Towards Evolutionary Network Reconstruction Tools for Systems Biology

Thorsten Lenser, Thomas Hinze, Bashar Ibrahim, and Peter Dittrich

Friedrich Schiller University Jena
Bio Systems Analysis Group
Ernst-Abbe-Platz 1–4, D-07743 Jena, Germany
{thlenser,hinze,ibrahim,dittrich}@minet.uni-jena.de

Abstract. Systems biology is the ever-growing field of integrating molecular knowledge about biological organisms into an understanding at the systems level. For this endeavour, automatic network reconstruction tools are urgently needed. In the present contribution, we show how the applicability of evolutionary algorithms to systems biology can be improved by a domain-specific representation and algorithmic extensions, especially a separation of network structure evolution from evolution of kinetic parameters. In a case study, our presented tool is applied to a model of the mitotic spindle checkpoint in the human cell cycle.

1 Introduction

Reverse engineering of biochemical networks, making sense of rapidly growing molecular proteomics data, is a promising and important field at the crossroads of optimisation and model selection. Supplementing human-curated models with automatically generated, data-based models will enhance our understanding of the function of cells as a whole, which is at the core of systems biology.

Evolutionary algorithms (EA) and especially genetic programming (GP) have a long-established history as heuristic optimisation techniques [2,13,15]. Recently, methodologies adapted from this field have been used to evolve artificial biochemical networks, capable of performing arithmetic calculations [7] or specific behaviours such as oscillations and switching [10,17]. Others have used similar techniques to reconstruct metabolic pathways from time series data of chemical species [14]. While these attempts were successful for small networks, they also highlighted the complexity of evolving larger networks.

To expand our capability of evolving networks, improvements on these algorithms have to be investigated. In this contribution, we propose a separation of structural evolution of the network from kinetic parameter evolution, which yields a pronounced increase in the algorithm's fitness performance. Our studies show that this separation helps to prevent premature convergence when evolving networks performing arithmetic calculations. We suppose this happens because parameter fitting after each structural mutation smoothes their effect, which is usually rather strong. In this way, network parameters can adapt to a new topology before this topology is evaluated.

E. Marchiori, J.H. Moore, and J.C. Rajapakse (Eds.): EvoBIO 2007, LNCS 4447, pp. 132–142, 2007.

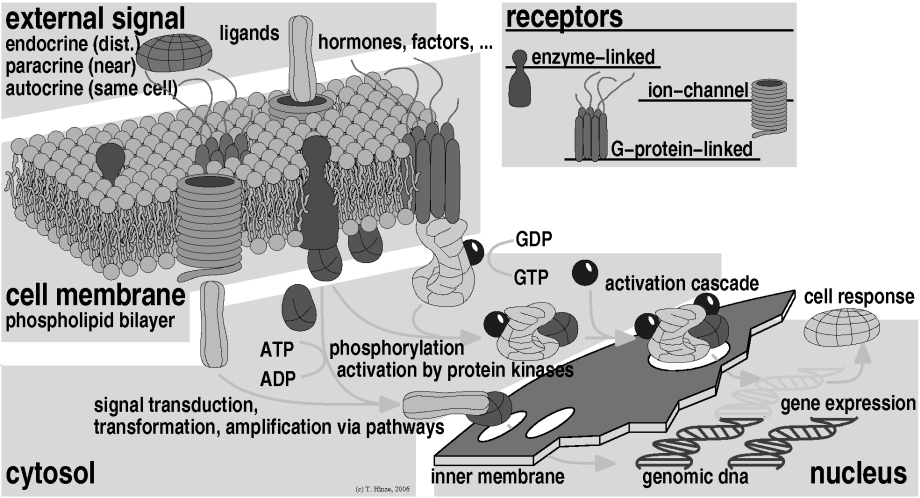

Fig. 1. Biological principle of signalling in eukaryotic cells: from arriving stimuli to specific cell response

Two other ideas are also investigated: the biologically-inspired mutation operator of species duplication, and the use of Akaike's Information Criterion (AIC) as a fitness function to evolve parsimonious models. By using the markup language SBML, the tool described here can work directly on systems biological problems, aiming at applications throughout this growing community.

In a case study, we apply our algorithm to a model of the human mitotic spindle checkpoint. By allowing the algorithm to introduce additional reactions, the performance of the model can be increased in comparison to a mere optimisation of parameters. Although biological plausibility is not considered, the example serves as a proof of concept for further investigations.

