Redundancy Creates Opportunity in Developmental Representations

Lisa Schramm, Yaochu Jin, Bernhard Sendhoff

2011

Preprint:

This is an accepted article published in IEEE Symposium on Artificial Life. The final authenticated version is available online at: https://doi.org/[DOI not available]

Redundancy Creates Opportunity in Developmental Representations

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

Yaochu Jin
Department of Computing
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 few 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 their genes and starts with a fertilized cell that develops into a mature organism. Simulating and analyzing biological development can on the one hand shed light on biological processes and improve our understanding of natural systems. On the other hand, developmental models are increasingly being used – often in conjunction with evolutionary algorithms – to improve computational engineering and computational design processes in general.

Developmental models allow the representation of systems as a process instead of parametrizing the final system itself. Although this promises advances with regard to such issues like system scalability, flexibility and robustness, it also poses new and challenging questions, in particular on how to represent the dynamics of growth processes in an evolvable way.

Robustness is one of the most important design principles of biological systems. It can be achieved by a variety of mechanisms [1]. Wagner has suggested 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 showing that degeneracy, i.e. partial redundancy, is a fundamental source for both robustness and evolvability [4].

A concept that is closely related to redundancy and robustness is neutrality [5]. Kimura has been the first to comprehensively analyze and discuss the role of neutrality in biological evolution [6]. He has argued that most allelic variation and substitution is neutral but a lot of mutations are deleterious. This suggests that random genetic drifts may be one of the main driving forces behind 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 principles have increasingly been employed to solve complex engineering problems, such as shape and structural design [10], [11], [12], evolvable hardware [13], [14], controller design [15], [16], and self-organization of swarm robots [17], [18]. Inspirations from biology can bring desirable properties such as scalability, robustness, self-organization, self-repair and sustainability into engineered 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 in [19] and has been applied successfully to structural design [12], development of primitive nervous systems [20], and body plans of artificial organisms [21]. In this work, we aim at a better understanding of the role of redundancy during the evolutionary process by means of *in silico* experiments. The insights gained from these experiments can in a second step either be used for the analysis of biological data or to optimally set up design processes that exploit biological principles for engineering tasks.

Biological evolution has to keep the "raw material" available from which innovations can be made. This is in contrast to a standard engineering approach where material can be supplied from external sources, i.e. if an engineer has a new idea on how to design a system, he can simply add the representation of this new idea to the system description. In biology, any "new idea" must arise from within the system's own representations. Therefore, the two approaches are conceptually different and we expect that such differences have profound consequences.

As we have noted, "redundancy" is closely related to other

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

system properties such as evolvability, stability and neutrality. However, most of these concepts lack a clear and well accepted definition. For example, redundancy has different definitions in different areas. Therefore, in Section II, we provide a 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 for growth. 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 any role during design. They are most likely added to the system after the major design phases have been completed. In biology, gene duplication plays a very important role during evolution for acquiring new genetic raw materials that can potentially lead to evolutionary innovation [22]. In the first step, gene duplication may lead to *genetic redundancy*, because two segments of genes now encode the same functionality. Therefore, it is very likely that genetic redundancy constitutes the first step toward evolutionary innovation. In biology, genetic redundancy resulting from gene duplications has four 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, (c) sub-functionalization, i.e., duplicates of a gene with multiple functions carry reduced, complementary sets of functions, and (d) 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 [23]. 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 [24].

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 [25]. 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 a measure of redundancy in the following, 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:

$$R = \frac{N_R}{N}, \tag{1}$$

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

III. A 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 [26]). Artificial embryogeny, an active subfield in artificial life, simulates biological cellular growth and pattern formation starting with one single cell [27], [19], [28], [29], [30], [31]. Stanley and Miikkulainen develop a taxonomy for artificial embryogeny based on cell fate, targeting, heterochrony, canalization, and complexification [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 100×80. Each cell can die or divide. The cells are not fixed on a grid and underlie physical interactions, i.e. overlapping cells push each other away and cells that do not overlap attract each other with decreasing forces with larger distances.

