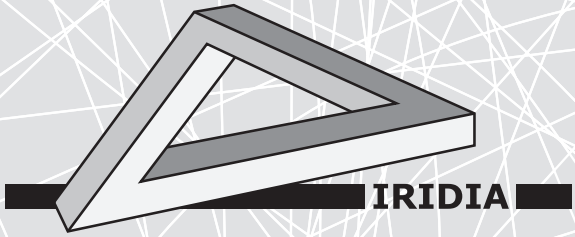


Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

On the difficulty of inferring gene regulatory networks: A study of the fitness landscape generated by relative squared error

Francesco Sambo, Marco A. Montes de Oca,
Barbara Di Camillo, and Thomas Stützle

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2009-008

May 2009

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:
IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2009-008

Revision history:
TR/IRIDIA/2009-008.001 May 2009

On the difficulty of inferring gene regulatory networks: A study of the fitness landscape generated by relative squared error

Francesco Sambo[a] sambofra@@dei.unipd.it
Marco A. Montes de Oca[b] mmontes@ulb.ac.be
Barbara Di Camillo[a] dicamill@@dei.unipd.it
Thomas Stützle[b] stuetzle@ulb.ac.be

[a]Dipartimento di Ingegneria dell'Informazione, Università di Padova, Padua, Italy
[b]IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium

May 2009

Abstract

Inferring gene regulatory networks from expression profiles is a challenging problem that has been tackled using many different approaches. When posed as an optimization problem, the typical goal is to minimize the value of an error measure, such as the relative squared error, between the real profiles and those generated with a model whose parameters are to be optimized. In this paper, we use recurrent neural networks to model regulatory interactions and study systematically the "fitness landscape" that results from measuring the relative squared error. Although the results of the study indicate that the generated landscapes have a positive fitness-distance correlation, the error values span several orders of magnitude over very short distance variations. This suggests that the fitness landscape has extremely deep valleys, which can make general-purpose state-of-the-art continuous optimization algorithms exhibit a very poor performance. Further results obtained from an analysis based on perturbations of the optimal network topology support approaches in which the spaces of network topologies and of network parameters are decoupled.

1 Introduction

In the cells of living organisms, genes are transcribed into mRNA (messenger RNA) molecules which, in turn, are translated into proteins [8]. Some proteins, called transcription factors, can increase (activate) or decrease (inhibit) the transcription rates of genes; other proteins can control the translation of mRNA into new proteins. The process whereby genes control, indirectly via the proteins they encode, the expression (i.e., the mRNA transcription rate) of other genes, is known as *genetic regulation* [8]. Knowing the regulatory relations among genes is important for understanding fundamental processes that occur within living cells.

DNA microarray technology [19] has enabled researchers to monitor the expression of the whole genome under various genetic, chemical and environmental perturbations. The output data from DNA microarray experiments, in the form of gene expression time series, can be used to infer a *gene regulatory network* (GRN). A GRN is a graph in which the nodes, representing genes or proteins, are connected by an edge if a regulatory relation exists between them. Different approaches have been adopted in the literature to model and infer GRNs from DNA microarray experiments [3, 9, 26]. A very common approach for inferring a GRN is to cast the problem as one of optimizing the free variables of a model that is capable of generating time expression profiles. In this case, the goal of the optimization process is to minimize a cost function quantifying the differences between the real temporal profiles and the profiles generated with the current estimation of the model's parameters. Unfortunately, the problem of inferring GRNs from gene expression profiles using optimization techniques has proved to be difficult even when dealing with very small networks (5-10 genes) [24].