2 Modelling and Evolving Biochemical Networks

Biochemical reaction networks found in pro- and eukaryotic cells represent important components of life. Despite their high degree of complexity, they are hierarchically arranged in modular structures of astonishing order. The function of a cell emerges from the interplay of connected reaction processes. Three essential types of biochemical networks can be distinguished: metabolic, cell signalling (CSN), and gene regulatory (GRN) networks [1]. While metabolism consists of coupled enzymatically catalysed reactions supplying energy, CSNs and GRNs perform information processing of external and internal signals [6]. Malfunctions or perturbations within these networks are the cause of many diseases.

We have built a software tool implementing an evolutionary algorithm that evolves artificial biochemical networks performing pre-specified tasks. As a representation format, the systems biology standard SBML [9] is used, the most

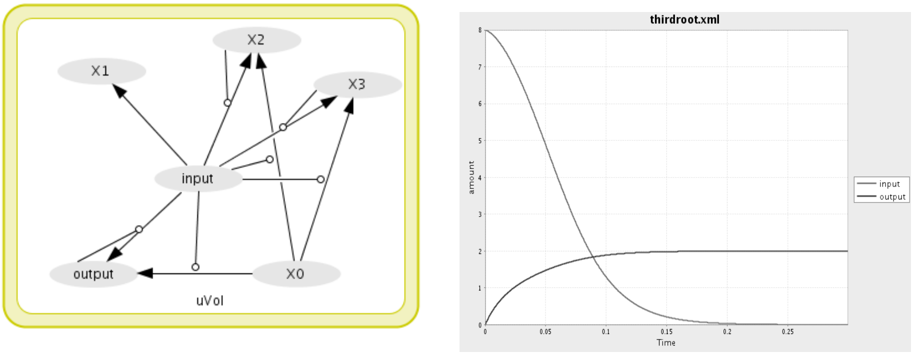

Fig. 2. Example solution and corresponding time series of *input* and *output* species for the third root network, produced using the CellDesigner [11] tool

common interchange format for biochemical models. This provides us with the opportunity to profit from an immense variety of tools developed for the analysis and interpretation of such models. The evolutionary algorithm used here employs eight different mutations:

- Addition / deletion of a species
- Addition / deletion of a reaction
- Connection / removal of an existing species to / from a reaction
- Duplication of a species with all its reactions
- Mutation of a kinetic parameter

While the first six and the last mutation have been used before, we are not aware of work that has used species duplication for the kind of network evolution discussed here. Crossover between networks is possible, but its effects are not part of this work and it has been disabled for all presented experiments.

Fitness evaluation in the algorithm is done by integrating the ODE system resulting from an individual model using the SBML ODE Solver Library [16], a tool designed precisely for that task. The resulting multidimensional time series is then compared to a target, and the weighted quadratic difference

$$f = 1/C \sum_{c=1}^{C} 1/S \sum_{i=1}^{S} 1/m_{c,i} \sum_{j=1}^{N} (x_{c,i}(t_j) - y_{c,i}(t_j))^2,$$

$$\text{with } m_{c,i} = \sum_{j=1}^{T} (x_{c,i}(t_j) + y_{c,i}(t_j))/2,\ i = 1, \dots, S,$$

between the resulting time series x and the target time series y defines the fitness. Here, $i = 1, \dots, S$ runs over the set of evaluated species, and $c = 1, \dots, C$ runs over the fitness cases. Thus, fitness values are minimised, 0 being the absolute lower bound. If a steady state value is regarded as the result of the computation,

