Redundancy in the Evolution of Artificial Gene Regulatory Networks for Morphological Development

Lisa Schramm
Technische Universität Darmstadt
Karolinenplatz 5
64289 Darmstadt, Germany
Email: lschramm@rtr.tu-darmstadt.de

Yaochu Jin
University of Surrey
Guildford, Surrey, GU2 7XH, UK
Email: Yaochu.Jin@surrey.ac.uk

Bernhard Sendhoff
Honda Research Institute Europe
Carl-Legien-Str. 30
63073 Offenbach, Germany
Email: Bernhard.Sendhoff@honda-ri.de

Abstract—This paper investigates the influence of redundancy on the evolutionary performance of a gene regulatory network governing a cellular growth process. Redundancy is believed to play a key role in robustness and evolvability of biological systems. We use a cellular model controlled by a gene regulatory network to evolve elongated morphologies. We show that removing the redundancy in the genome during the evolution decreases the performance of the evolution strategy. A comparing run with the fewe parameters and therefore no redundancy performs worst, which supports the hypothesis that redundancy improves evolvability.

I. INTRODUCTION

The development of biological organisms is controlled by its genes and starts with an egg which develops to the whole organism. Simulating and analyzing biological development should on the one hand shed some light in biological processes but on the other hand be helpful to build technical systems.

Robustness is one of the most important design principles of biological systems. It can be achieved by a variety of mechanisms [1]. Wagner found that genetic redundancy is one of the main mechanisms in biology [2]. Meanwhile, an inherent trade-off between redundancy and evolvability has been revealed in [3] for a redundant genotype-phenotype mapping. Whitacre confirmed such a trade-off from a slightly different perspective by presenting the evidence that degeneracy, a partial redundancy, is a fundamental source of both robustness and evolvability [4].

A concept that is closely related to redundancy and robustness is neutrality [5]. Kimura might have been the first to notice the role of neutrality in biological evolution [6]. He argued that most mutations in nature are selectively neutral, which suggests that random genetic drifts may be one of the main driving forces behind the evolution. The relationship between neutrality and robustness has also been widely studied in evolutionary computation [7]. Yu and Miller analyzed different problems with different types of neutrality and found that redundancy can but need not be beneficial for evolution depending on the implementation [8]. Banzhaf proposed a model using a genotype-phenotype mapping with neutrality and found that neutrality allows the system to work more flexibly [9].

Biological design principles are more and more employed to solve complex engineering problems, such as shape and structural design [13], [14], [15], evolvable hardware [16], [17], controller design [18], [19], and self-organization of swarm robots [20], [21]. Biological design principles can augment standard engineering practices and thereby inject some properties that we admire in biological systems like scalability, robustness, self-organization, self-repair and sustainability into technical systems.

This paper investigates the role of genetic redundancy in a computational model for evolutionary development for evolving an elongated body morphology. The developmental model is conceptually based on the one proposed by [22] and has been applied successfully to structural design [15], development of primitive nervous systems [23], and body plans [24]. Therefore, using in silicio experiments, we aim at a better understanding of the role of redundancy during the evolutionary process. This increased understanding can then be fed back into the analysis of biological data and it can be used to optimally set-up design processes that exploit biological principles for engineering tasks. Biological evolution has to carry the “raw material” from which innovations can be made with it as opposed to a standard engineering approach where the material is supplied externally. The two approaches are conceptually very different and we expect that the difference has profound consequences.

As we have noted “redundancy” is closely related to other system properties such as evolvability, stability, and neutrality, most of which lack a clear and well accepted definition. For example, redundancy has many diverse definitions in different disciplines. Therefore, in Section II, we provide the definition of redundancy that we will use for analysis in this paper.

Following a brief introduction to computational models of GRNs, the GRN model for cellular growth studied in this work is described in greater detail in Section III. The evolutionary
algorithm used for evolving the developmental model, the fitness function of the evolution and a number of experimental setups for studying the role of redundancy are provided in Section IV. Results from the different experimental setups are presented and analyzed in Section V. Section VI provides a summary and a conclusion of the paper.