In this paper, we address the issue of the difficulty of inferring GRNs by performing an analysis based on the notion of fitness-distance correlation (FDC) [11, 10]. To model regulatory interactions we decided to use recurrent neural networks (RNNs) [15], which model the set of genes as a system of nonlinear differential equations, and we adopted the relative squared error (RSE) as a measure of the lack of accuracy of time profiles generated by an inferred network with respect to those of a target GRN. As a first contribution, we present an analysis of the error surface generated by the combination RNN-RSE (Section 4.1). The main result of this analysis is that the RNN-RSE error surface has a strong positive fitness-distance correlation; however, the data also shows the existence of many local optima of extreme depth, which seems to be the main cause for the poor performance shown by optimization algorithms on this problem. A second contribution is the quantification of the effect that a priori information on the target's GRN structure has on the fitness landscape (Section 4.1). The final contribution is the analysis of the behavior of a state-of-the-art continuous optimization algorithm (NEWUOA [22] with multiple restarts) on the problem with and without *a priori* network structure information (Section 4.2). The results obtained from this analysis constitute strong evidence in favor of inference approaches in which the spaces of network topologies and of network parameters are decoupled.

2 Modeling Gene Regulatory Networks

Many mathematical models exist in the literature to describe gene regulatory interactions: Relevance Networks [17], Boolean Networks [16], Dynamic Bayesian Networks [6] and systems of additive or differential equations, being them linear [1], ordinary nonlinear [7, 13, 23, 25, 28, 27] (including recurrent neural networks) or S-systems [21, 14, 24].

Systems of equations are commonly used as a modeling tool by the metaheuristics community, because the problem of fitting the model to data can be mapped easily to an optimization problem. In that case, the model's parameters form the search space and the fitness function is usually a variant of the error between the real temporal profile and the on estimated from the fitted model.

Linear and additive models lack the capability to capture real regulatory

relations, which in general are highly nonlinear and differential; S-systems, on the contrary, are suited to accurately describe the behavior of small sets of genes, but are impractical on large scale scenarios, because of the high number of free parameters ($2n(n+1)$ for a network of n genes).

Considering the limitations of the methods mentioned above, we use recurrent neural networks (RNNs) [15], which model the set of genes with a system of nonlinear differential equations of the form

$$\frac{dx_i}{dt} = \frac{k_1}{1 + \exp\left(-\sum_{j=1\ldots n} w_{ij} x_j + b\right)} - k_2 x_i \,, \qquad i = 1 \ldots n \qquad (1)$$

where n is the number of genes in the system, x_i is the rate of expression of gene i, w_{ij} represents the relative effect of gene j on gene i ($1 \leq i, j \leq n$), b is a bias coefficient, k_1 is the maximal rate of expression and k_2 is the degradation rate. For our analysis, we set for simplicity $b = 0$, $k_1 = 1$ and $k_2 = 1$. The search space for an optimization algorithm, then, is formed by the matrix W of coefficients w_{ij}.

An identical model is suggested in [25] for the analysis of microarray data from an experiment on *Saccharomyces Cerevisiae* cell cycle, and is adopted in [28] and [27] for a reverse engineering algorithm based on particle swarm optimization [12]. In the latter two cases, however, derivatives are approximated with finite differences and estimated from temporal data: such an approach amplifies the effects of noise and requires a large amount of data points. Thus, we decided to maintain derivatives and to generate temporal profiles with numerical integration of the whole system. For this purpose, we chose a Runge-Kutta-Fehlberg method with adaptive step size control [5].

3 Experimental Dataset

Experimental data are generated with the simulator recently introduced in [4]. In this simulator, the regulatory network's topology is generated according to the current knowledge of biological network organization, including scale-free distribution of the connectivity and a clustering coefficient independent of the number of nodes in the network. The resulting networks are very *sparse*, that is, the number of edges at most doubles the number of nodes, therefore the majority of elements in the connectivity matrix are equal to zero. Nonzero elements of the matrices generated by the simulator (w_{ij} terms in Equation 1) are then set uniformly at random in the range $[-10, 10]$.

To generate simulated gene expression time series, the expression of each gene is initialized uniformly at random and the system is let free to evolve to a steady state. Gene profiles are then sampled with logarithmic time spacing, so that the majority of samples are taken right after the initialization. This practice is common in real microarray experiments, because meaningful information usually concentrates right after the external stimulation of a dynamical system.