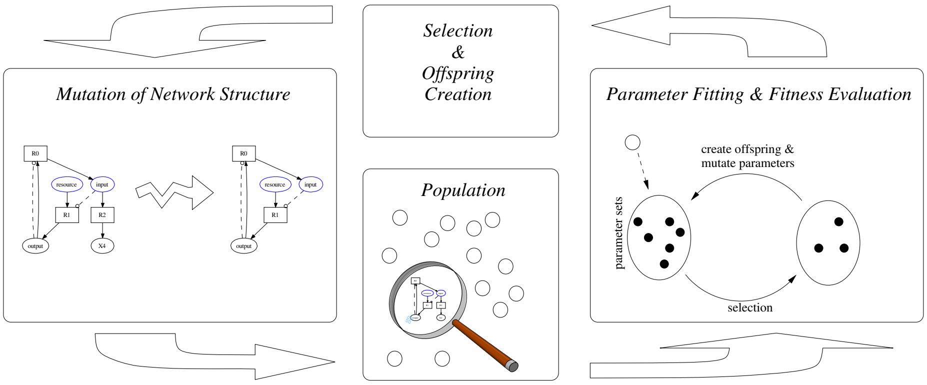

Fig. 3. Outline of the two-level evolutionary algorithm

a constant time series is the target and the first few timesteps are discarded. When Akaike's Information Criterion (see Section 4) is applied, the number of free parameters in the model k (kinetic parameters plus free initial conditions) and the number of data points $n = CTS$ are incorporated and the fitness is modified in the following way:

$$f_{\mathrm{AIC}} = 2k + (n\log(f)) + 2k(k+1)/(n-k-1)$$

In this case, fitness is still minimised but can be negative without lower bound.

Selection is elitistic, with a certain percentage of the population surviving to the next generation, which is filled by mutants of survivors. It is possible to fit kinetic parameter before evaluating the model structure, a technique described in detail in the next paragraph. The software and all data shown in this paper is available from the authors upon request.

3 Separating Structural from Parameter Evolution

The evolution of an artificial network model can be separated into two parts: On the one hand, a set of species and reactions adequate for the task has to be found. On the other hand, the parameters of this model structure have to be optimised. The problem is analogue to model inference, where a dataset is used not only to fit the parameters of a model, but rather to choose a model structure together with a set of parameters. For nonlinear problems, this is still a largely unresolved challenge. Here, we show that a separation of model-structure evolution from parameter-fitting helps to prevent premature structural convergence.

In traditional GP approaches, parameters are usually evolved together with programme structure. In our approach, we use the opportunity to differentiate mutation and selection on the model structure from parameter fitting. To this end, a two-level evolutionary algorithm was implemented (Fig. 3), where the

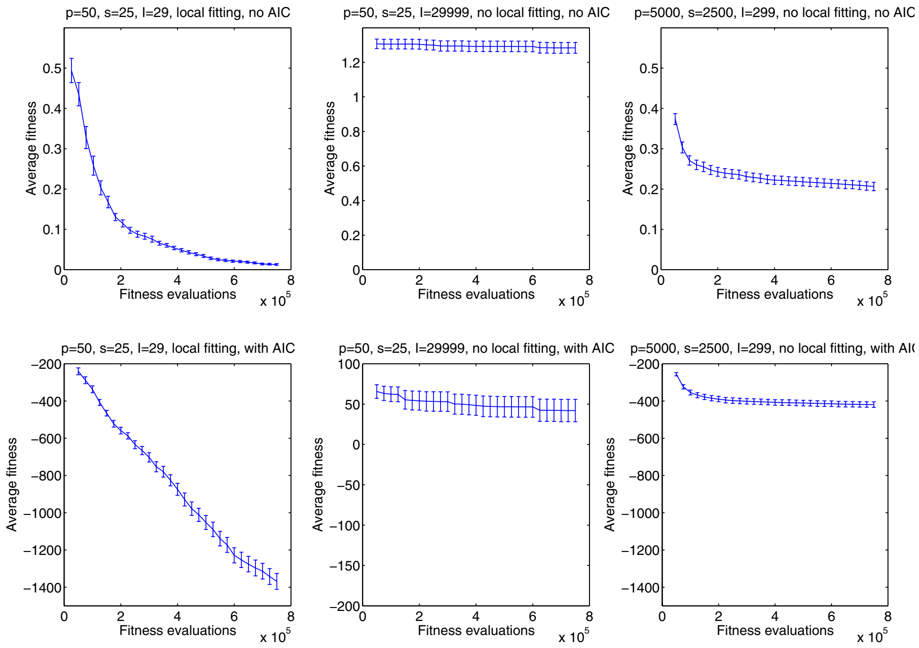

Fig. 4. Average best-fitness with standard error over ten runs of the evolution of logarithm networks. Left column: two-level EA, middle column: one-level for many generations (note the different scale), right column: one-level with a large population. Upper row is without AIC, lower row uses AIC. Headers: p = population size, s = number of survivors, I = number of generations.

upper level evolves a model structure in analogy to GP, while the lower level takes care of the parameters with an evolution strategy (ES) [3].