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^-).

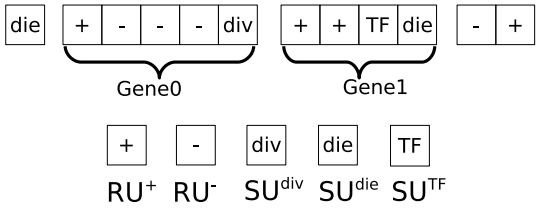

Fig. 1. An example chromosome for the development. The first gene (gene 0) starts at the first RU of the genome. Each SU-RU changeover defines a boundary between two genes.

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 values are encoded in the RUs and the SUs that produce a TF and are, as well as all values in the genome, limited to an interval of $[0, 1]$. The affinity similarity ($\gamma_{i,j}$) between the i-th TF and j-th RU is defined by:

$$\gamma_{i,j} = \max\left(\epsilon - \left|\text{aff}_i^{TF} - \text{aff}_j^{RU}\right|, 0\right). \quad (2)$$

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:

$$b_{i,j} = \begin{cases} \max\left(c_i - \vartheta_j, 0\right) & \text{if } \gamma_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}. \quad (3)$$

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

$$a_j = \sum_{i=1}^{M} b_{i,j}, \quad (4)$$

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

$$\boldsymbol{\alpha} = \boldsymbol{g}(\boldsymbol{c}), \quad (5)$$

$$g_k(\boldsymbol{c}) = 100 \sum_{j=1}^{N} l_j a_j (2s_j - 1),\ s_j \in (0, 1). \quad (6)$$

$2s_j - 1$ 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.

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{h}(\boldsymbol{\alpha}), \\ h_i(\alpha_k) &= \begin{cases} \beta\left(\frac{2}{1+e^{-20 \cdot f \cdot \alpha_k}} - 1\right) & \text{if } \alpha_k > 0 \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \quad (7)$$

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. Note, that the external TFs are computed on a grid but the positions of the cells are continuous and therefore not limited to this grid. 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 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 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, (μ, λ)-ES with elitism for evolving the developmental model, where μ and λ 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 σ 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