The analyses reported in this paper are carried over on gene networks of size 10, in line with experimental results from the state-of-art [24, 27, 28].

4 Analysis

To investigate the structure of the fitness landscape of our optimization problem, we performed a fitness-distance correlation analysis [11, 10]. We randomly sampled interesting areas of the search space and studied, for sampled solutions, the distribution of fitness values versus distance from the optimal solution. In our case, a fitness value is considered to be better if the solution associated with it has a lower value of the objective function.

For the fitness function, we used the relative squared error (RSE) between real and estimated temporal profiles, which is defined as

$$RSE = \frac{1}{Tn} \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{[\hat{x}_i(t) - x_i(t)]^2}{x_i(t)}, \tag{2}$$

where n is the number of genes, T is the number of time samples, $x_i(t)$ is the real value for gene i at time t and $\hat{x}_i(t)$ is the estimated value for the same sample.

Preliminary analyses with mean squared error, another measure widely used as fitness function, showed the same behavior for the two types of errors, thus we concentrated the study only on RSE. As distance measure between candidate solutions and the optimal solution, we used the Euclidean distance.

Fitness-distance correlation analysis is a standard tool for search space analysis that is used in many research efforts on evolutionary algorithms and has lead to a number of interesting insights, as an example see [18].

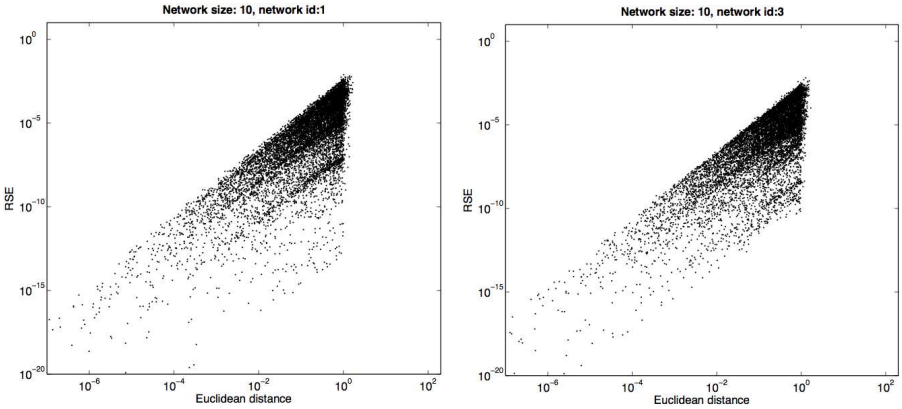
4.1 Fitness Distance Correlation Analysis

Three type of analysis have been performed. In the first, we introduce perturbations that affect any of the n^2 matrix elements, that is, zero and nonzero elements. In the second, only nonzero elements are affected. Finally, in the third, we perturbe the structure of the network, changing the pattern of nonzero elements.

4.1.1 Step 1

As a first step of our analysis, we explored relations between fitness and distance among a set of random perturbations of the optimal solution: each element of the optimal matrix was perturbed with the addition of a log-uniformly distributed random variable (*i.e.* a random variable uniformly distributed in logarithmic scale) in the interval $[10^{-a}, 10^{-0}]$, where a was tuned to account for different problem sizes. Results for 10000 iterations of the perturbation procedure on two networks of 10 genes are shown in Figure 1.

