Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Quantitative Analysis of Redundancy in Evolution of Developmental Systems

Lisa Schramm, Yaochu Jin, Bernhard Sendhoff

2012

Preprint:

This is an accepted article published in IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology. The final authenticated version is available online at: https://doi.org/[DOI not available]

Quantitative Analysis of Redundancy in Evolution of Developmental Systems

Lisa Schramm
Technische Universität Darmstadt
Darmstadt, Germany.
lschramm@rtr.tu-darmstadt.de.

Yaochu Jin
Department of Computing
University of Surrey, Guildford, UK.
yaochu.jin@surrey.ac.uk.

Bernhard Sendhoff
Honda Research Institute Europe
Offenbach, Germany.
Bernhard.Sendhoff@honda-ri.de

***Abstract*—Redundancy is believed to play a key role in robustness and evolvability of biological systems. This paper investigates the influence of redundancy on the evolutionary performance of a gene regulatory network governing a cellular growth process. Extensive simulation results suggest that, for the developmental model studied in this work, maintaining sufficient redundancy helps to improve the ability of the evolutionary algorithm to achieve better performance. To examine the change of redundancy during the evolutionary process and its relationship to evolutionary performance, we propose a quantitative definition for measuring different aspects of redundancy, namely, structural redundancy, functional redundancy and functional proximity. Our results show that evolution attempts to increase the functional redundancy after pruning of redundant genes if the evolution is under a larger selection pressure. It is also interesting to notice that an increase in functional proximity enhances the evolutionary performance.**

***Index Terms*—redundancy, gene regulatory network, development**

I. Introduction

Computational modeling of evolutionary development has received increasing attention over the last couple of years. Several models have been suggested in the literature (e.g. [1], [2], [3]),which differ in their levels of biological abstraction and details of the physical simulations. Besides serving as simulation tools for biological *in silico* experiments in particular in the context of computational systems biology [4], one additional important motivation for many of these models has been to contribute towards addressing major challenges in evolutionary computation, most notably scalability. Although interesting results have been achieved, e.g. [5], none of the models have lived up to the initial high expectations. Almost all computational models for evolutionary development represent the dynamics of a process, which after a certain time, results in a desired phenotype. Therefore, one obvious drawback of evolutionary developmental systems is that they add complexity to the evolutionary search process, which is even with a direct representation quantitatively only understood for simple problems, i.e., unimodal problems like high-dimensional sphere or ridge functions. Even though a thorough theoretical analysis is well beyond our current capabilities, a better empirical understanding would already help to shed some light on the effect of developmental representations on the search process.

In this paper, we will analyze the role that redundancy plays in the evolution of developmental systems. Most developmental systems are highly redundant and in this paper we will answer the question whether this redundancy is necessary for the evolutionary search process or whether it is merely a result of the way most developmental representations are organized. In evolutionary biology, it has been argued that genetic redundancy is one of the main mechanisms that contribute substantially to mutational robustness [6], which in turn is a pre-requisite for evolutionary innovation. At the same time, computational experiments have shown that there is an inherent trade-off between redundancy and evolvability for redundant genotype-phenotype mappings [7]. Such a trade-off is confirmed from a slightly different perspective by presenting the evidence that degeneracy, a partial redundancy, is a fundamental source of both robustness and evolvability [8].

A concept that is closely related to redundancy and robustness is neutrality.The relationship between neutrality and robustness has also been widely studied in evolutionary computation.Yu and Miller analyzed different problems with different types of neutrality and found that redundancy can, but must not be beneficial for evolution depending on the implementation [9]. Miller and Smith found that “junk” genes increase the evolvability in a graph-based Cartesian genetic programming system [10]. Banzhaf proposed a model using a genotype-phenotype mapping with neutrality and found that neutrality allows the system to work more flexibly [11].

The developmental model that we will use for the experiments in this paper is conceptually based on the one proposed by Eggenberger Hotz et al. [2] and has been applied successfully to structural design [12], development of primitive nervous systems [13], and body plans [14]. First analysis of redundancy for the developmental model has been presented in [15].

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 a few concepts related to redundancy that we will analyze in this paper.

The work was conducted while Y. Jin was with the Honda Research Institute Europe.

Furthermore, we will propose a new measure that aims at quantifying redundancy in developmental representations in an evolutionary search context. So far redundancy has always been defined as a yes or no measure, i.e. a system component is either redundant or not. The basic idea behind the new measure is to express, how "far away" a redundant system component is from becoming a functional subsystem in the course of evolution. This way we aim to be able to distinguish between a constructive organization of redundancy. i.e., one that is useful to the search process, and a type of redundancy that is unlikely to eventually lead to evolutionary innovation, i.e., to genes that take up a new functionality.

