does indeed occur in artificial developmental models, there are still several questions remaining unanswered. First, it is still not clear to what extent these models support heterochrony, or if they direct evolution to certain kinds of heterochronic occurrences more than others. A second, and perhaps even more important question is to understand how heterochrony relates to EA performance: for instance, can we expect certain kinds of heterochronic occurrences to be

more conducive to evolvability than others? As a starting point for the first question, in a previous paper (Matos et al., 2005), we have applied a heterochrony framework to a well known developmental model — the cellular encoding model developed by Gruau (Gruau, 1994). This framework, by Alberch *et al* (Alberch et al., 1979), offers both a precise terminology and methodology to study heterochronic phenomena in living systems. In this paper, we extend our previous results, and also attempt to undertake the second question. For this, we have used the cellular encoding model to evolve neural networks to solve a boolean problem, and applied the framework to a large ensemble of networks. This allows us to characterize heterochrony in artificial systems in a more complete way than the previous studies. For the second question, we have applied the framework to both successful and unsuccessful experiments, and compared the heterochronic occurrences between them, to check for any meaningful differences.

Alberch *et al*'s framework

The framework by Alberch *et al* is widely used in biological systems for analyzing heterochronic phenomena. This framework is based on the measurement and comparison of quantitative traits, for instance, body length, width or height. The traits are measured as development unfolds, yielding growth curves. These growth curves can then be compared between related species for understanding the heterochronic change involved.

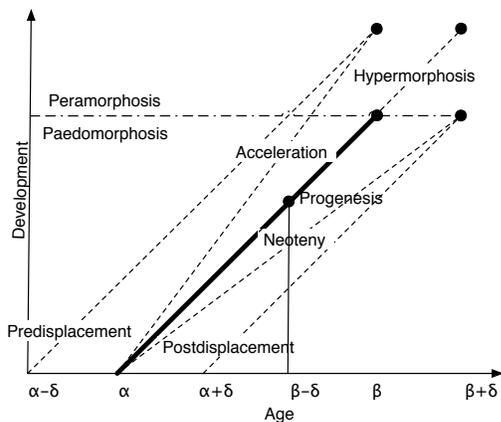


Figure 1: The formalism of Alberch *et al*. A trait measure is plotted against developmental time in the X axis. The solid line plotted from α to β represents the growth curve for the ancestor, while the remaining ones possible heterochronic outcomes for the descendant.

The basis for comparison lies on three metrics that can be extracted from the growth curves: α — the time when growth starts, β — the time when growth ends, and K — the growth rate. Comparing these values between species yields the outcomes summarized in Figure 1. For instance,

considering only changes in the K parameter, two outcomes are possible: if the descendant would grow faster than the ancestor (K would be larger), the corresponding outcome is *acceleration*. The reverse process — the descendant growing slower than the ancestor — is labeled *neoteny*. For the β parameter, *hypermorphosis* means that a descendant grows for a longer period than its ancestor, while *progenesis* refers to the opposite — the descendant stops its growth at an earlier time than its ancestor. For the α parameter, *predisplacement* means that growth starts earlier in the descendant species when compared with its ancestor, with *postdisplacement* referring to the opposite. Furthermore, *isomorphosis* refers when no change occurs between the growth of the ancestor and descendant.

Cellular encoding

Cellular encoding is a developmental model originally proposed by Gruau (Gruau, 1994) for evolving neural networks. We have chosen this model because it has a proven track for evolving neural networks, and for a wide range of problems, including threshold neural networks for boolean problems (Gruau, 1994) and controllers for robot locomotion (Gruau, 1995). Instead of using GRNs as in (Cangelosi, 1999) or (Bongard, 2002), the cellular encoding model specifies development as a set of graph-rewriting instructions that are evolved directly. The original model, that we follow in this paper, only evolved simple threshold networks: the neurons were threshold neurons with thresholds of either 0 or 1, with the connections between the neurons being either -1 or 1.