In order to test the effect of this separation on the performance, we evolved networks supposed to perform two tasks: calculating the third root and logarithm of a positive real number. Here, "calculating" means that the input is set as initial concentration of species *input*, while the output is read from the steady state concentration of species *output*. Therefore, the target time series for the *output* species is simply the desired output value, constant over a period of time, where the first few timepoints are excluded from the fitness evaluation. An example solution for a third root network is shown in Fig. 2. While the third root has been observed to be solvable but substantially more difficult than a square root network [7], no precise solution to the logarithmic problem is known yet. Therefore, the best possible approximation to the logarithm is sought. In this work, the main focus is not put on the evolved networks, but rather on the evolutionary process.

Three different strategies were used, each with and without AIC:

1. Two-level evolution using ES for local fitting (upper level: (25+25)-elitist selection, 29 generations, only structural mutations; lower level: (5,15)-ES, 99 generations, only parameter mutations)

2. One-level evolution running for more generations ((25+25)-elitist selection, 29999 generations, structural and parameter mutations)
3. One-level evolution employing a larger population ((2500+2500)-elitist selection, 299 generations, structural and parameter mutations)

The parameter settings were chosen such that the number of fitness evaluations and the ratio of structural vs. parameter mutations are identical, enabling an objective comparison. The one-level strategies invested the saved fitness evaluations into more generations (2) or more individuals (3). In the ES, adaptive stepsizes were disable to make the results comparable. Computations were carried out as single-processor runs on a cluster of workstations equipped with two Dual Core AMD Opteron(tm) 270 processors running Rocks Linux.

Results of the evolution of a logarithm-network (Fig. 4) show that the two-level structure of the algorithm improves fitness development drastically in comparison to a larger number of generations, while it prevents the premature convergence seen with a larger population. A large population seems to enable the algorithm to guess a good initial network, but it is unable to improve upon this. In contrast, the two-level approach improves the network continuously, yielding significantly better results in the end.

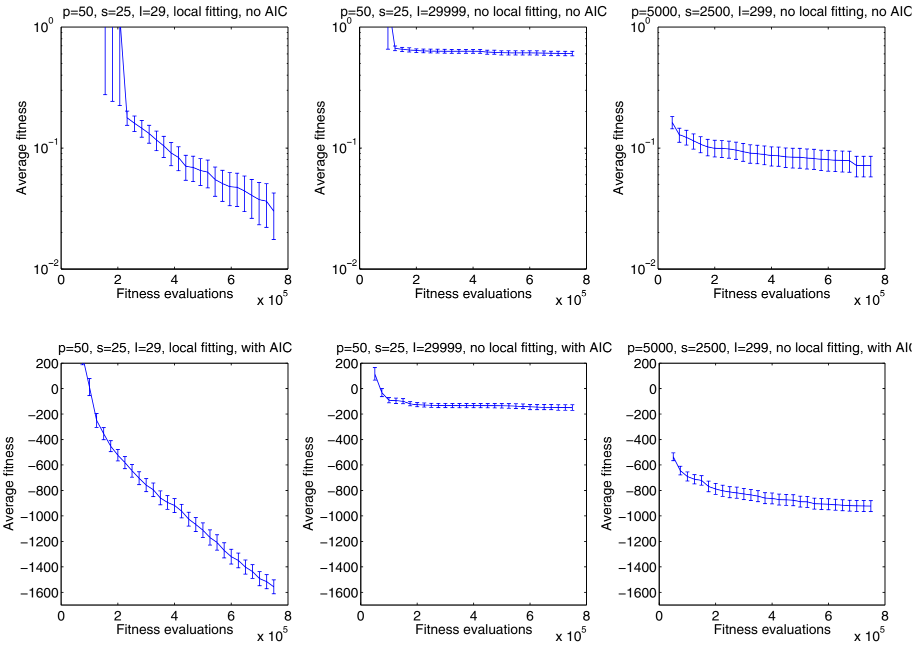

Fig. 5. Average best-fitness with standard error over 100 runs of the evolution of third root networks. Left column: two-level EA, middle column: one-level for many generations, right column: one-level with a large population. Upper row (log scale) is without AIC, lower row uses AIC. Headers: see Fig. 4.

Figure 5 shows the average fitness development for the third root task. Results are similar to Fig. 4, although not as pronounced. Again, the two-level strategy drastically outperforms the setting with more generations, while its initial progress is slower than for a large population. However, the large-population approach converges too early, while the two-level setting continuous to improve in a smooth fashion. In this task, the networks were also required to be mass-conserving, i.e. it was demanded that a feasible configuration of molecular masses for the different species exists. This constraint might explain the slower rate of convergence in comparison to the logarithm-trials.