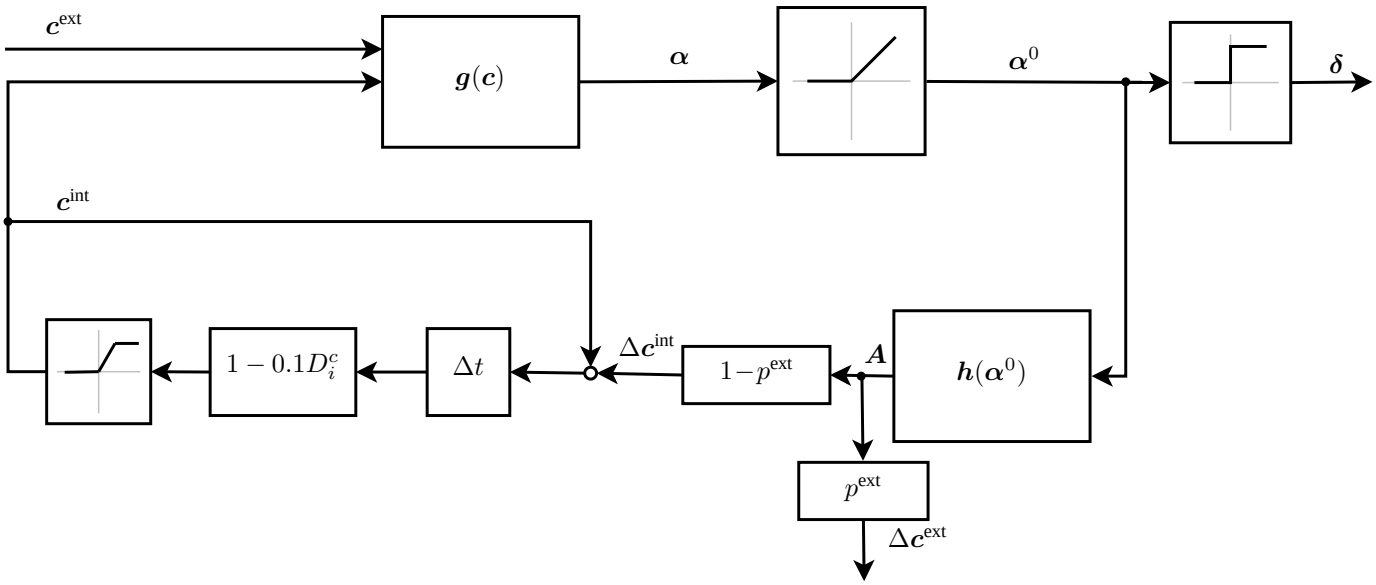

Fig. 2. Block diagram of the model of a single cell.

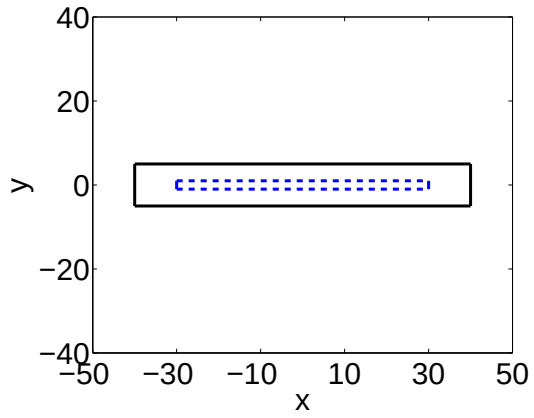

Fig. 3. Optimal shape of the individuals. There should be cells inside the blue, dashed box but not outside the black, solid box.

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 $b_{max} = 80$. The following fitness function is minimized:

$$f = p_1 - p_2 - \min\left\{\min_i\left\{\boldsymbol{x}^i(1)\right\}, -\frac{a_{max}}{2}\right\} + \max\left\{\max_i\left\{\boldsymbol{x}^i(1)\right\}, \frac{a_{max}}{2}\right\}, \tag{8}$$

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

$$p_1 = \begin{cases} 70 - \min_i\left\{\boldsymbol{x}^i(0)\right\} & \text{if } \min_i\left\{\boldsymbol{x}^i(0)\right\} < -\frac{b_{max}}{2} \\ -30 & \text{if} -\frac{b_{max}}{2} < \min_i\left\{\boldsymbol{x}^i(0)\right\} < -\frac{b_{min}}{2} \\ \min_i\left\{\boldsymbol{x}^i(0)\right\} & \text{otherwise} \end{cases} \tag{9}$$

and

$$p_2 = \begin{cases} -70 - \max_i\left\{\boldsymbol{x}^i(0)\right\} & \text{if } \max_i\left\{\boldsymbol{x}^i(0)\right\} > \frac{b_{max}}{2} \\ 30 & \text{if } \frac{b_{max}}{2} > \max_i\left\{\boldsymbol{x}^i(0)\right\} > \frac{b_{min}}{2} \\ \max_i\left\{\boldsymbol{x}^i(0)\right\} & \text{otherwise} \end{cases}. \tag{10}$$

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. Each individual is computed for 15 developmental steps, the computation is aborted if more than 700 cells are reached.

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.

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

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.

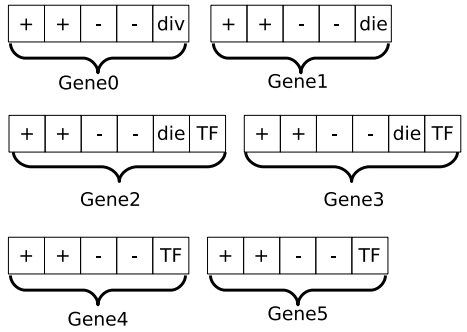

Fig. 4. A predefined chromosome in setup 6, where the positions of all RUs and SUs, the sign of the RUs and the type of the SUs are fixed.

predefined and hand-coded. This setup has the fewest parameters and is defined for comparison because an optimization should be easier the less parameters are to be optimized. 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, the results are excluded from the figure. The detailed fitness profiles are shown in Figure 7 - 15.