II. REDUNDANCY DURING THE EVOLUTIONARY PROCESS

In this contribution, we aim at achieving a better understanding of the ways in which the evolutionary process succeeds in building regulatory systems. In particular, we want to shed some light on the role of redundancy during this process by analyzing the evolution of simple models of regulatory systems in computer simulations.

Numerous definitions for redundancy have been proposed in the literature both in an engineering as well as a biological context. Here we analyze redundancy during the evolutionary design process as opposed to during the operation time or lifetime. In designing engineering systems using direct redundancy, we usually duplicate system components to increase robustness and fault-tolerance, i.e., the additional components are only active once the working components fail. These components are redundant in the sense of “not being used” during normal operation and usually do not play a key role during design. They are most likely added to the system after the major design phases have been concluded. In biology, gene duplication plays a very important role during evolution for acquiring new genetic raw materials that can potentially lead to evolutionary innovation [25]. In the first step, gene duplication leads to genetic redundancy, because two segments of genes now encode the same functionality. Therefore, it is reasonable to say that genetic redundancy possibly constitutes the first step toward evolutionary innovation. In biology, genetic redundancy resulting from gene duplications has three possible fates: (a) neo-functionalization, i.e., genes assume a new functionality which is preserved by natural selection; (b) non-functionalization, i.e., genes become pseudogenes, and (c) the original and the duplicated genes assume overlapping functionalities. Recently, it has been suggested that bacteria can contain a substantial number of pseudogenes for a limited period of time [26]. Therefore, it seems that genetic redundancy has a limited time window within which it can be turned into evolutionary innovation. Lynch and Connery estimated the average time window for a gene duplication to be about 4 million years [27].

So far we have mainly focused on redundancy as a necessity for the evolutionary process to have genetic materials that can assume new functions, i.e., evolutionary innovations. However, redundancy has also been believed to be a means for providing organisms with mutational robustness in particular for small population sizes [28]. As mentioned in the Introduction, it becomes evident that redundancy plays different roles during the evolutionary process.

In order to get a better understanding of these different roles, we introduce in the following measure of redundancy, which is tuned toward the influence of redundancy during the design phase.

Redundant genes are those whose deletion would have no effect on the phenotype. E.g. genes can express certain proteins, which, however, have no or negligible effect on the phenotype. In this notation, most gene duplications lead to redundancy:

\[ R = \frac{N_R}{N}, \]

where \( N_R \) denotes the number of redundant genes in the whole genome containing \( N \) genes.

III. THE COMPUTATIONAL MODEL FOR MORPHOLOGICAL DEVELOPMENT

A number of computational models have been developed to model biological gene regulatory networks, either for reconstruct biological gene regulation subnetworks using biological data, or to simulate biological signal transduction or development for analyzing fundamental properties such as robustness in systems biology, and for simulating important life phenomena in artificial life (see e.g. the review of de Jong [29]). Artificial embryogeny, an active subfield in artificial life, simulates biological cellular growth and pattern formation starting with one single cell [30], [22], [31], [32].

The morphological development simulated in this work is under the control of a gene regulatory network (GRN) and physical cellular interactions. The morphological development starts with a single cell put in the center of a two-dimensional computational area of size 100 \( \times \) 80.