Cellular encoding defines a set of commands for operating on graphs, that changes the graph as development unfolds. In its original description, development is described as a sequence of these commands, grouped in a Genetic Programming (GP) tree. These trees are then used just as standard genotypes in a GP system. Each network starts as a single neuron, with a pointer pointing to the root of the tree. Development on each neuron proceeds by executing sequentially the nodes with the developmental commands. There are different kinds of commands, including commands for dividing cells, creating new connections, setting thresholds and so on. For instance, the `PAR` instruction, when applied to an existing neuron, creates a new neuron and copies the connections (both the input and output connections) from the original neuron to the new one. `PAR` nodes contain two children, that are inherited by each neuron (the original and the new one), allowing for cell differentiation to occur. Development in the network occurs in a parallel fashion: on each time step each neuron executes the command pointed by its register in the tree and moves to the following leaf. The developmental process is over when all the neurons have reached their final leaf node in the tree. A summary of the commands is shown in table 1, and a small example in figure 2.

Experiments

In his original experiments, Gruau evolved networks only for solving boolean problems, for instance, the odd-parity problem and the symmetry problem. In this paper, we decided to use a different boolean problem: the function we are trying to optimize has 3 inputs and 1 output, with the inputs ranging from 000 to 111. All the inputs in this function produce the output 0 except for two entries corresponding to inputs 011 and 100, where the output is 1. Our experiments have shown that this problem is particularly difficult to solve, and therefore could be used as a good problem for checking relationships between heterochrony and evolvability.

The fitness function used consisted of two parts: the first accounting for the number of right output values, and the other part rewarding networks with the right number of input and output neurons. The first part just computes the number of correct output values, ranging from 0 to 8. The second part returns the number of input neurons (or 3, whichever is smaller), and the number of output neurons (or 1, whichever is smaller). This latter part was deemed necessary to ensure that the problem could be solved at all. They are then weighted (with 0.75 for the number of right output values, and 0.25 for the number of right input/output neurons), and normalized between the range $[0, 1]$ with 1 as the best fitness. Furthermore, because tree depth tended to increase rapidly during evolution (tree bloat), we have imposed an upper limit of 30 neurons in the networks; all the networks that exceed this limit were assigned a fitness value of 0.

We used a GP system, with both population size and the number of generations set to 300. Tournament selection was used, with a tournament size of 7. Crossover was not used, and all the individuals were mutated with a 70% probability. The mutation operator follows Koza's GP original description: if a mutation occurs, first a node is selected at random in the GP tree; The node is then replaced with a new random subtree, with a maximum depth of 5. Experiments were conducted with the java-based ECJ software.

We have conducted several evolutionary runs, all with different random seeds. Because of the problem difficulty, several of the runs never found any optimal network, and converged into a local optima. For analysis, we kept 20 runs of all the ones conducted: 10 where the optimum was reached, and 10 that converged to a local optima.

Analysis

(α, β, K) dynamics

All of the successful runs exhibited similar behavior. On average, it took 110.7 generations to reach the optimum. A fitness graph of a typical run can be seen in Figure 3, and an optimal network in Figure 4.

In order to apply the Alberch *et al* framework to the model we needed to choose suitable traits for analysis. We decided

Table 1: Cellular encoding commands.

Command	Description
PAR	Parallel division: divides a neuron into two, with the incoming and outgoing connections copied to the new neuron.
SEQ	Sequential division: divides a neuron into two, the first neuron inherits the incoming connections, while the second neurons inherits the outgoing connections. A single connection is added to the two.
DECBIAS	Sets the neuron's threshold to 1.
INCBIAS	Sets the neuron's threshold to 0.
DECLR	Decreases the link register by 1. The link register is an internal variable in each neuron that points to the current incoming connection that is being manipulated.
INCLR	Increases the link register by 1.
VAL+	Sets the weight of the current incoming connection to 1.
VAL-	Sets the weight of the current incoming connection to -1.
CUT	Removes the current incoming connection.