From the fitness profiles, we can often observe long plateaus with sometimes large jumps. For all setups (except for setup 10), some runs achieve a good or optimal solution very quickly, some with large jumps in their fitness profile find a good solution in a later stage, and others fail. The number of runs that find a good solution differs among different setups and therefore will be 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 later stage of evolution, or when the evolution is already more or less close

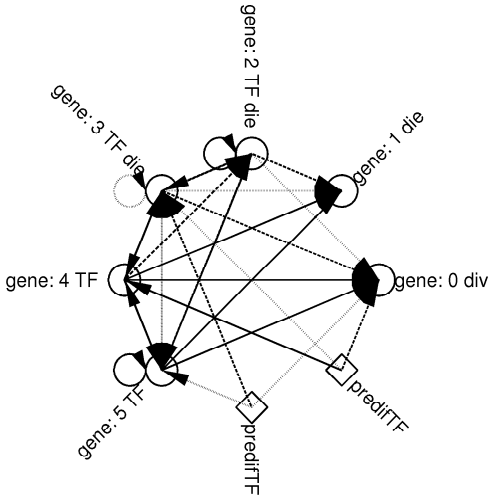

Fig. 5. The genome and its connections of a good individual (the fitness is optimal) of setup 6. The dots are the genes, the predefined TFs are diamond shaped. The arrows define the activations between the different genes, an activation is represented by a dashed line, an inhibition by a dotted line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

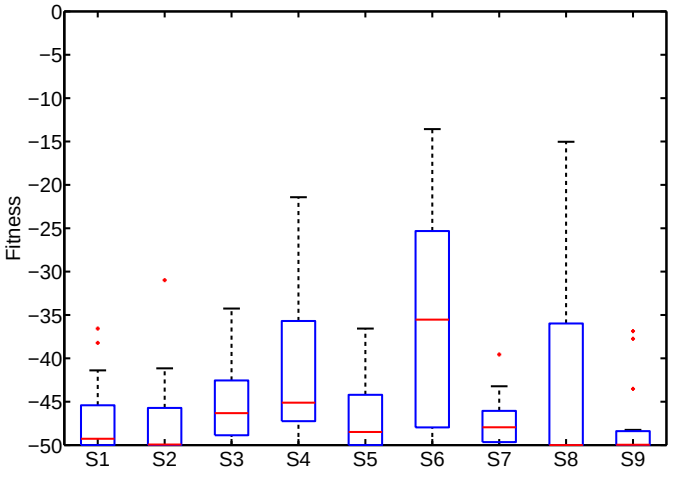

Fig. 6. The boxplots of the best fitness from 15 independent runs of setup 1 to 9.

to the optimal solution, will not degrade the evolutionary performance. Basically, this means that no genetic “raw material” is needed anymore in later generations.

On the other hand, the results from setup 3 (pruning every 100th generation), 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 generation), 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.

We tested the difference in the mean values with the Mann-Whitney U test with a statistical significance of 95% (see [34]). The means of setup 1, 2, 5, 8, and 9 are lower than the ones of