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 experimental set-up for studying the role of redundancy are provided in Section IV. Results 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 a better understanding of the role of redundancy in the evolution of computational developmental systems. Numerous definitions for redundancy have been proposed in the literature both in an engineering and computer science as well as a biological context. In biology, gene duplication plays a very important role during evolution for acquiring new genetic raw materials that can potentially lead to evolutionary innovation [16].

In order to get a better understanding of genetic redundancy we introduce in the following three different measures related to redundancy, which are tuned toward the influence of redundancy during the design phase.

- **Structural Redundancy**
  Structural redundancy R_S denotes the disconnectedness or the non-functionality of a gene. In this notation, pseudogenes would be structurally redundant:

$$R_S = \frac{N_{R_S}}{N}, \tag{1}$$

  where N_{R_S} denotes the number of structurally redundant genes in the whole genome containing N genes.

- **Functional Redundancy**
  Functionally redundant genes are those whose deletion would have no effect on the phenotype. Thus structurally redundant genes are also functionally redundant but not necessarily vice-versa. 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 functional redundancy:

$$R_F = \frac{N_{R_F}}{N}, \quad R_F \geq R_S, \tag{2}$$

  where N_{R_F} denotes the number of functionally redundant genes in the whole genome containing N genes.

- **Functional Proximity**
  All redundancy measures are generally discrete: a gene is either functionally redundant or not. However, for the evolutionary process it is interesting to analyze how far away redundant genes are from assuming functionality, e.g., on average how many mutations or how much time is needed in order for evolution to either turn the genetic redundancy into innovation or to get rid of the genetic materials that have turned into pseudogenes. *Functional Proximity* measures the transition probability from a functionally redundant gene to a functional gene under mutations. If we denote functional genes by g and functionally redundant genes by $g_{R_F} \in G_{R_F}$, then we can define the transition probability $P_\Theta(g_{R_F} \rightarrow g)$ through a mutation parametrized by Θ and *Functional Proximity* FP by

$$fp = P_\Theta(g_{R_F} \rightarrow g) \tag{3}$$

$$FP = \frac{1}{N_{R_F}} \sum_{g_{R_F} \in G_{R_F}} P_\Theta(g_{R_F} \rightarrow g) \tag{4}$$

III. The Computational Model for Morphological Development

A number of computational models have been developed to model biological gene regulatory networks [17], either for re-constructing 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. Artificial embryogeny, an active subfield in artificial life, simulates biological cellular growth and pattern formation starting with one single cell [18], [2], [3].

The morphological development simulated in this work is under the control of a gene regulatory network (GRN) and physical cellular interactions, refer to [14] for more details. The morphological development starts with a single cell put in the center of a 2D computational area of size 100×80.

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 ϵ (in this work ϵ 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:

$$\gamma_{i,j} = \max\left(\epsilon - \left|\text{aff}_i^{TF} - \text{aff}_j^{RU}\right|, 0\right). \tag{5}$$

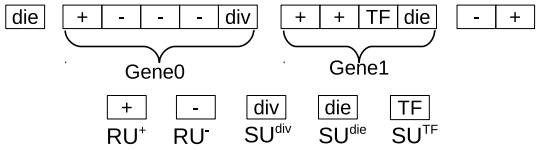


Fig. 1: An example chromosome for the development. The first gene (gene 0) starts at the first RU of the genome. Each SU-RU changeover defines a boundary between two genes.

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 ϑ_j defined in the j-th RU:

$$b_{i,j} = \begin{cases} \max\left(c_i - \vartheta_j,\, 0\right) & \text{if } \gamma_{i,j} > 0 \\ 0 & \text{else} \end{cases}. \tag{6}$$

Thus, the activation level contributed by the j-th RU (denoted by $a_j, j = 1, ..., N$) can be calculated as follows:

$$a_j = \sum_{i=1}^{M} b_{i,j}, \tag{7}$$

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

$$\boldsymbol{\alpha} = \boldsymbol{g}(\boldsymbol{c}), \tag{8}$$

$$g_k(\boldsymbol{c}) = 100 \sum_{j=1}^{N} l_j a_j (2s_j - 1),\ s_j \in (0, 1). \tag{9}$$

$2s_j - 1$ denotes the sign (positive for activating and negative for repressive) of the j-th RU and l_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^f, as well as the amount of the TF$_i$ to be produced. Which TF$_i$ is produced is defined in terms of the affinity value.

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{h}(\boldsymbol{\alpha}), \\ h_i(\alpha_k) &= \begin{cases} \beta\left(\frac{2}{1+e^{-20 \cdot f \cdot \alpha_k}} - 1\right) & \text{if } \alpha_k > 0 \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{10}$$

where f and β are both encoded in the SU^{TF}.