As it can be seen from the figure, there is a strong correlation between Euclidean distance and RSE, because samples distribute along a band with positive slope, but the band is rather wide (approximately 10 orders of magnitude of RSE for Figure 1(a) and 6 orders for Figure 1(b)), thus leading to an average correlation coefficient of 0.471: this suggests the presence of extremely deep valleys in the fitness landscape, which can be an obstacle for a general purpose continuous optimization algorithm. We formulate the hypothesis that the difficulty in solving the particular problem instance is closely related to the width of the



(a) Example of widely distributed samples (network 1)

(b) Example of more narrowly distributed samples (network 3)

Figure 1: RSE *vs* Euclidean distance of 10000 log-uniform perturbations of the elements of the optimal system matrix, for two networks of 10 genes.

band in the fitness-distance plot. The perturbation procedure was repeated for 20 different problem instances of 10 genes. For the majority of them (17 over 20) the band in the RSE *vs* distance plot exhibits a width close to the one of network 3 (Figure 1(b)), and for the remaining instances the width is larger, close to the one of network 1 (Figure 1(a)). Therefore, we decided to use network 1 and 3 throughout the paper as two representative examples of problem instances, to validate empirically our hypothesis.

4.1.2 Step 2

As a second step, we decided to investigate the relation between the features of the search space and the structure of the networks (*i.e.* the pattern of zero and nonzero elements in the weight matrix): gene networks are largely sparse, thus the number of parameters to be fine tuned by an optimization procedure is small with respect to the number of variables in the search space. We wanted to understand how much the search space is affected by information on the network structure.

To this end, we perturbed only nonzero elements of the optimal solution for each problem instance, fixing to zero the other elements. As before, perturbations were obtained with the addition of a log-uniformly distributed random variable. RSE vs Euclidean distance of 10000 perturbations for network 1 and 3 are shown in Figure 2.

Even though the average correlation coefficient is 0.424, thus slightly lower than the one form the previous step, fitness-distance plots tend to be more structured: as it is clear from the figure, most of the samples lie on straight lines parallel to the bands of the previous experiment, and the vertical span of the lines reflects the width of the bands. Further experiments (data not shown) showed that each line corresponds to a single nonzero element of the weight matrix. This suggests that some variables may be optimized independently. Such an hypothesis was not necessarily evident from the mathematical description of the system and should be explored in future work.

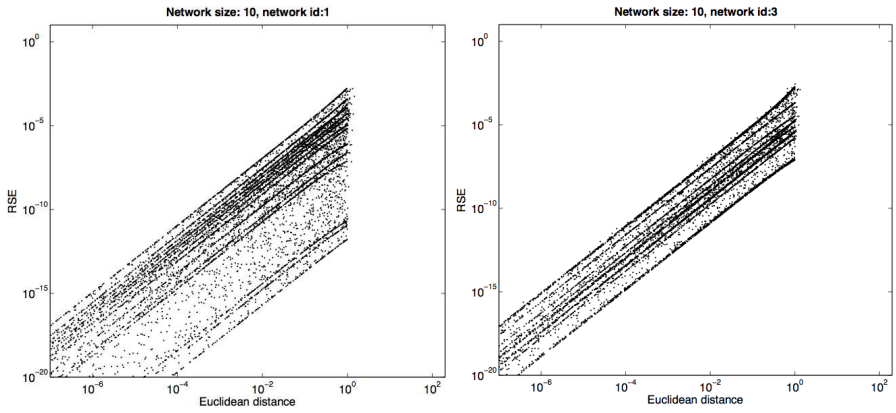


Figure 2: RSE *vs* Euclidean distance of 10000 log-uniform perturbations of nonzero elements of the optimal system matrix, for two networks of 10 genes. Plots are cut to keep the same scale adopted in the other figures; the diagonal lines spread with the same behavior down to 10^{-15} for Euclidean distance and 10^{-35} for RSE.