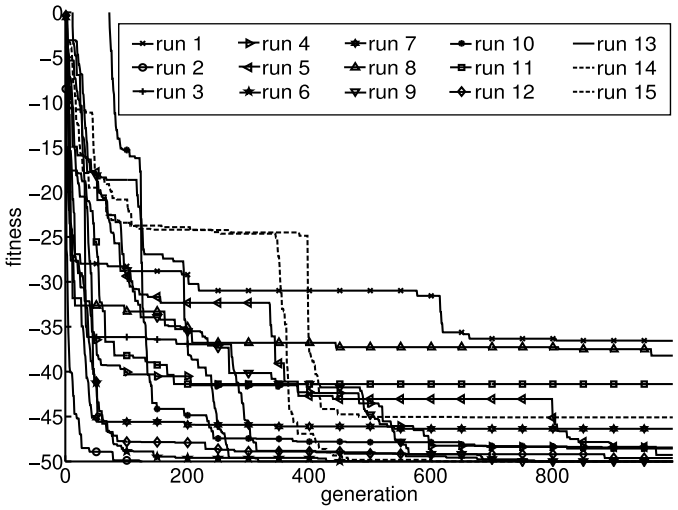

Fig. 7. Fitness setup 1

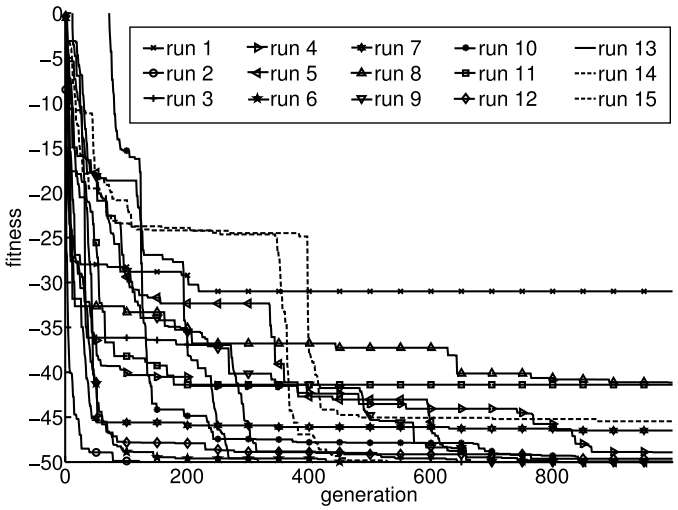

Fig. 8. Fitness setup 2

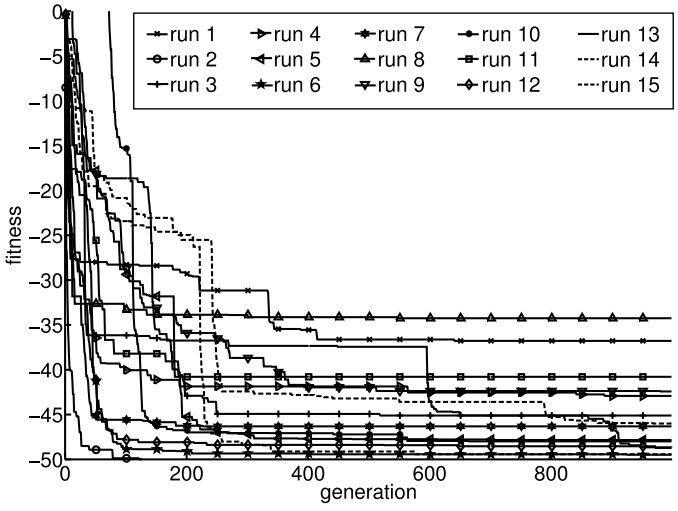

Fig. 9. Fitness setup 3

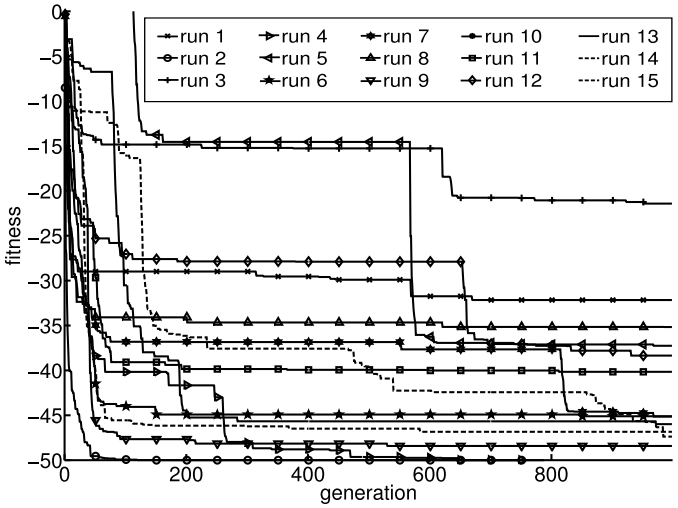

Fig. 10. Fitness setup 4. The fitness of setup 4 run 10 is always 600 and not displayed here.