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

$$\text{Diffusion:} \quad \mathbf{u}_i^*(t) = \mathbf{u}_i(t-1) + 0.1 \cdot D_i^f \cdot (\mathbf{G} \cdot \mathbf{u}_i(t-1)), \tag{11}$$

$$\text{Decay:} \quad \mathbf{u}_i(t) = ((1 - 0.1 \cdot D_i^c)\, \mathbf{u}_i^*(t)), \tag{12}$$

where $\mathbf{u}_i$ is a vector of the concentrations of the i-th TF at all grid points and the matrix $\mathbf{G}$ defines which grid points are adjoining. The internal TFs underlie only a decay process:

$$c_i^{\text{int}}(t) = (1 - 0.1 \cdot D_i^c)\, c_i^{\text{int}}(t-1). \tag{13}$$

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 death dies at the developmental timestep 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.

IV. Experimental Setup

We use an evolutionary algorithm that is formally similar to an evolutionary strategy, (μ, λ)-ES with elitism for evolving the developmental model, where μ and λ are parent and offspring population size, respectively. In this work, $\mu = 30$, $\lambda = 200$, and 3 elitists are adopted.

Similar to standard ES, Gaussian mutations are applied to the real-valued parameters in the chromosome. 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 benchmark goal of the evolution is to develop an elongated shape. The individuals should have an approximated width-to-height ratio of $a : b$, we used $a_{max} = 10$, $b_{min} = 60$

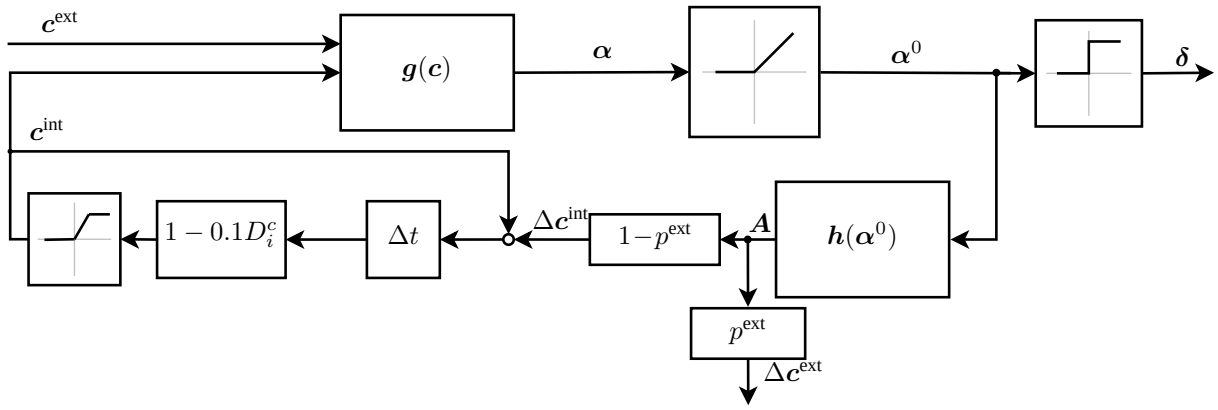


Fig. 2: Block diagram of the model of a single cell.

and $b_{max} = 80$. So, the fitness function is defined as follows:

$$f = p_1 - p_2 - \min\left\{\min_i\left\{\boldsymbol{x}^i(1)\right\}, -\frac{a_{max}}{2}\right\} + \max\left\{\max_i\left\{\boldsymbol{x}^i(1)\right\}, \frac{a_{max}}{2}\right\}, \quad (14)$$

where $\boldsymbol{x}^i$ represents the position of the i-th cell and

$$p_1 = \begin{cases} 70 + \min_i\left\{\boldsymbol{x}^i(0)\right\} & \text{if } \min_i\left\{\boldsymbol{x}^i(0)\right\} < -\frac{b_{max}}{2} \\ -30 & \text{if} -\frac{b_{max}}{2} < \min_i\left\{\boldsymbol{x}^i(0)\right\} < -\frac{b_{min}}{2} \\ \min_i\left\{\boldsymbol{x}^i(0)\right\} & \text{otherwise} \end{cases}$$

and

$$p_2 = \begin{cases} 70 + \max_i\left\{\boldsymbol{x}^i(0)\right\} & \text{if } \max_i\left\{\boldsymbol{x}^i(0)\right\} > \frac{b_{max}}{2} \\ 30 & \text{if } \frac{b_{max}}{2} > \max_i\left\{\boldsymbol{x}^i(0)\right\} > \frac{b_{min}}{2} \\ \max_i\left\{\boldsymbol{x}^i(0)\right\} & \text{otherwise} \end{cases}.$$

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 functionally redundant genes found in the chromosome are pruned. A gene is considered as functionally redundant if the deletion of the gene results in no fitness change. It should be pointed out that pruning of functionally 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 ten different pruning setups for comparison. The definitions of the different setups are listed in Table I. In order to get statistically sound results, we performed 15 evolutionary runs with different random seeds for each setup. 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 structure of one individual that achieved the optimal fitness obtained in this setup is provided in Fig. 3.