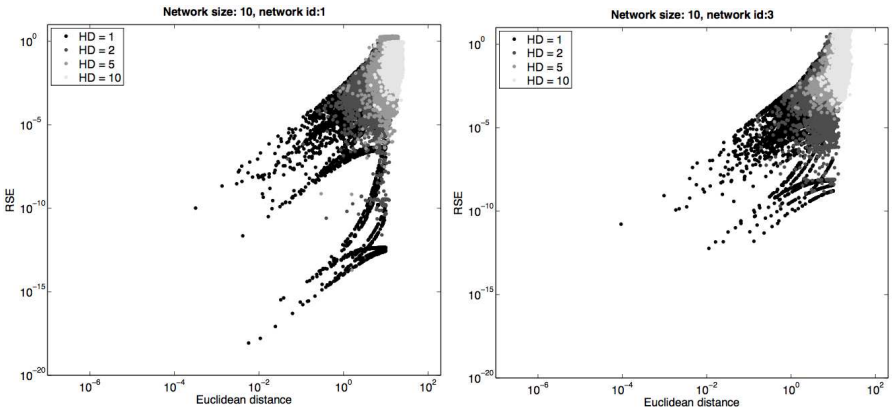
4.1.3 Step 3

We then decided to further explore the shape of the fitness landscape in regions close in structure to the global optimum; for this purpose, we exploited the concept of *Hamming distance* between two connectivity matrices, *i.e.* the number of bits that differ between the two matrices, and we randomly sampled Boolean matrices at Hamming distance 1, 2, 5 and 10 from the global optimum.

We then kept original values for elements that are nonzero in both matrices, the original one and the sampled one, and set new values for the other nonzero elements, drawing them uniformly at random from the interval $[-10, 10]$. 10000 samples for each value of Hamming distance are shown in Figure 3, where lighter gray corresponds to higher Hamming distance, for networks 1 and 3.

From the figure, it is evident that at higher Hamming distances there is no particular correlation between fitness and distance, but when the Hamming distance decreases the fitness vs distance plot becomes more and more organized, approaching the global shape of the bars from Figure 1. Indeed, average correlation coefficients are 0.219, 0.196, 0.185 and 0.177 for networks at Hamming distance 1, 2, 5 and 10, respectively. At Hamming distance 1, samples tend to appear as curved lines and the structure of the plots become closer to the one from Figure 2.

This latter analysis outlines that portions of the search space which correspond to networks structurally close to the optimum (i.e., at a low Hamming distance) present more organization in the fitness landscape, and can thus be a local basin of attraction for an algorithm which searches in the discrete space of network structures. To test the quality of a particular network structure, a second algorithm can be alternated to the first, to optimize continuous nonzero values of the network; for the second algorithm, the probability of finding the optimal solution should increase as the network structure become closer to the optimal structure.



(a) (b)

Figure 3: RSE *vs* Euclidean distance of 10000 log-uniform perturbations of the optimal system matrix at Hamming distance 1, 2, 5, 10, for two networks of 10 genes.

4.2 Algorithm Behavior

The analysis presented above gives an overall picture of the fitness-distance relationship in the search space. In addition, it is of interest to study the behavior of a specific algorithm in the search space. The question we want to address is whether an algorithm is capable of inferring the structure of the target network using only the information provided by the RSE measure. If that is not the case, a second experiment consists in measuring the performance of the algorithm when the optimal network topology is known *a priori*. For our experiments, we use NEWUOA [22], which is a software for unconstrained continuous optimization in many dimensions that does not need information about the derivatives of the objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ it is applied to. At each iteration, NEWUOA creates a quadratic model that interpolates k values of the objective function which is used in a trust-region procedure [2] to update the variables. The main advantage of NEWUOA is that it can be used to solve large scale optimization problems thanks to the reduced number of interpolation points it needs to build the quadratic model (usually $k = 2m + 1$, where m is the number of variables to optimize, is recommended). NEWUOA is considered to be a state-of-the-art continuous optimization technique [20]. By definition, trust-region methods search locally, which means that they may converge to some local optimum in the case the objective function is multimodal. For this reason, we used NEWUOA with multiple restarts, so as to explore different regions of the search space in order to reduce the chances of converging to low-quality local optima. In our setting, NEWUOA is restarted from a new initial solution after it has reached a maximum number of function evaluations, or when the final radius of the trust region reaches a certain threshold. In Table 1, we show the parameters used in our experiments. These parameters were chosen after an initial non-exhaustive experimentation phase.