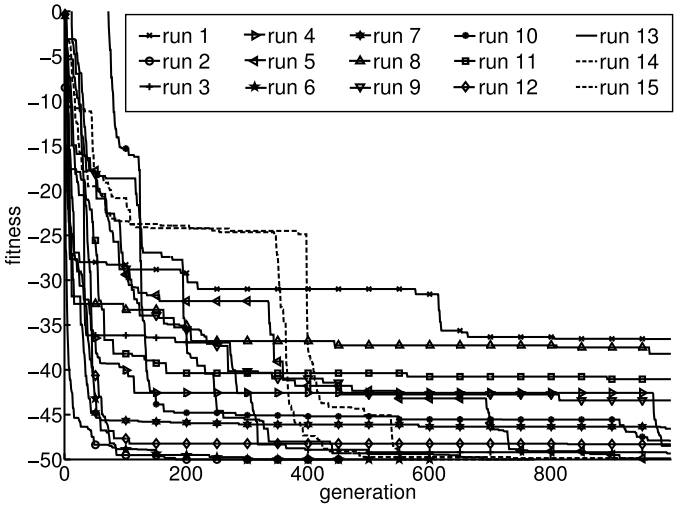

Fig. 11. Fitness setup 5

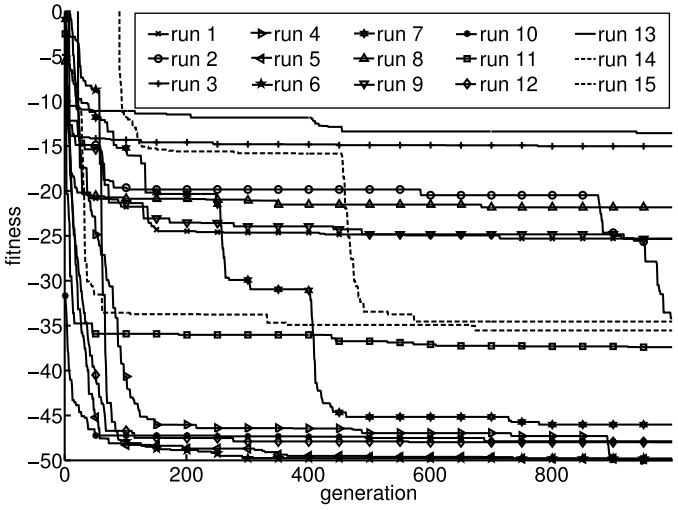

Fig. 12. Fitness setup 6

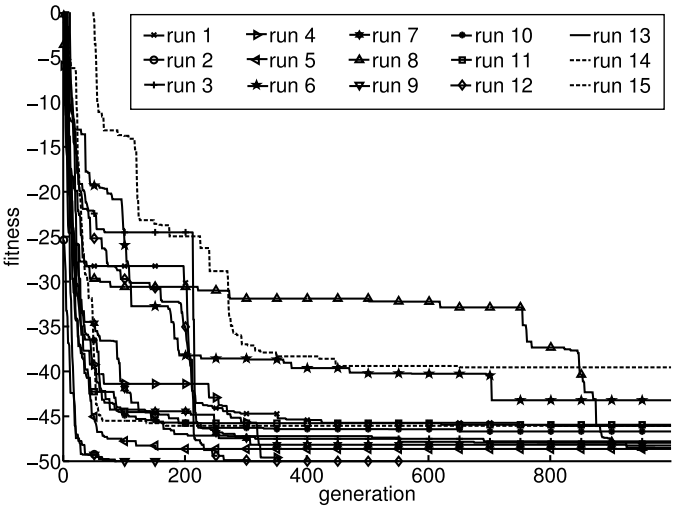

Fig. 13. Fitness setup 7

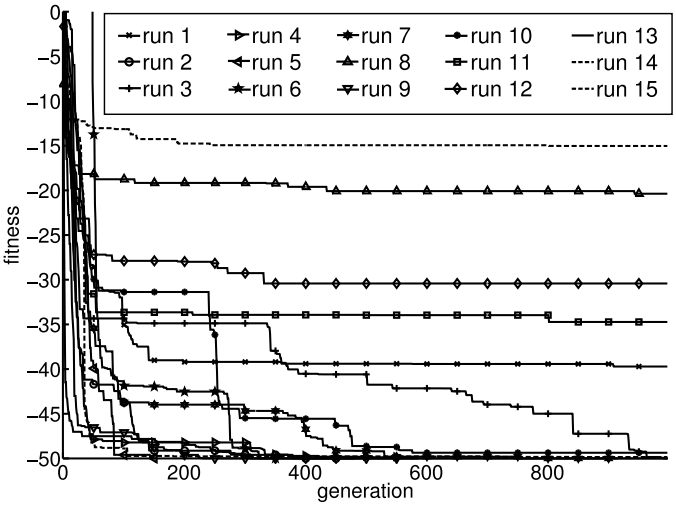

Fig. 14. Fitness setup 8

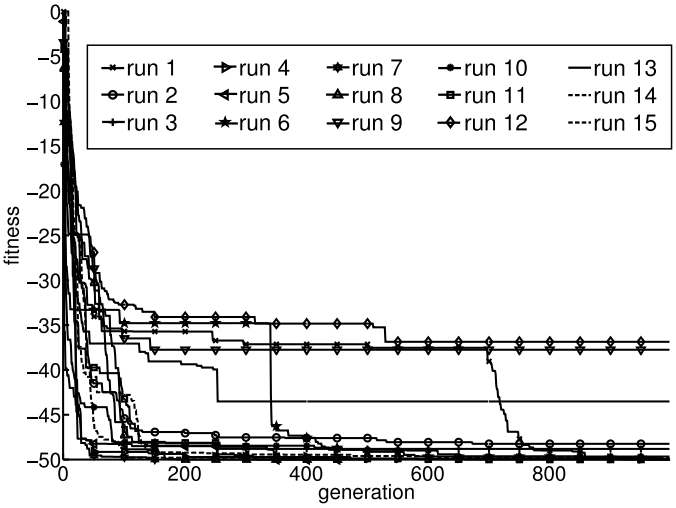

Fig. 15. Fitness setup 9

setup 4 and 6. Additionally setups 3 and 7 are better than setup 6, setup 2 and 8 are better than setup 3. More experiments would be helpful to increase the statistical significance, e.g. the difference in the length of the 25th and 75th percentiles of setups 4, 6 and 8 should become smaller.

Although setup 6 is the setup with the fewest parameters, only one of the 15 runs converges to the optimal fitness. This indicates that for a representation that does not allow redundancy, the evolution has difficulties in finding the optimal solution, even if the optimal solution exists (as shown in Figure 5). 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 above which evolution does not work properly anymore.

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. The development is controlled by a gene regulatory network and the redundancy of its genes is analyzed.

In the experiments, we limited the redundancy of different genomes by pruning all redundant genes in a variety of setups. Statistical analysis shows that there is a significant decrease in the performance of the evolutionary runs if pruning is carried out frequently during the evolution. We also observe 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.

For future work more analyzes of the runs are of course necessary, so we want to measure the percentage of redundant genes during the evolutions. An analysis on how often redundant genes change to functional genes during the evolutions is also important.

Acknowledgement

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

References

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

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

[3] Y. Jin, R. Gruna, I. Paenke, and B. Sendhoff, “Evolutionary multi-objective optimization of robustness and innovation in redundant genetic representations,” in *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, 2009, pp. 38–45.

[4] J. Whitacre, “Degeneracy: a link between evolvability, robustness and complexity in biological systems,” *BMC Journal of Theoretical Biology and Medical Modeling*, 2010, accepted.

[5] E. van Nimwegen, J. Crutchfield, and M. Huynen, "Neutral evolution of mutational robustness," *PNAS*, pp. 9716–9720, 1999.