As a control test, we performed a random search with the same parameters, replacing mutations in the evolutionary algorithm with creation of new random individuals. The EA drastically outperformed random search, resulting in fitness values one order smaller after 750000 fitness evaluations (data not shown). Even though random search finds good initial solutions, it cannot narrow its search and thus lacks the ability to fine-tune the network for the desired calculation.

4 Using AIC to Evolve Parsimonious Models

Another focus of our investigations was the effect of using Akaike's Information Criterion (AIC) as a fitness measure. This measure weights the goodness-of-fit of a model against the number of its free parameters. Given that more parameters will lead to a better fit, but not always to a better explanation of a dataset, AIC formalises a compromise between free parameters and data-fitting. For an overview of information-theory model selection tools, including AIC, see [4].

To investigate AIC, we compared fitness values and free parameters after runs with AIC with those without. Our results are mixed: while AIC has a tendency to reduce model size (not shown), it can drastically affect fitness development, especially for the one-level approaches (Fig. 4 and 5). It seems that AIC either causes premature convergence to small models with bad function, or models achieve the desired function while size increases. This effect is explained by considering that AIC assumes stochastic data, which target time-courses here are not. When models can be fitted perfectly to desired values, the goodness-of-fit dominates the number of free parameters.

5 Species Duplication - A Soft Mutation Operator

A major problem with evolving biochemical networks seems to be the often deleterious effect of structural mutations on network behaviour. Additions and deletions of species and reactions usually change the resulting time series drastically, especially for smaller networks. Therefore, we are looking for "softer" mutation operators. Inspired by biology, one such operator is the duplication of a species and all the reactions it participates in. When the rate constants of all reactions producing the species are halved, this operator does not affect the concentrations of non-mutated species. Later on, deletions and rate mutations can exploit the additional freedom gained by duplication.

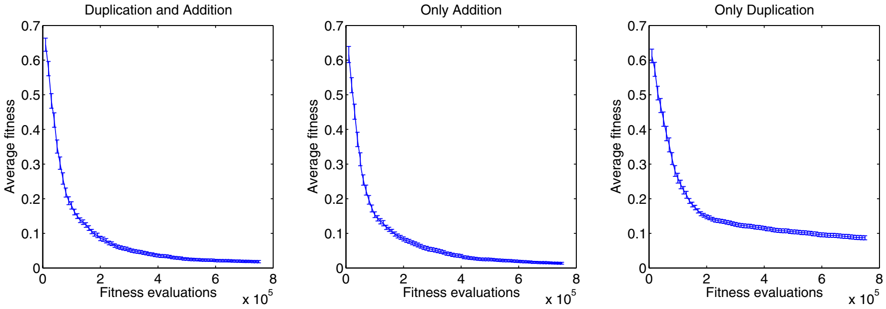

Fig. 6. Average fitness development with standard error for logarithm networks, results from 100 independent runs. Left: species addition and duplication together, middle: only addition, right: only duplication. Also shown are the best 5 runs per setting (gray). Global selection is (50+100)-elitistic, local fitting is a (1+10)-ES, and 50 generations were calculated.

Our results (Fig. 6) show that even though species duplication alone is inferior to addition of new species with random reactions, combining both operators does not yield an inferior result. However, it is still open under which conditions the combined approach improves the random addition of new species.

6 Case Study: The Human Cell Cycle Spindle Checkpoint

Segregation of newly duplicated sister chromatids into daughter cells during anaphase is a critical event in each cell division cycle. Any mishap in this process gives rise to aneuploidy that is common in human cancers and some forms of genetic disorders [5]. Eukaryotic cells have evolved a surveillance mechanism for this challenging process known as the spindle checkpoint. The spindle checkpoint monitors the attachment of kinetochores to the mitotic spindle and the tension exerted on kinetochores by microtubules and delays the onset of anaphase until all the chromosomes are aligned at the metaphase plate [8].