TABLE I: Definitions of the different setups

Setup no.	Specification
setup 1	never prune
setup 2	prune in generation 500
setup 3	prune every 100th generation
setup 4	prune every 10th generation
setup 5	prune once, when fitness of best individual crosses -40
setup 6	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.
setup 7, 8, 9, 10	fixed DNA with mutation and transposition and without duplication and deletion. The number of RUs and SUs is 30, 50, 100, 500 respectively.

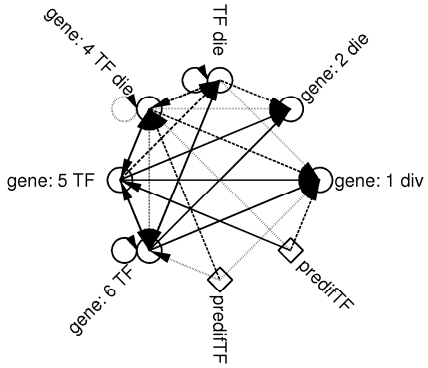


Fig. 3: The genome and its connections of a good individual (the fitness is optimal) of setup 6. The dots are the genes, the predefined TFs are diamond shaped. The arrows define the activations between the different genes, an activation is represented by a dashed line, an inhibition by a dotted line and the solid lines are both, activations and inhibitions.

V. Results and Analysis

A. Fitness

The boxplots of the best fitnesses from 15 independent runs for the first 9 setups are given in Fig. 4. Note, however, that in setup 10, all 15 runs result in a fitness of 600, which means that there are no cells at the end of the development. Therefore, the results are excluded from the figure.

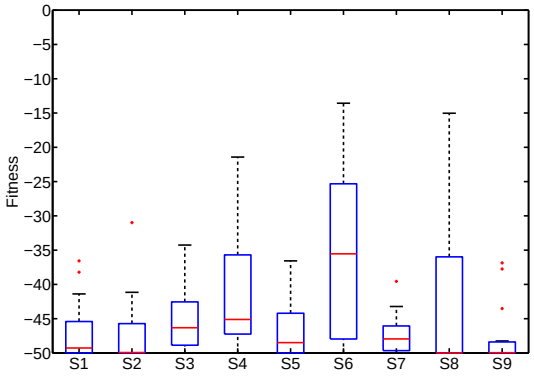


Fig. 4: The boxplots of the best fitness from 15 independent runs of setup 1 to 9.

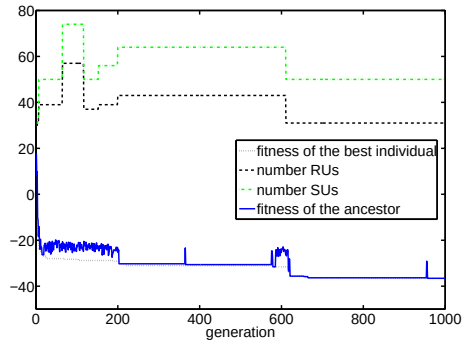


Fig. 5: Number of RUs and SUs and fitness of the ancestors of the best individual from setup 1, run 1.

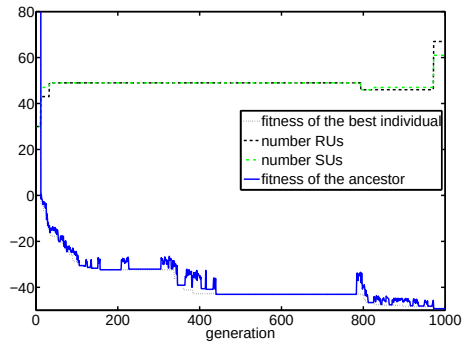


Fig. 6: Number of RUs and SUs and fitness of the ancestors of the best individual from setup 1 run 5.

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 generations), 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 generations), which are significantly worse than those in setups 1, 2 and 5, confirm that continuous pruning of the redundant genes leads to much worse performance.

It is worth discussing the results of setup 6. Although this setup has the fewest parameters to be optimized, only one of the 15 runs converges to the optimal fitness. This indicates that for a representation that does not allow functional redundancy, the evolution has a big difficulty in finding the optimal solution, even if an optimal solution exists (as shown in Fig. 3). This result also supports the hypothesis that redundancy improves evolvability.

A common belief in evolutionary computation, where direct coding is often used, is that the performance of evolutionary algorithms does not scale well with the search dimension. The results from setup 9 show surprisingly that this belief might not be correct for developmental systems. However, it should be noted that the extremely poor results in setup 10 (500 RUs and SUs), in which none of the runs have been successful, indicate that there is a certain upper bound of the search space in which evolution can no longer work properly.