[6] M. Kimura, *The neutral theory of molecular evolution*. Cambridge University Press, 1983.

[7] M. Ebner, P. Langguth, J. Albert, M. Schakleton, and R. Shipman, "On neutral networks and evolvability," in *Congress on Evolutionary Computation*. IEEE Press, 2001, pp. 27–30.

[8] 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.

[9] W. Banzhaf, "Genotype-phenotype-mapping and neutral variation - a case study in genetic programming," in *Parallel Problem Solving from Nature - PPSN III*, 2006, pp. 322–332.

[10] J. Rieffel and J. Pollack, "Automated assembly as situated development: Using artificial ontogenies to evolve buildable 3-d objects." in *Genetic and Evolutionary Computation Conference*, 2005, pp. 99–106.

[11] T. Steiner, Y. Jin, and B. Sendhoff, "A cellular model for the evolutionary development of lightweight material with an inner structure," in *Proceedings of the GECCO*, 2008, pp. 851–858.

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

[13] G. Tufte, "Gene regulation mechanisms introduced in the evaluation criteria for a hardware cellular development system." in *Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems*, 2006, pp. 137–144.

[14] S. Zhan, J. Miller, and A. Tyrrell, "An evolutionary system using development and artificial gene regulatory networks for electronic circuit design," *BioSystems*, vol. 98, pp. 176–192, 2009.

[15] P. J. Bentley, "Evolving fractal gene regulatory networks for robot control," in *In Proceedings of ECAL 2003*. Springer-Verlag, 2003, pp. 753–762.

[16] T. Taylor, "A genetic regulatory neywork inspired real-time controller for a group of underwater robots," in *8th International Conference on Intelligent Autonomous Systems*, 2004, pp. 403–412.

[17] H. Guo, Y. Meng, and Y. Jin, "A cellular mechanism for multi-robot construction via evolutionary multi-objective optimization of a gene regulatory network," *BioSystems*, vol. 98, pp. 193–203, 2009.

[18] Y. Jin and Y. Meng, "Morphogenetic robotics: A new paradigm for designing self-organizing, self-reconfigurable and self-adaptive robots," in *Morphogenetic Engineering: Toward Programmable Complex Systems*. Springer, 2010, to appear.

[19] 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 *The Eighth International Conference on Artificial Life*. MIT Press, 2003, pp. 243–251.

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

[21] L. Schramm, Y. Jin, and B. Sendhoff, "Emerged coupling of motor control and morphological development in evolution of multi-cellular animats," in *Accepted for publication in Proc. of ECAL*, 2009.

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

[23] S. Abby and V. Daublin, "Comparative genomics and the evolution of prokaryotes," *Trends in Microbiology*, vol. 15, no. 3, pp. 135–141, 2007.

[24] M. Lynch and J. Connery, "The evolutionary fate and consequences of duplicate genes," *Science*, vol. 290, pp. 1151–1155, 2002.

[25] D. Krakauer and J. Plotkin, "Redundancy, antiredundancy, and the robustness of genomes," *Proceedings of the National Academy of Sciences*, vol. 99, no. 3, pp. 1405–1409, 2002.

[26] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67–103, 2002.

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

[28] S. Harding and W. Banzhaf, "Artificial development," in *Organic Computing*. Springer-Verlag Berlin Heidelberg, 2008, pp. 201–219.

[29] M. Joachimczak and B. Wròbel, "Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension," in *Accepted for publication in Proc. of ECAL*, 2009.

[30] R. Doursat, "Facilitating evolutionary innovation by developmental modularity and variability," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO 09)*, 2009, pp. 683–690.

[31] J. F. Knabe, M. J. Schilstra, and C. Nehaniv, "Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information." in *Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems (ALife)*, 2008.

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

[33] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies –a comprehensive introduction," *Natural Computing: an international journal*, vol. 1, no. 1, pp. 3–52, 2002.

[34] J. D. Gibbons, *Nonparametric Statistical Inference*. Marcel Dekker, 1985.