To demonstrate the usefulness of our approach in systems biology, we applied combined structural- and parameter-optimisation to a recent model by Ibrahim et al. [12] of the mitotic spindle checkpoint. This model, which is originally crafted by hand according to literature and laboratory data, describes in details the concentration dynamics of 17 species, namely Mad2, Mad1, BubR1, Bub3, Mad2*, Mad1*, BubR1*, BubR1:Bub3, APC, Cdc20, MCC, MCC:APC, Cdc20:Mad2, and APC:Cdc20, CENPE, Mps1 together with Bub1, and the kinetochore as a pseudospecies. Different kinetochores are represented by three compartments coupled by diffusion, each with the same 11 reaction rules. The last four species represent input signals to the model, reflected in the rate constants of certain reactions. The model corresponds to biological experimental results, which characterise the main components of the mitotic checkpoint.

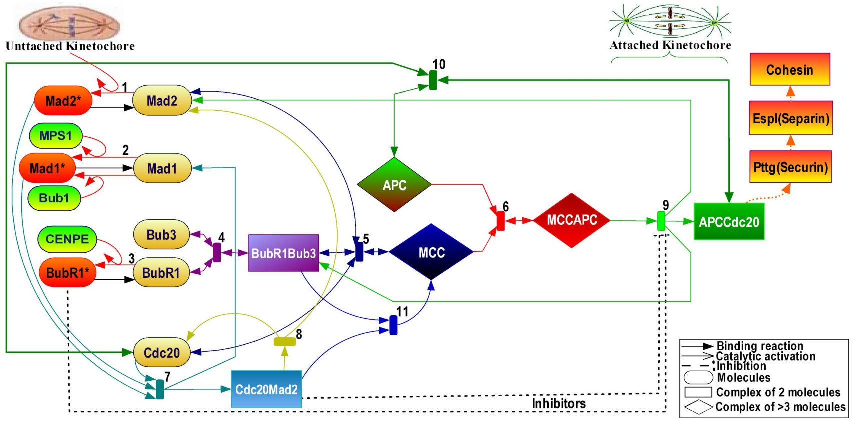

Fig. 7. Schematic network model of mitotic spindle checkpoint. Figure taken from [12].

Table 1. Steady-state concentrations of APC:Cdc20 for four settings of the cell: All kinetochores unattached (1), one attached (2), two attached (3), all three attached (4). The unoptimised model has all parameters set to 0.1, the parameter optimised one has the values from [12], and the structurally optimised model is the result of the procedure described here. Note that the fitness function used is different to the one in [12].

	Level 1	Level 2	Level 3	Level 4	Fitness
Desired value	0	0	0	0.3	0
Without optimisation	0.086154	0.0865285	0.0869323	0.0872706	27.0122
Parameter optimised	0.010924	0.011051	0.011359	0.298768	0.470562
Structure optimised	0.000106	0.000051	0.000037	0.29972	0.000077

As an optimisation target, the concentration of the central species APC:Cdc20 is supposed to be low as long as not all kinetochores are attached, but to rise to a higher value when they all are. In [12], this target has been combined with behaviour from knockout-experiments (which we do not consider here) to fit the rate constants. Here, we test which results can be achieved when the algorithm is allowed to add additional reactions. Any reaction given in the original model cannot be deleted. In future, it will be interesting to loosen this, which could lead to an evolutionary model reducer isolating only the important parts of a given model.

Our results, summarised in Table 1, show that performance of the model has improved compared to the optimisation of rate constants alone. In principle, this could also help to uncover additional structure in the data and to propose additional features of the system which can then be verified experimentally. To achieve biological plausibility, these additional features have to be constrained, which has not been observed in this proof of concept. The best model evolved contains only two additional reactions compared to the original,

$$\begin{aligned} \mathrm{BubR1}_Z &\rightarrow \mathrm{Mad1}^*_X + \mathrm{BubR1}^*_Y \\ \mathrm{BubR1}^*_X + \mathrm{Cdc20}_Y &\rightarrow \mathrm{Mad2}_X + \mathrm{Cdc20}_Y, \end{aligned}$$

but these require species from different compartments to interact, which is not intended in the model. We are currently extending our work to include plausibility constraints.

7 Conclusions

As our results for the third root and logarithm tasks indicate consistently, separating structural network evolution from parameter evolution in a two-level algorithm improves the fitness performance significantly. It can be expected that the inclusion of an adaptive stepsize - which has been excluded here to focus on the separation effect - will deepen this advantage. This result is especially interesting as it is in contrast to traditional GP approaches, where parameters are usually evolved simultaneously to the programme structure.