To examine the possible correlation between the genetic variations and the changes in fitness, we analyze the evolutionary 'history' of the best individual in the last generation of run 1 and run 5 of setup 1. Fig. 5 and Fig. 6 show the fitnesses of the best individual of each generation and of the ancestors of the best individual of the last generation, together with the number of RUs and SUs in the individual. In run 1, refer to Fig. 5, the increase in fitness at about generation 600 correlates with a change in the number of RUs and SUs, where a gene deletion is performed. In run 5, refer to Fig. 6, a few 'jumps' in fitness correlate to a gene duplication or deletion (in generations 800 and 950). However, the increase in fitness between generations 300 and 400 has not resulted from duplication or deletion. In other words, this change in fitness may be a result of gene transposition or mutation. Note, however, that the ancestor of the best individual is often not the best individual before a jump in fitness.

B. Structural and Functional Redundancy

To quantitatively analyze the role of redundancy in evolution, we computed the structural and functional redundancy defined in (1) and (2) based on the results from the first 10 runs of setup 1. The number of structurally redundant genes N_{R_S} counts the genes that can never be activated or have no functionality, e.g., those genes have no SUs for TFs, cell division or cell death. Genes that are functionally redundant are detected by comparing the fitness values with and without a gene. If the deletion of the gene does not influence the fitness, then the gene is functionally redundant. We computed both structural and functional redundancy for the ancestors of the best individual in every 10th generation, see Fig. 7. From the figure, no clear tendency in the profile of the redundancy

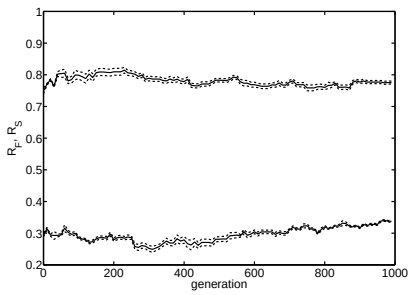


Fig. 7: Mean value and variance of R_S and R_F of the first 10 runs of setup 1 ($R_F > R_S$).

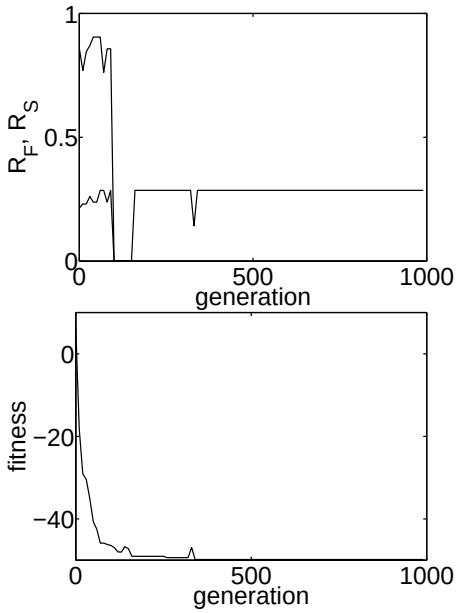


Fig. 8: R_S, R_F, $R_S \leq R_F$ (top) and fitness curve (bottom) of run 1 after pruning in generation 100.

measures can be observed. One conservative observation we can make is that the value of R_S and R_F stays within a certain range: $20\% \leq R_S \leq 40\%$ and $70\% \leq R_F \leq 90\%$, with a few exceptions. There is also no significant difference in the profiles of the redundancy measures between the successful and the nonsuccessful runs. Given the hypotheses that redundancy improves evolvability, we expected an increase in functional redundancy during evolution, which is not confirmed by the results shown in Fig. 7. One explanation might be that the functional redundancy of the individuals in this run is already very high from the beginning, which is sufficient for creating innovative solutions. Thus, a higher value of functional redundancy is no longer necessary.

In order to test this hypothesis, we performed additional experiments, where we prune redundant genes in generation 100. Then we computed the structural and functional redundancy for the remaining generations. Figures 8 and 9 present the number of redundant genes during the evolution and the fitness values of the related individuals. From the figures, we see that

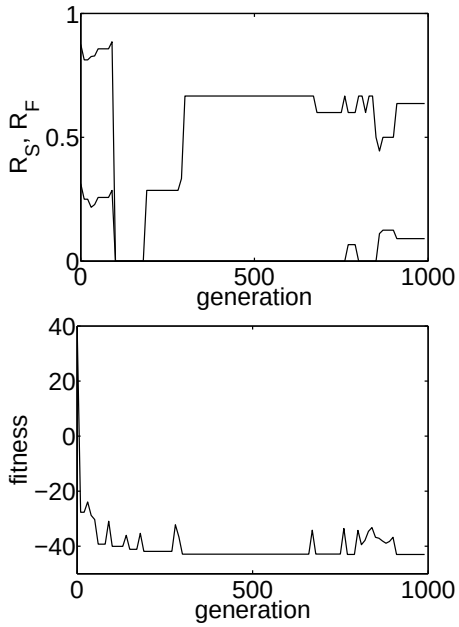


Fig. 9: R_S, R_F , $R_S \leq R_F$ (top) and fitness curve (bottom) of run 2 after pruning in generation 100.