The results obtained from running NEWUOA with multiple restarts without any *a priori* information about the correct topology of the target GRN are shown

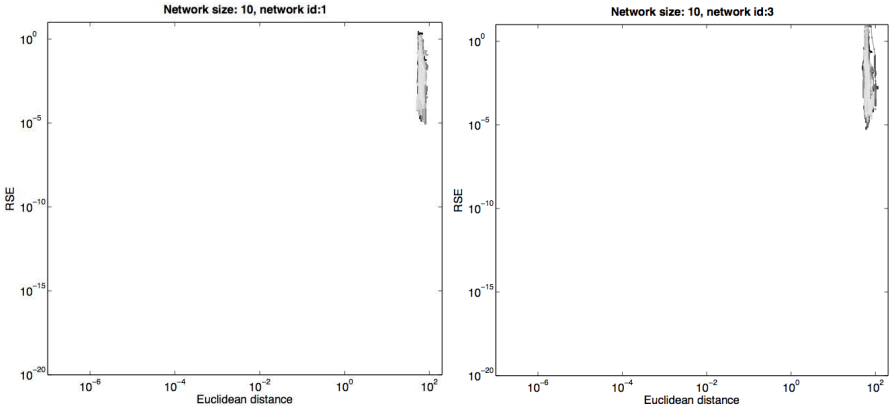
Table 1: Parameters used with NEWUOA with multiple restarts

Parameter	Value
Initial trust region radius	0.2
Final trust region radius	1×10^{-10}
Number of interpolation points	$k = 2m+1$, where m is the number of variables to optimize
Maximum number of function evaluations per NEWUOA run	2×10^4
Maximum total number of function evaluations	2×10^5, with structure information
	1×10^6, without structure information
Number of independent runs	100

in Figure 4. Each shade of gray represents a run of the algorithm. Although the algorithm is capable of making progress in terms of the value of the objective function (it descends from a value in the order of 10^0 to a value in the order of 10^{-5}), it does not make any progress towards the actual target GRN. This can be seen by the (almost) vertical lines that appear on the upper right corner of the plots in Figure 4.

In Figure 5, we show the results obtained after running NEWUOA with multiple restarts when the correct topology of the target GRN was used by the algorithm, which is equivalent to reducing the size of the search space so that only nonzero entries are optimized. As before, each shade of gray represents a run of the algorithm. In this case, the behavior of the algorithm depends on the target network. With network 1, the algorithm moves towards the optimal solution while improving the value of the RSE in both cases over several orders of magnitude. However, in the vast majority of cases, the algorithm cannot find solutions that are closer than a distance of 100 to the optimal solution. In contrast, with network 3, the algorithm is capable to find the optimal solution in each run.

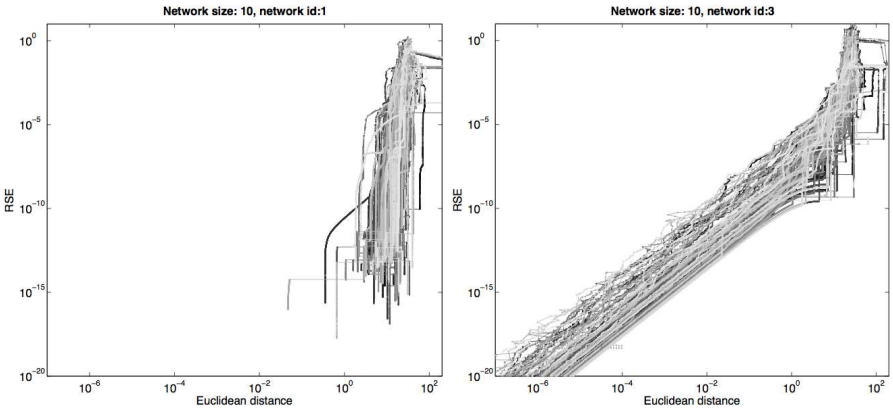
The results presented above, together with those of the analysis based on structure perturbations, constitute strong evidence in favor of optimization algorithms that explicitly intertwine a network structure search phase with a network's parameters search phase. A reduction in the distance from the optimal network topology allows a continuous optimization algorithm to make more progress toward the truly optimal solution. Although we tested our hypotheses using only one specific algorithm, we do not expect our observations to change substantially if another algorithm is used. This is because, as evidenced in Figure 5, even with a perfect information about the correct topology of the target GRN, the error surface generated by the RSE measure is still hard to search as it is multimodal in nature.