Results for species duplication show that this operator indeed has a beneficial effect on the algorithm, but cannot be used alone, i.e. without the addition of species with random connections. The right balance between creative potential and soft adaptations in different stages of the run seems to be crucial here. For Akaike's Information Criterion, results were unexpected: instead of facilitating the evolution of parsimonious models with a good fitness, evolved solutions were either stuck to small size (usually in one-level approaches), or were of the same size as models evolved without AIC. While the first aspect results from the general tendency of one-level approaches to premature convergence, the second aspect can be explained by the noise-free target data that was used, allowing an almost perfect fit in which the size term in AIC is dominated by the logarithm of goodness-of-fit.

In Section 6, we show that the demonstrated approach can be used to automatically improve realistic models. The next steps are clearly visible now: plausibility constraints have to be included in order to restrict the evolution to solutions that are biologically meaningful. With this in mind, interesting results from this field can be expected in the near future.

Acknowledgements. We thank Stephan Diekmann and Eberhard Schmitt at the FLI Jena for fruitful collaboration on the spindle checkpoint model, and Anthony Liekens at the Technical University of Eindhoven for support with the computations. Funding from the EU (ESIGNET, project no. 12789), Federal Ministry of Education and Research (BMBF, grant 0312704A) and German Academic Exchange Service (DAAD, grant A/04/31166) is gratefully acknowledged. The ESIGNET project also provided funding for the cluster computing facility utilised for this work.

References

1. B. Alberts, A. Johnson, J. Lewis. *Essential Cell Biology.* Garland Publishing, 2003
2. W. Banzhaf, P. Nordin, R.E. Keller, F.D. Francone. *Genetic Programming, An Introduction: On The Automatic Evolution of Computer Programs And Its Applications.* Morgan Kaufmann, 1998
3. H. Beyer and H. Schwefel. *Evolution strategies.* Natural Computing 1:3-52, 2002
4. K.P. Burnham, D.R. Anderson. *Model selection and inference : a practical information-theoretic approach.* Springer, 1998
5. E. Chung, R.-H. Chen. *Spindle Checkpoint Requires Mad1-bound and Mad1- free Mad2.* Molecular Biology of the Cell 13, pp. 1501-1511, 2002
6. B.L. Cooper, N. Schonbrunner, G. Krauss. *Biochemistry of signal transduction and regulation.* Wiley-VCH, 2001
7. A. Deckard and H.M. Sauro. *Preliminary Studies on the In Silico Evolution of Biochemical Networks.* ChemBioChem 5:1423-1431, 2004
8. G. Fang. *Checkpoint Protein BubR1 Acts Synergistically with Mad2 to Inhibit Anaphase-promoting Complex.* Molecular Biology of the Cell 13, pp. 755-766, 2002
9. A. Finney, M. Hucka. *Systems biology markup language: Level 2 and beyond.* Biochem Soc Trans, 31(Pt 6):1472–1473, 2003.
10. P. François, V. Hakim. *Design of Genetic Networks With Specified Functions by Evolution in silico.* PNAS 101:580-585, 2004
11. A. Funahashi, N. Tanimura, M. Morohashi, H. Kitano. *CellDesigner: a process diagram editor for gene-regulatory and biochemical networks.* BIOSILICO 1:159-162, 2003
12. B. Ibrahim, S. Diekmann, E. Schmitt, P. Dittrich. *Compartmental Model of Mitotic Spindle Checkpoint Control Mechanism.* BMCBioinformatic, Submitted Paper, 2006
13. J.R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA: MIT Press, 1992
14. J.R. Koza, W. Mydlowec, G. Lanza, J. Yu, M.A. Keane. *Automatic Synthesis of Both the Topology and Sizing of Metabolic Pathways using Genetic Programming.* Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), pp. 57–65, Morgan Kaufmann, 2001
15. W.B. Langdon, R. Poli. *Foundations of Genetic Programming.* Springer, 2002
16. R. Machne, A. Finney, S. Muller, J. Lu, S. Widder, C. Flamm. *The SBML ODE Solver Library: a native API for symbolic and fast numerical analysis of reaction networks.* Bioinformatics 22(11), pp. 1406-7, 2006
17. S.R. Paladugu, V. Chickarmane, A. Deckard, J.P. Frumkin, M. McCormack, H.M. Sauro. *In Silico Evolution of Functional Modules in Biochemical Networks.* IEE Proceedings-Systems Biology 153(4), 2006