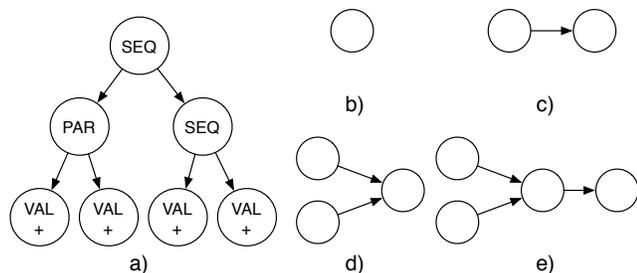


Figure 2: Development of a simple genotype in the cellular encoding model. a) the sample genotype, b) the initial neuron, c) the network after the first SEQ division, d) the left neuron executes the PAR instruction, e) the right neuron executes the SEQ instruction. The VAL+ instructions at the leaves set the weights of the connections to 1.

to consider the number of nodes and the average network degree (considering both incoming and outgoing connections), as they are the essential measures related to the network topology. We also decided to analyze the fitness dynamics, and treated the intermediate fitness values — how the fitness changes during development — as another trait in the framework. By proceeding this way, we could then check if the fitness dynamics played an important role in evolution, although for evaluation purposes only the fitness value of the last developmental step is considered. To get an idea how these traits change with development, Figure 5 shows growth curves of these 3 traits, in two different individuals. The dynamics on the right are from the best network in Figure 4.

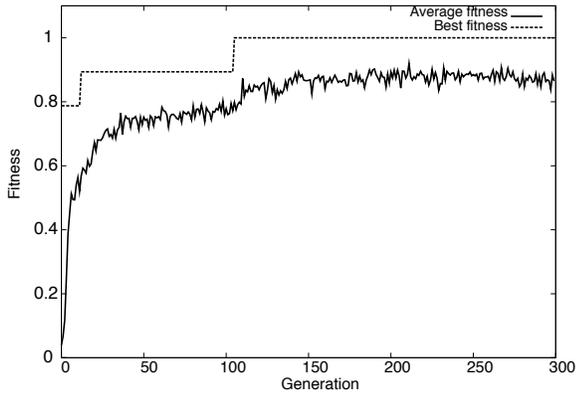


Figure 3: Fitness graph of a typical run.

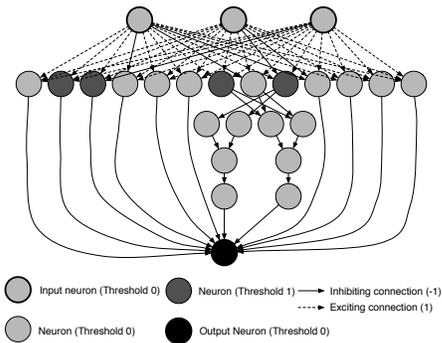


Figure 4: An example of an optimal network, found during our evolutionary runs.

Furthermore, to apply the framework, we need to extract the three parameters — α , β and K — from the growth curves. The framework itself does not define a method for doing this, and several methods have been proposed in the literature. In biological studies, one common method is to fit the data to a growth model (for instance, the Von Bertalanffy growth curve) by using non-linear regression, and

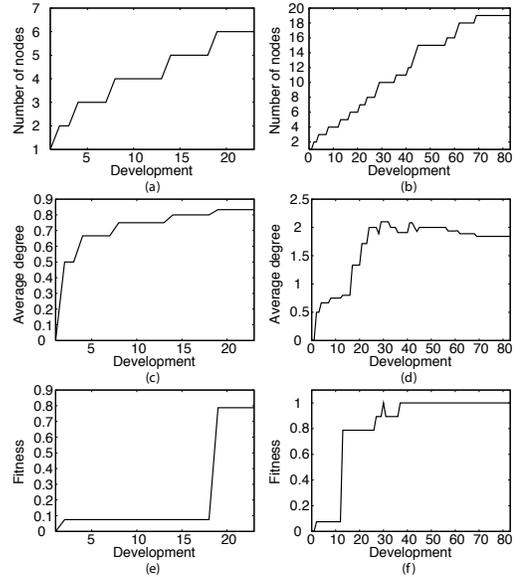


Figure 5: Growth curves of the analyzed traits. These correspond to the best lineage taken from Figure 3.

then extract the parameters from the fitted equation. This works well for biological data because they tend follow well-known patterns, and there are several sensible growth models available in the literature. In contrast, as shown in Figure 5, our growth curves tend to be rather irregular, so this approach is not feasible. Therefore we decided to use a simpler approach: for our growth curves, we defined α and β as the developmental time where growth effectively starts and stops in the data, that is, where changes in the values occur for the first and last time during development. As for K , we defined it as the average growth rate. Figure 6 shows the evolution of these three parameters in one typical, successful run. This example refers to the best lineage, that is, all of the individuals, starting from the first generation, that gave rise to the best individual in the last generation.