The GRN is defined by a set of genes, each consisting of a number of regulatory units (RUs) and structural units (SUs). SUs define cellular behaviors, such as cell division, cell death or the production of transcription factors (TFs) for intra- and inter-cellular interactions. Whether the SUs of a gene are expressed is determined by the activity level of the RUs of the gene, refer to Fig. 1. Note that a single or multiple RUs may regulate the expression of a single or multiple SUs and that RUs can be activating (\( RU^+ \)) or repressive (\( RU^- \)). The activation level of RUs is influenced by the TFs that can “bind” to the RU. If the difference between the affinity values of a TF and a RU is smaller than a predefined threshold \( \epsilon \) (in this work \( \epsilon \) is set to 0.2), the TF can bind to the RU to regulate the gene activation. The affinity similarity \( (\gamma_{i,j}) \) between the \( i \)-th TF and \( j \)-th RU is defined by:

\[ \gamma_{i,j} = \max \left( \epsilon - \left| \text{aff}_{i}^{TF} - \text{aff}_{j}^{RU} \right|, 0 \right). \]
If $\gamma_{i,j}$ is greater than zero, then the concentration $c_i$ of the $i$-th TF is checked whether it is above a threshold $\vartheta_j$ defined in the $j$-th RU:
\[
b_{i,j} = \begin{cases} 
\max (c_i - \vartheta_j, 0) & \text{if } \gamma_{i,j} > 0 \\
0 & \text{else}
\end{cases}.
\]
(3)

Thus, the activation level contributed by the $j$-th RU (denoted by $a_j$, $j = 1, \ldots, N$) can be calculated as follows:
\[
a_j = \sum_{i=1}^{M} b_{i,j},
\]
(4)

where $M$ is the number of TFs that bind to the $j$-th RU. Assume the $k$-th gene is regulated by $N$ RUs, the expression level of the gene can be defined by
\[
\alpha = g(c),
\]
(5)

\[
g_k(c) = 100 \sum_{j=1}^{N} h_j a_j (2s_j - 1), \ s_j \in (0, 1).
\]
(6)

$2s_j - 1$ denotes the sign (positive for activating and negative for repressive) of the $j$-th RU and $h_j$ is a parameter representing the strength of the $j$-th RU. If $\alpha_k > 0$, then the $k$-th gene is activated ($\delta_k = 1$) and its corresponding behaviors coded in the SUs are performed.

An SU that produces a TF (SU$_{TF}$) also encodes all parameters related to the TF, such as the affinity value, the decay rate $D_i^c$, the diffusion rate $D_i^s$, as well as the amount of the TF to be produced. Which TF is produced is defined in terms of the affinity value.

\[
A = h(\alpha),
\]

\[
h_i(\alpha_k) = \begin{cases} 
\beta \left(\frac{2}{\alpha_k + e^{-2\alpha_k}} - 1\right) & \text{if } \alpha_k > 0 \\
0 & \text{otherwise}
\end{cases},
\]
(7)

where $\beta$ and $\alpha$ are both encoded in the SU$_{TF}$.

A TF produced by an SU can be partly internal and partially external. To determine how much of a produced TF is external, a percentage ($p^{ext} \in (0, 1)$) is also encoded in the corresponding gene. Thus, $\Delta c^{ext} = p^{ext} \cdot A_i$, $A_i$ is the amount of external TF to be produced and $\Delta c^{ext} = (1 - p^{ext}) \cdot A_i$ is that of the internal TF.

External TFs are put on four grid points around the center of the cell, which undergo first a diffusion and then a decay process. The internal TFs underlie only a decay process. All internal and external concentrations of TFs are limited to an interval of $[0, 1]$.

In our experiments we put two prediffused, external TFs without decay and diffusion in the computation area. The first TF has a constant gradient in the $x$-direction and the second in $y$-direction.

The SU for cell division encodes the angle of division, indicating where the daughter cell is placed. A cell with an activated SU for cell division dies at the developmental timesteps it is activated. When both cell death and cell division are active at the same developmental step, only cell death is performed. There are two additional SUs for other possible actions, which are not used in this work. As a result, it can happen that some genes perform no action, that is one cause of structural redundancy.