the first redundant genes appear between generations 50 and 100 after the pruning. In the first run, the fitness converges very fast to the optimum. Meanwhile, we see that the structural and functional redundancy keep constant after generation 350. This is reasonable since the best solution has already been found and there is no selection pressure for a better fitness. In this case, additional redundancy is not needed. In the second run, in contrast to the results in the first run, the functional redundancy first rises to 30% and then increases further to nearly 70% which is in the range of Fig. 7. This implies that the evolution is under pressure to increase redundancy since the quality of the population is still low (respectively the fitness is high). Note, however, that the statistical significance of the above observations remains to be verified, since they are based only on two independent runs.

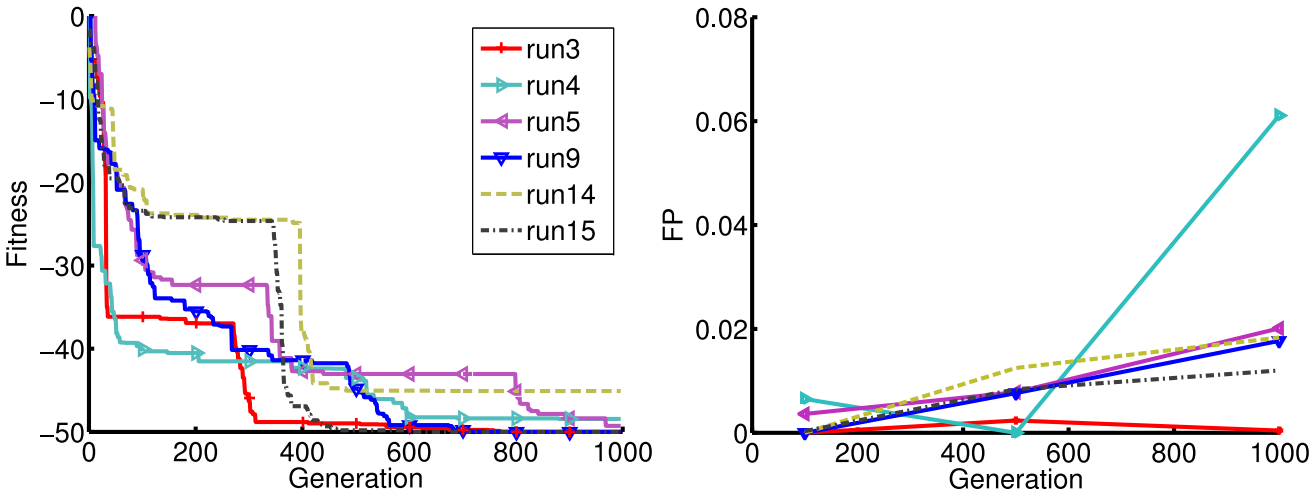
C. Functional Proximity

In the following, we analyze the functional proximity (FP) of different individuals taken from different stages of the evolutionary process. The best individuals in an early, middle and late stage of each evolution are taken (generation 100, 500 and 999). For this purpose, we mutate the individuals 200 times and compute their functional redundancy R_F. The functional proximity of gene j can be estimated by