In this example, we can see that beta increased and K decreased through generations. This heterochronic change corresponds to the combination of hypermorphosis and neoteny in Figure 1. One interesting point is that α didn't change at all during evolution, for any of the 3 traits, and in any of the conducted runs. The reasons beyond this are two: the first reason is that the cellular encoding model represents the developmental events sequentially in the genotype, following tree order. Although the mutation operator does choose a node randomly in the tree, successful mutations tend to target nodes at a greater depth, because these tend to create smaller changes in the phenotype, and allow evolution to occur gradually. A change in the top-level node of the tree basically amounts to replace the original individual with a completely unrelated one, so it does not take into account

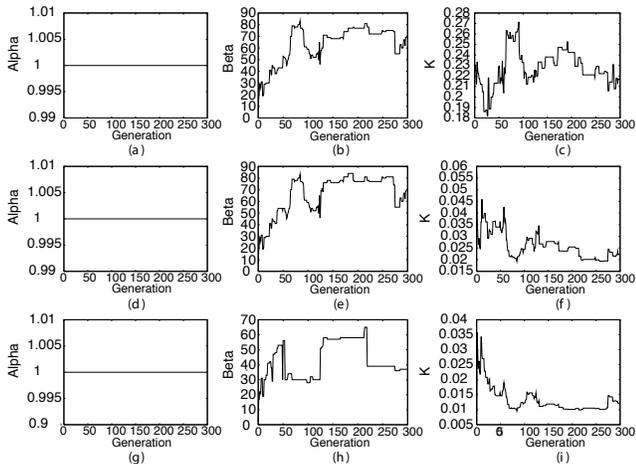


Figure 6: Evolution of the α , β and K parameters in the best lineage, in one of the conducted runs. Top row: number of nodes; middle row: average degree; bottom row: fitness.

any of the previously found solutions. The second reason is that in the cellular encoding model, almost all of the instructions must always change the network in some way; therefore it is difficult for the model to generate delays in the timing of events.

Correlation analysis

We decided further to check if there was any significant correlation in the dynamics, by applying the Pearson's correlation coefficient to the data:

$$\text{cor}(X, Y) = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)S_X S_Y} \quad (1)$$

with X_i and Y_i the parameter values of X and Y at generation i , \bar{X} and \bar{Y} as the mean values of X and Y , and S_X and S_Y as their standard deviation. A coefficient of 1 indicates a perfect correlation, and -1 a negative perfect correlation.

We have applied this coefficient between individual parameters in the same trait (for instance between β and K of the number of nodes trait), as well as for the same parameters in different traits (for instance, between β in the number of nodes trait and β in the average degree trait). Because α never changes during the runs, α was not considered in this analysis. We have also computed the coefficients separately for successful and unsuccessful runs. The results are shown in Table 2. This data refers to the best lineages only, as in the example shown in Figure 6.

The correlation table shows how the cellular encoding model further constrains the developmental dynamics. To understand it better, first it is necessary to explain how the model creates connections in the networks: in the cellular encoding model, there is no explicit command for adding

Table 2: Correlation table for the heterochronic parameters. Entries in gray represent values where the differences between successful and unsuccessful runs showed a strong statistical difference (at the $p = 0.05$ level). NN - number of nodes; AD - average degree; F - fitness.

Correlation	Successful		Unsuccessful		p
	Mean	SD	Mean	SD	
Same Trait:					
($\beta - K$)					
NN	0.328	0.435	0.165	0.258	0.3211
AD	-0.836	0.061	-0.714	0.059	0.0002
F	-0.523	0.336	-0.134	0.344	0.0202
Trait-Trait:					
NN - AD					
$\beta - \beta$	0.983	0.013	0.951	0.043	0.0357
$K - K$	-0.125	0.361	0.109	0.251	0.1086
NN - F					
$\beta - \beta$	0.57	0.394	0.124	0.321	0.0127
$K - K$	-0.147	0.398	-0.011	0.234	0.3626
AD - F					
$\beta - \beta$	0.597	0.379	0.123	0.316	0.0072
$K - K$	0.804	0.07	0.727	0.118	0.0965