Figure 2 shows a block diagram of the main components of a GRN in one cell, describing the cell dynamics. The cell dynamics can become coupled through external transcription factors, which underlie a diffusion and decay process and are position dependent. The number of TFs involved in gene regulation of the cellular behaviors is defined by the genome and the parameters in the resulting GRN as well. The number of cells also changes during development, though we start with one single cell and two external TFs. The maximum number of cells is limited to 700 cells for reducing computational cost. From a control system point of view, the developmental system is composed of a changing number of nonlinear dynamical sub-systems with a changing number of system states, and the dynamics of the sub-systems are strongly coupled with each other.

IV. Experimental Setup

We use an extended evolution strategy, $(\mu, \lambda)$-ES with elitism for evolving the developmental model, where $\mu$ and $\lambda$ are parent and offspring population size, respectively [33]. In this work, $\mu = 30$, $\lambda = 200$, and 3 elitists are used.

Similar to standard ES, Gaussian mutations are applied to the real-valued parameters in the chromosome. The strategy parameter $\sigma$ is fixed to $\sigma = 10^{-4}$ in this work.

Different to standard ES, genetic variations such as gene duplication, gene transposition and gene deletion are also employed in addition to mutations. Gene duplication randomly copies a sequence of RUs and SUs in the chromosome and then inserts it, again randomly, into the chromosome. In the case of gene transposition or deletion, this randomly picked out sequence of RUs and SUs is moved to another randomly chosen site on the chromosome, or simply removed.

Mutation is performed with a probability one, while gene duplication, gene transposition, and gene deletion is performed with a probability of $p_{dup} = 0.05$, $p_{trans} = 0.02$, and $p_{del} = 0.03$, respectively. Gene duplication, transposition and deletion are exclusive, i.e., only one of them will be performed to the same chromosome in one generation.

The goal of the evolution is to evolve an elongated shape. The individuals should have an approximated width-to-height ratio of $a : b$, we used $a_{max} = 10$, $b_{min} = 60$ and $b_{max} = 80$. So, the fitness function is defined as follows:
\[
f = p_1 - p_2 - \min \left\{ \min \left\{ x^i(1) \right\}, \frac{a_{max}}{2} \right\} \\
+ \max \left\{ \max \left\{ x^i(1) \right\}, \frac{a_{min}}{2} \right\},
\]
(8)

where $x^i$ represents the position of the $i$-th cell and
\[
p_1 = \begin{cases} 
70 + \min_i \left\{ x^i(0) \right\} & \text{if } \min_i \left\{ x^i(0) \right\} < -\frac{b_{max}}{2} \\
-30 & \text{if } -\frac{b_{max}}{2} < \min_i \left\{ x^i(0) \right\} < -\frac{b_{min}}{2} \\
\min_i \left\{ x^i(0) \right\} & \text{otherwise}
\end{cases}
\]
(9)
\[ p_2 = \begin{cases} 
70 + \max_i \{ x_i^i(0) \} & \text{if } \max_i \{ x_i^i(0) \} > \frac{b_{\max}}{2} \\
30 & \text{if } \frac{b_{\max}}{2} > \max_i \{ x_i^i(0) \} > \frac{b_{\min}}{2}.
\end{cases} \]

To achieve a sensible yet computationally tractable size of body morphology, the number of cells \( n_c \) is constrained between 10 and 500. A penalty of \( 600 - n_c \) will be applied if \( n_c < 10 \) and a penalty of \( n_c \) if \( n_c > 500 \). If the cells in the developed morphology are not fully connected, a poor fitness of 50 will be assigned.

During some of the evolutionary runs, all redundant genes found in the chromosome are pruned. A gene is considered as redundant if the deletion of the gene results in no fitness change. It should be pointed out that pruning of redundant genes is different to gene deletion in that deletion of a randomly chosen sequence of RUs and SUs may change the fitness of the individual.

To investigate the influence of redundancy on the performance of evolution, we examined 10 different pruning setups for comparison. The definitions of the different setups are listed in Table I. We performed 15 evolutionary runs with different random seeds for each setup.