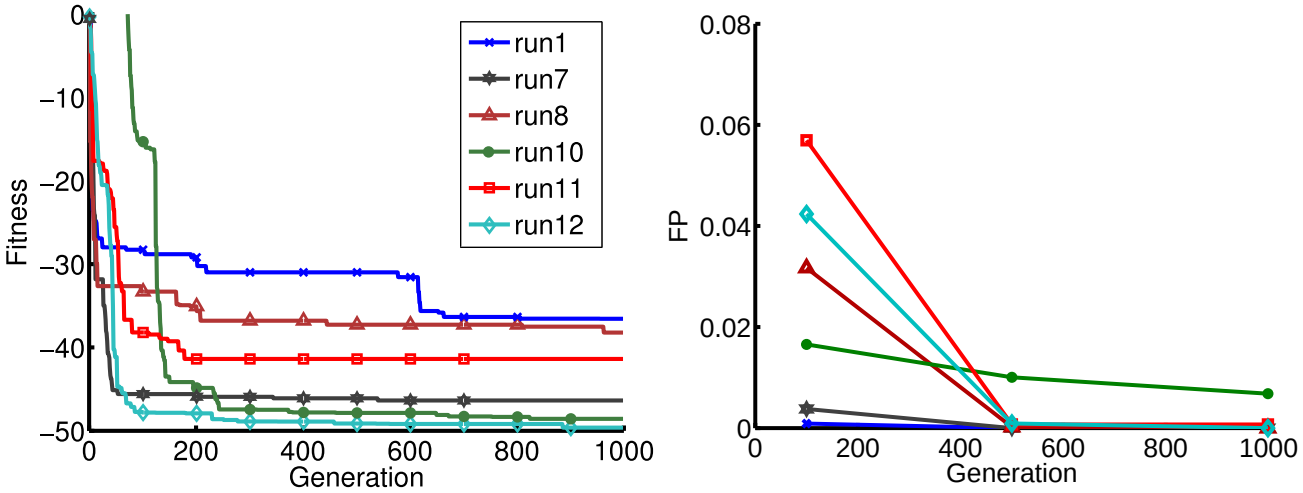
$$\hat{f}_{p,j} = \frac{1}{N_{mut}} \sum_{i=0}^{N_{mut}} R_{F,i,j}, \tag{15}$$

where the number of mutations is $N_{mut} = 200$. The functional proximity $\widehat{FP}$ of the individual is estimated by

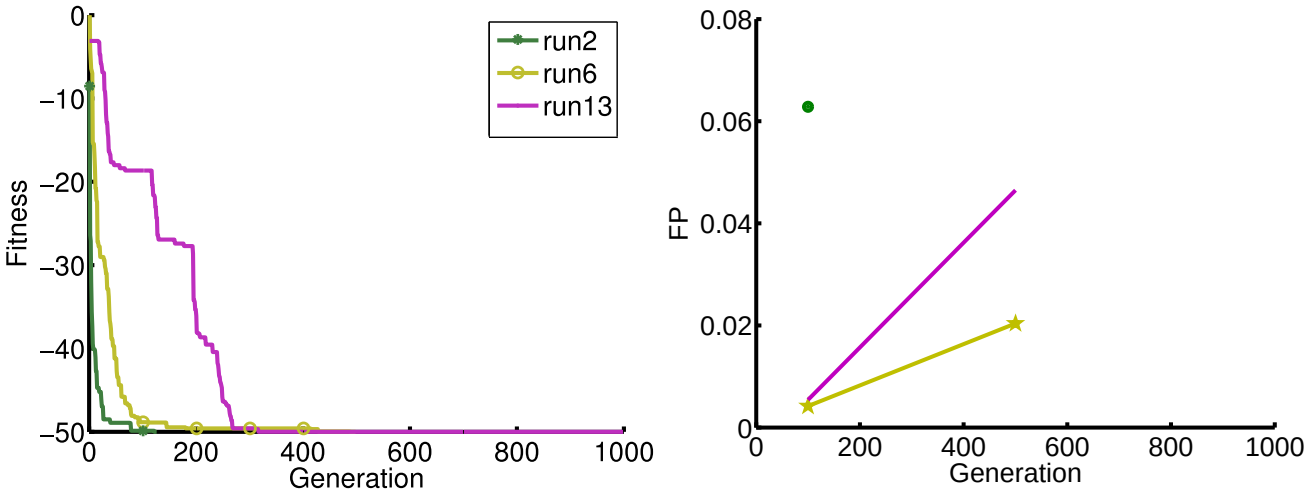
$$\widehat{FP} = \frac{1}{N} \sum_{j=0}^{N} \hat{f}_{p,j}, \tag{16}$$



(a) The evolutionary runs shown in these figures show an increasing fitness with good results at the end.



(b) These runs converged fast or have bad fitness values.



(c) The runs shown here found the global optimum too fast to analyze the influence of the functional proximity.

Fig. 10: The functional proximity is shown on the right side and the associated fitness curves are on the left. The results are assigned to three different categories.

where N is the number of genes of the individual.

Fig. 10 shows the fitness curves and the associated FP values. To analyze the FP, we separated the different evolutionary runs into different categories. The first category (see Fig. 10a) comprises fitness curves with a gradually decreasing fitness and good results at the end. Note that the target is fitness minimization. The curves assigned to the second category (see Fig. 10b) show few and small fitness jumps or bad results. The huge fitness decrease at the beginning is not taken into account. In the third category, we have grouped three runs (see Fig. 10c) that found the global optimum very fast, which makes it difficult to analyze the influence and trend of the FP.