new connections, although there is a command for removing them. The only way to create new connections in the networks is by cell division, that is, by creating new neurons. Therefore the number of nodes trait and average degree cannot evolve independently, and become highly correlated. This can be seen in the correlation coefficient of the number of nodes β and average degree β : in most cases, the growth of the average degree trait stops growing effectively when the number of nodes also stops growing, except when remove connections commands are found after the last cell division command. This seems to rarely happen, however, as it is shown in the table. A similar reason is behind the high correlation between the average degree's β and K : as indicated before, our fitness function imposes an upper limit on the number of neurons that a network may have; this combined with what was explained now effectively defines an upper limit on the average degree as well. This makes β and K negatively correlated, in order for the networks not to exceed this upper limit. This also explains why on the unsuccessful runs this correlation is lower.

The lower rows in the table, concerning correlations between the topological traits and fitness are also important, because they show how the dynamics contribute to the fitness as whole. The first thing that can be observed is that the average degree parameters are more strongly correlated with fitness than the number of nodes. This is to be expected, because on this problem the connectivity of the network (the way that the neurons are connected to each other) is more important for solving the problem than the number of

Table 3: Classification of heterochrony occurrences in all runs. for the number of nodes trait. I - Isomorphosis; A - acceleration; N - neoteny; P - progenesis; HM - hypermorphosis.

Trait	Successful		Unsuccessful		p
	Mean	SD	Mean	SD	
NN					
I	198.5	16.5	165.6	23.6	0.002
A	48	6.7	64.9	12.4	0.0014
N	51.9	10.2	67.9	12.8	0.0065
P	33	4.3	46.8	8.9	0.0004
HM	47.9	8.3	59.5	13	0.02914

nodes. Other important point is that higher β correlations between the topological traits and fitness seem important to assure successful evolution. Unfortunately, because we have only conducted analysis using this problem so far, at this time it still difficult to say why this occurs, and if it can be generalized to other problems.

Occurrence analysis

Based on the previously computed parameters, we further classified the changes according to the framework, with the results depicted on Table 3. When applying this framework a single comparison between an ancestor and a descendant yield changes in the three parameters simultaneously: for instance, a mutation could generate an increase in α , a decrease in β and an increase in K corresponding to a combined event of postdisplacement, progenesis and acceleration. In order to simplify the analysis, we have decided to classify the changes in the three parameters as if they were occurring independently. Also, because the results for all the three traits were similar, only the number of nodes trait is displayed.

As it is shown in the table, all possible outcomes (with the exception of predisplacement and postdisplacement, as it was explained before) occur with significant values; this, therefore, shows that the cellular encoding model is able to generate most kinds of heterochronic events. One interesting property is that isomorphosis (that is no change in the 3 parameters), is the most frequent occurrence in all the runs, and it occurs with more frequency in successful than unsuccessful runs; on the other hand, all the other possible outcomes (real changes in the parameters) occur with more frequency in unsuccessful than successful runs, and these differences are all statistically significant.

On average, 49% of the occurrences in Table 3 were real changes (no-isomorphic); this is much lower than what should be expected, considering the mutation probability that was used (70%). This means that selection favors isomorphic changes in the system (that is, neutral mutations), and overall it should be favoring isomorphic changes in the

Table 4: Ratios of heterochronic occurrences before the optimum was reached, for the number of nodes trait.

Occurrence	Successful		Unsuccessful	
	Mean	SD	Mean	SD
I	0.4323	0.089	0.3226	0.2794
A	0.1409	0.0284	0.1920	0.0991
N	0.1837	0.0410	0.2166	0.1222
HM	0.1627	0.0342	0.1273	0.0919
P	0.0802	0.0328	0.1413	0.0721

successful runs. One possible reason for this is that there could be an increased pressure for neutral changes after the optimum has been reached. We decided to check if this occurred, by performing the following analysis: on all experiments, we have divided the individuals in the best lineages into two groups: the first group with all the individuals before the optimum has been reached, and the second group with the remaining ones. For the unsuccessful experiments, we considered the highest reached value as the optimum. Then, for each group, we have computed the ratios between each kind of heterochronic occurrence divided by the total number of occurrences. Statistical tests showed that there is significant difference between the ratios before and after the optimum has been reached in 4 of the 10 successful runs.