(a) NEWUOA with multiple restarts on network 1

(b) NEWUOA with multiple restarts on network 3

Figure 4: The progress of NEWUOA with multiple restarts on two 10-gene-network inference problems. Each shade of gray represents a run of the algorithm. The plots shown correspond to the case in which no *a priori* information about the correct topology of the target GRN is provided to the algorithm.



(a) NEWUOA with multiple restarts on network 1

(b) NEWUOA with multiple restarts on network 3

Figure 5: The progress of NEWUOA with multiple restarts on two 10-gene-network inference problems. Each shade of gray represents a run of the algorithm. The plots shown correspond to the case in which the correct topology of the target GRN is provided to the algorithm.

5 Conclusions and related works

In this work, we presented a study of the fitness landscape for the problem of gene regulatory networks inference, when recurrent neural networks are adopted as a model for gene regulation and relative squared error is chosen as a fitness function. As far as we know, this is the first study on fitness-distance correlation analysis for the problem of gene regulatory networks inference.

The study consists in a fitness-distance correlation analysis of different random samplings around the problem's optimal solution, which is in the form of a weight matrix W. The optimal matrix was first perturbed globally, then only on its nonzero elements and at fixed Hamming distance. Results show that the error surface has a strong positive fitness-distance correlation, but they also reveal the presence of extremely deep valleys in the fitness landscape, which are responsible for the poor performance of optimization algorithms not designed explicitly for this problem.

The network structure perturbation analysis highlights that: (i) RSE alone is not sufficient to guide a search algorithm towards regions of the search space close to the global optimum, (ii) even if information about the optimal network structure is provided to the algorithm, convergence to the global optimum is not guaranteed because the fitness landscape presents many deep local optima, and (iii) the closer a network structure is to the one of the optimal solution, the higher the chances are that an algorithm converges to the optimum. This last fact seems to be due to the higher level of organization in the fitness landscape in the proximity of the optimal structure.

Because of these observations, we conclude that a two-phase algorithm, which alternates between a search step in the discrete space of network structures and a search step in the continuous space of nonzero system parameters, has the potential of reaching high-quality solutions. Research in this direction has already been done, for example in [27, 23, 13], but no analysis of the underlying fitness landscape had been performed before.

References

[1] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernardo. How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, 3(78), February 2007.

[2] Andrew R. Conn, Nicholas I.M̃. Gould, and Philippe L. Toint. *Trust-Region Methods*. MPS-SIAM Series in Optimization. SIAM, Philadelphia, PA, 2000.

[3] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[4] B. Di Camillo, G. Toffolo, and C. Cobelli. A gene network simulator to assess reverse engineering algorithms. *Annals of the New York Academy of Sciences*, 1158(1):125–142, 2009.

[5] Erwin Fehlberg. Low-order classical runge-kutta formulas with step size control and their application to some heat transfer problems. Technical Report 315, NASA, 1969.

[6] F. Ferrazzi, P. Sebastiani, M. F. Ramoni, and R. Bellazzi. Bayesian approaches to reverse engineer cellular systems: a simulation study on nonlinear gaussian networks. *BMC Bioinformatics*, 8 Suppl 5, 2007.