In general, few functionally redundant genes change to a functional gene. The FP of the curves of the first category increases in four of six cases. In the second category, the FP decreases to very small values at generations 500 and 999 in five of the six cases. The optimizations in the third category found the global optimum fast, so it is difficult to analyze a tendency, but the value of FP is high or increases to a high value. This supports the assumption that high, increasing values of FP are beneficial for evolution. The values at generation 100 seem less significant, because the runs in the second category have different values of FP at generation 100 and there is no clear relation to their performance, e.g. run 12 (Fig. 10b) has a high FP and shows good performance at the beginning, but run 11 (Fig. 10b) has a higher FP but the performance is worse.

An exception is run 3 (Fig. 10a), which has a small value of FP in all three test generations but shows good results. The FP of run 4 (Fig. 10a) shows no clear tendency, it first decreases to a very small value and then increases to a very high value and its performance is good. Run 10 (Fig. 10b) shows low performance and a high but decreasing FP. Recall that an evolutionary optimization is a stochastic process and the FP is a probabilistic value. Nevertheless, these results suggest that there is a positive correlation between an increase of functional proximity and a performance increase of the evolution.

VI. Discussion and Conclusion

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

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

In a second set of experiments, we observed the quantitative changes in structural and functional redundancy, which stay at a constant level of about 20% (70%) for most evolutionary runs. From the analysis of two experiments where all functionally redundant genes were removed after generation 100, it seems that the evolutionary process tries to increase redundancy if additional genetic "raw material" can have a selective advantage.

Furthermore, we analyzed a new measure aimed at quantifying redundancy in an evolutionary context termed *functional proximity*. We were able to relate an increase of functional proximity during evolution to more successful evolutionary runs. The data we obtained for functional proximity are not sufficient to give statistically significant results and since the calculation of functional proximity is rather computationally expensive, it would be difficult to gather enough data for developmental systems. However, for future experiments it would be interesting to apply the functional proximity measure to simpler benchmark problems.

Acknowledgment

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

References

[1] K. O. Stanley and R. Miikkulainen, "A taxonomy for artificial embryogeny," *Artificial Life*, vol. 9, no. 2, pp. 93–130, 2003.

[2] P. Eggenberger Hotz, G. Gómez, and R. Pfeifer, "Evolving the morphology of a neural network for controlling a foveating retina - and its test on a real robot," in *Alife VIII*, 2003, pp. 243–251.

[3] M. Joachimczak and B. Wròbel, "Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension," in *Proc. of ECAL*, 2009, pp. 33–40.

[4] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.

[5] J. F. Miller, "Evolving a self-repairing, self-regulating, french flag organism," in *Proceedings of the GECCO*, 2004, pp. 129–139.

[6] A. Wagner, "Distributed robustness versus redundancy as causes of mutational robustness," *BioEssays*, vol. 27, pp. 176–188, 2005.

[7] Y. Jin, R. Gruna, I. Paenke, and B. Sendhoff, "Evolutionary multi-objective optimization of robustness and innovation in redundant genetic representations," in *Proc. of IEEE MCDM*, 2009, pp. 38–45.

[8] J. Whitacre, "Degeneracy: a link between evolvability, robustness and complexity in biological systems," *Theor Biol Med Model*, vol. 7, pp. 6–18, 2010.

[9] T. Yu and J. F. Miller, "Through the interaction of neutral and adaptive mutations, evolutionary search finds a way," *Artificial Life*, vol. 12, no. 4, pp. 525–551, 2006.

[10] J. Miller and S. Smith, "Redundancy and computational efficiency in cartesian genetic programming," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 2, pp. 167–174, 2006.

[11] W. Banzhaf, "Genotype-phenotype-mapping and neutral variation - a case study in genetic programming," in *Proc. of PPSN III*, 2006, pp. 322–332.

[12] T. Steiner, J. Trommler, M. Brenn, Y. Jin, and B. Sendhoff, "Global shape with morphogen gradients and motile polarized cells." in *Proc. of CEC*, 2009, pp. 2225–2232.

[13] Y. Jin, L. Schramm, and B. Sendhoff, "A gene regulatory model for the development of primitive nervous systems," in *Proc. of ICONIP 2008*, 2008, pp. 48–55.

[14] L. Schramm, Y. Jin, and B. Sendhoff, "Emerged coupling of motor control and morphological development in evolution of multi-cellular animats," in *Proc. of ECAL*, 2009, pp. 25–32.

[15] ——, "Redundancy creates opportunity in developmental representations," in *Proc. of IEEE ALIFE 2011*, 2011, 203–210.

[16] A. Woollard, *Wormbook*. http://www.wormbook.org, 2005, ch. Gene duplications and genetic redundancy in *C.elegans*.

[17] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *J Comput Biol*, vol. 9, no. 1, pp. 67–103, 2002.

[18] T. Andersen, R. Newman, and T. Otter, "Shape homeostasis in virtual embryos," *Artificial Life*, vol. 15, no. 2, pp. 161–183, 2009.