Nevertheless, as shown in Table 4, the distribution of heterochronic occurrences is different among successful and unsuccessful runs, although we need more experimental data so as to improve the reliability of the analysis. The results are summarized in Table 4. Only the ratios before the optimum are shown. In both runs, neoteny is the largest heterochronic occurrence after isomorphosis, but hypermorphosis is much more common in successful than unsuccessful runs. It is the second most frequent occurrence in successful runs, while the least occurring one in unsuccessful runs. This could mean that hypermorphosis is an important factor for evolvable runs, at least for this problem.

With this in mind, we decided to check if this occurred by computing the total changes in fitness that each kind of occurrence has lead to: for instance, for the neoteny case, we have computed all the differences in fitness values due to neotenic occurrences, and divided it by the total number of neotenic changes. The results are shown in Table 5. These results back up our previous analysis, by showing that hypermorphosis contributed much more to increase fitness in successful than unsuccessful runs; in contrast, in unsuccessful runs, acceleration, the most contributing heterochronic occurrence, only contributed as half as hypermorphosis in the successful case. Therefore, hypermorphic changes in successful runs are more adaptive than the ones occurring in unsuccessful runs, but it still not clear in what way they contribute to evolvability. One possibility is that hypermorphosis is increasing the search space used by evol-

Table 5: Ratios between changes in fitness and the number of heterochronic occurrences.

Occur.	Successful		Unsuccessful		p
	Mean	SD	Mean	SD	
I	0.00006	0.00019	0.00005	0.00017	0.9027
A	0.00704	0.00738	0.00454	0.00648	0.4313
N	0.00803	0.00726	0.00376	0.00456	0.1327
HM	0.01072	0.00866	0.0024	0.00383	0.0124
P	0.00292	0.00193	0.0039	0.00669	0.6616

ution, and therefore making mutations more efficient. Because hypermorphic changes correspond to increases in developmental time (β), this also increases the range, in developmental time, where mutations may occur. This, in turn, increases the range of phenotypes reachable by mutation, allowing the search space to be more efficiently explored.

Conclusion

In this paper we have shown that the framework by Alberch *et al* is a valid method for studying heterochrony in artificial systems, by applying it to a typical artificial developmental model, in this case the cellular encoding model. Our results show that this model constrains developmental dynamics at least in two ways: first, because the GP trees are executed sequentially during development, predisplacement and postdisplacement events are very unlikely to occur; second, because the cellular encoding model can only create new connections by adding new neurons, development of the topological traits are highly correlated.

Our results concerning heterochrony and evolvability are summarized as follows: first, our results point out that changes in the average degree parameters are strongly correlated with fitness. Second, higher β (growth ending time) correlations between the topological traits and fitness seems important to ensure successful evolution. Third, successful runs had more isomorphic occurrences than unsuccessful runs, and hypermorphic changes lead to better evolvability in successful runs. However, it is still necessary to apply this framework to other tasks to see if these results generalize to other problems. Future work will focus on these topics, and also extend the current analysis to different kinds of models.

References

Alberch, P., Gould, S. J., Oster, G. F., and Wake, D. B. (1979). Size and shape in ontogeny and phylogeny. *Paleobiology*, 5(3):296–317.

Bongard, J. (2002). Evolving modular genetic regulatory networks. In *Proceedings of The IEEE 2002 Congress on Evolutionary Computation*, Piscataway, NJ, USA. IEEE Press.

Cangelosi, A. (1999). Heterochrony and adaptation in developing neural networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1241–1248. Morgan Kaufmann.

Gruau, F. (1994). *Neural Network Synthesis using Cellular Encoding and the Genetic Algorithm*. PhD thesis, Laboratoire de la Informatique du Parallelisme, Ecole Normale Supérieure de Lyon.

Gruau, F. (1995). Automatic definition of modular neural networks. *Adaptive Behaviour*, 3(2):151–183.

Klingenberg, C. P. (1997). Heterochrony and allometry: the analysis of evolutionary change in ontogeny. *Biological Reviews*, 73(1):79–123.

Matos, A., Suzuki, R., and Arita, T. (2005). Evolution of development and heterochrony in artificial neural networks. In *Proceedings of the Tenth International Symposium on Artificial Life and Robotics*, pages 635–638, Beppu, Oita, Japan.

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