[7] Peter Gennemark and Dag Wedelin. Benchmarks for identification of ordinary differential equations from time series data. *Bioinformatics*, 25(6):780–786, 2009.

[8] Lawrence Hunter. Life and its molecules: A brief introduction. *AI Magazine - Special issue on AI and Bioinformatics*, 25(1):9–22, 2004.

[9] T. Ideker, O. Ozier, B. Schwikowski, and A.F̃. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18(Suppl. 1):S233–S240, 2002.

[10] Terry Jones. Evolutionary algorithms, fitness landscapes and search. Working Papers 95-05-048, Santa Fe Institute, May 1995.

[11] Terry Jones and Stephanie Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192, San Francisco, CA, 1995. Morgan Kaufmann.

[12] James Kennedy, Russell Eberhart, and Yuhui Shi. *Swarm Intelligence*. Morgan Kaufmann, San Francisco, CA, 2001.

[13] Kyriakos Kentzoglanakis, Matthew Poole, and Carl Adams. Incorporating heuristics in a swarm intelligence framework for inferring gene regulatory networks from gene expression time series. In *ANTS '08: Proceedings of the 6th international conference on Ant Colony Optimization and Swarm Intelligence*, pages 323–330, Berlin, Heidelberg, 2008. Springer-Verlag.

[14] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya. Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. *Bioinformatics*, 21(7):1154–1163, 2005.

[15] Stefan C. Kremer. *Field Guide to Dynamical Recurrent Networks*. Wiley-IEEE Press, 2001.

[16] S. Liang, S. Fuhrman, and R. Somogyi. Reveal: a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific Symposium on Biocomputing*, pages 18–29, 1998.

[17] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, and A. Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7 Suppl 1, 2006.

[18] P. Merz and B. Freisleben. Fitness landscapes and memetic algorithm design. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 244–260. McGraw Hill, London, UK, 1999.

[19] M. Molla, M. Waddell, D. Page, and J. Shavlik. Using machine learning to design and interpret gene-expression microarrays. *AI Magazine - Special issue on AI and Bioinformatics*, 25(1):23–44, 2004.

[20] Jorge J. Moré and Stefan M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[21] N. Noman and I. Iba. Reverse engineering genetic networks using evolutionary computation. *Genome Informatics*, 16(2):205–214, 2005.

[22] M. J. D. Powell. *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, chapter The NEWUOA software for unconstrained optimization, pages 255–297. Springer-Verlag, Berlin, Germany, 2006.

[23] Habtom W. Ressom, Yuji Zhang, Jianhua Xuan, Yue Wang, and Robert Clarke. Inference of gene regulatory networks from time course gene expression data using neural networks and swarm intelligence. In *IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology*, pages 1–8. IEEE, 2006.

[24] C. Spieth, R. Worzischek, F. Streichert, J. Supper, N. Speer, and A. Zell. Comparing evolutionary algorithms on the problem of network inference. In Mike Cattolico, editor, *Genetic and Evolutionary Computation Conference, GECCO 2006, Proceedings, Seattle, Washington, USA, July 8-12, 2006*, pages 305–306. ACM, 2006.

[25] Vu, Tra Thi, Vohradsky, and Jiri. Nonlinear differential equation model for quantification of transcriptional regulation applied to microarray data of saccharomyces cerevisiae. *Nucleic Acids Research*, 35(1):279–287, January 2007.

[26] R. Xu, X. Hu, and D. Wunsch II. Inference of genetic regulatory networks from time series gene expression data. In *Proceedings of the International Joint Conference on Neural Networks, Vol. 2.*, pages 1215–1220, Piscataway, NJ, 2004. IEEE Press.

[27] R. Xu, D. Wunsch, and F. Ronald. Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. *IEEE Transactions on Software Engineering*, 4(4):681–692, 2007.

[28] Rui Xu, Ganesh K. Venayagamoorthy, and Donald C. Wunsch, II. Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. *Neural Networks*, 20(8):917–927, 2007.