**TABLE I**

<table>
<thead>
<tr>
<th>Setup no.</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>setup 1</td>
<td>never prune</td>
</tr>
<tr>
<td>setup 2</td>
<td>prune in generation 500</td>
</tr>
<tr>
<td>setup 3</td>
<td>prune every 100th generation</td>
</tr>
<tr>
<td>setup 4</td>
<td>prune every 10th generation</td>
</tr>
<tr>
<td>setup 5</td>
<td>prune once, when fitness of best individual crosses (-40)</td>
</tr>
<tr>
<td>setup 6</td>
<td>fixed DNA with mutation, without duplication, deletion and transposition using 24 RUs and 8 SUs. The order of the RUs and SUs is predefined, also the type of the SUs.</td>
</tr>
<tr>
<td>setup 7, 8, 9, 10</td>
<td>fixed DNA with mutation and transposition and without duplication and deletion. The number of RUs and SUs is 30, 50, 100, 500 respectively.</td>
</tr>
</tbody>
</table>

Setup 6 is designed for investigating the performance of evolution if compact chromosomes are used. In this setup, the positions of all RUs and SUs and the types of the SUs are predefined. The predefined genome is shown in Figure 4, the structure of one individual that achieved the optimal fitness obtained in this setup is provided in Figure 5.

**V. RESULTS AND ANALYSIS**

The boxplots of the best fitnesses from 15 independent runs for the first 9 setups are given in Figure 6. Note, however, that in setup 10, all 15 runs result in a fitness of 600, which means there are no cells at the end of the development. Therefore,
Fig. 4. A predefined chromosome in setup 6, where the positions of all RUs and SUs, the sign of the RUs and the type of the SUs are fixed.

Fig. 5. The genome and its connections of a good individual (the fitness is optimal) of setup 6. The dots are the genes, the predefined TFs are diamond shaped. The arrows define the activations between the different genes, an activation is represented by a dashed line, an inhibition by a dotted line and the solid lines are both, activations and inhibitions.

the results are excluded from the figure. The detailed fitness profiles are shown in Figure 7 - 15.

Generally, long plateaus with sometimes huge jumps in the fitness can be observed. In all setups (except setup 10) are runs which find a good or optimal solution very fast, some runs with large jumps which find a good late and some runs which fail. The number of runs that find a good solution differs between the setups and is therefore analyzed in the following.

The results of setups 1 to 5 suggest that more frequent pruning leads to a worse performance. In addition, we notice that setups 1, 2 and 5 perform comparably well, which suggests that pruning of redundant genes in a late stage of evolution, or when the evolution is already more or less close to the optimal solution, will not degrade the evolutionary performance. Basically, this means that no genetic “raw material” is needed anywhere in later generations.

On the other hand, the results from setup 3 (pruning every 100th gen.), which are worse than those from setups 1, 2 and 5 (yet not statistically significant), indicate that more frequent pruning tends to worsen the performance of the EA. The results of setup 4 (pruning every 10th gen.), which are
Fig. 9. Fitness setup 3

Fig. 10. Fitness setup 4. The fitness of setup 4 run 10 is always 600 and not displayed here.

Fig. 11. Fitness setup 5

Fig. 12. Fitness setup 6

Fig. 13. Fitness setup 7

Fig. 14. Fitness setup 8
significant decrease in the performance of the evolutionary runs if pruning is carried out frequently during the generations. We also see that individuals with short genomes of a fixed length - which would theoretically be sufficient to reach high quality solutions - show significantly lower performance than individuals with redundant genomes of a variable length.

ACKNOWLEDGEMENT

We thank T. Steiner for inspiring discussions. This work was supported by the Honda Research Institute Europe.

REFERENCES


VI. DISCUSSION AND CONCLUSION

In this paper, we have analyzed the role of redundancy during evolution in a simplified computational model for the development of a cellular elongated artificial organism with GRNs.

In a first set of experiments, we limited the redundancy of the different genomes by pruning all redundant genes in a variety of setups. Statistical results show that there is a


