Evolutionary Optimization Through Extinction Dynamics

Jesús Marín*
Dept. of Electronics (IT)
E.U. d'Enginyers T. Industrials de Barcelona
Comte d'Urgell, 187
08036 Barcelona (Spain)
jmarin@lsi.upc.es

Ricard V. Solé†
Complex Systems Research Group
Dept. of Physics, Campus Nord - B4
Universitat Politècnica de Catalunya
08034 Barcelona (Spain)
ricard@complex.upc.es

Abstract

A recent method of optimization has been presented involving macroevolutionary dynamical rules. Extinction and diversification are introduced by following a set of steps where species/solutions are wired through an antisymmetric matrix of interactions. These connections are defined in terms of the fitness landscape and allow to explore the landscape in a highly effective way. It is shown that this search method is able to outperform genetic algorithms in a wide range of conditions.

1 INTRODUCTION

Most methods of optimization based on evolutionary algorithms involve a set of rules somewhat inspired in biological microevolution (Goldberg, 1989; Holland, 1992; Mitchell, 1996). Using the basic framework of Darwin's Natural Selection, strings of bits representing solutions to given problems (like optimization in rugged landscapes) are manipulated by following simple sets of rules mimicking mutation, recombination and competition for resources (typically for a finite space). The underlying philosophy is that of population genetics, and the results obtained from these methods have been widely used in both theoretical and applied sciences (Back et al. 1997).

But another aspect of the evolutionary process can also be used as a source of inspiration for optimization purposes in fitness landscapes. This aspect is the set of special phenomena arising in the so called macroevolutionary dynamics. By macroevolution we refer to the long-term time scale where species extinction and diversification (together with some special external stresses) dominate the large-scale dynamics. Some authors have in fact suggested that the rules operating at this large scale are basically different than those involved at the microscale. Several recent developments have shown that the large-scale dynamics of species extincion can be satisfactorily modelled through a network of species interacting through a matrix of connections (Solé and Manrubia, 1996; Solé et al. 1996; 1997). More recently, this model has been used as a basic framework to develop a new optimization method which has been shown to outperform genetic algorithms in several instances (Marín and Solé, 1998). In this paper we further explore this model and introduce a simplification which is applied to some simple cases.

2 MODEL OF EXTINCTION DYNAMICS IN MACROEVOLUTION

Following previous papers (Solé and Manrubia, 1996; Solé et al. 1996) we briefly introduce the original formulation of the macroevolution model before to use it as the basic framework to optimization purposes. The model is a network ecosystem where the dynamics is based only on the relation between species. The links indicating trophic influence between units/species (solutions in the optimization problem) are essential to determine the new state (alive or extinct) of each species at each generation and are given by real numbers $-1 < W_{ij} < +1$. The "state" of each species i (in a population of size P) in generation t is updated by following a set of simple rules. Briefly, each generation in the biological model consists in the following set of steps:

1. *Random variation*: for each species i, a connec-

*Dept. Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 08034 Barcelona (Spain)

†Santa Fe Institute, Hyde Park Road 1399, Santa Fe, New Mexico 87501 (USA)

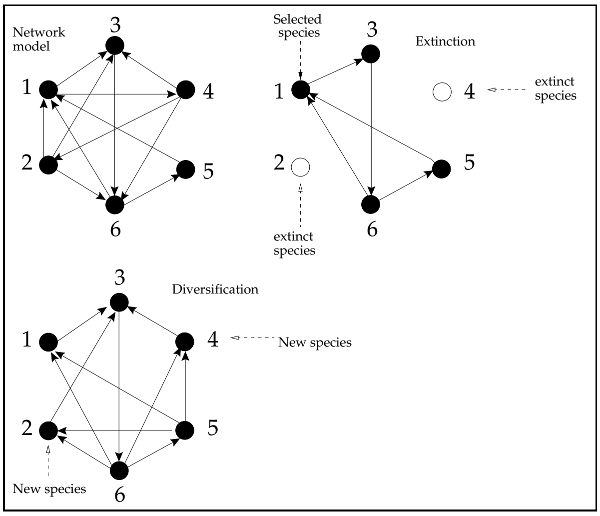

Figure 1: *Basic rules in the macroevolution model, for a system of $P = 6$ species. Starting from a random network, we first make random changes in the connections (one per species). As a consequence of these changes, some species become extinct (empty circles). Now one of the survivors is chosen (number 1 here) and copied into the (two) empty spaces (2 and 4 in this example). Arrows indicate trophic relation direction.*

tion $W_{i,j}(t)$ is randomly chosen, and a new random value between -1 and 1 is assigned

2. *Extinction*: the relation of each species with the rest of the population determines its survival coefficient h defined as

$$S_i(t+1) = \begin{cases} 1 & \text{(alive) if } \sum_{j=1}^{P} W_{i,j}(t) \geq 0 \\ 0 & \text{(extinct) otherwise} \end{cases} \tag{1}$$

This step allows for the selection and extinction of species. Here the sum $h_i = \sum_j W_{i,j}$ is the so called *local field* in statistical physics (Weisbuch, 1991).

3. *Diversification*: we colonize vacant sites freed by extinct species with surviving species. Specifically, a colonizer c will be randomly chosen from the set of survivors. For all vacant sites (i. e. those such that $S_k(t) = 0$) the new connections will be updated in this way:

$$\begin{aligned} W_{k,j} &= W_{c,j} + \eta_{k,j} \\ W_{j,k} &= W_{j,c} + \eta_{j,k} \end{aligned} \tag{2}$$

where η is a small random variation and we have $S_k(t+1) = 1$.

These rules are sumarized in figure 1, where an example for a very small system is shown. This model was shown to reproduce most of the statistical features of macroevolution (Solé et al. 1996) and it provided a natural source for the decoupling between microevolution and macroevolution. It generates avalanches of extinction of many different sizes, and small changes in the network structure are able to quickly propagate through the system. Perhaps this sensitivity could be used in optimization problems where small differences among nearest species need to be amplified in order to reach better solutions.

3 MACROEVOLUTIONARY ALGORITHM

In this section we show how to map the previous model of extinction into a model of optimization. The new model has been called *macroevolutionary algorithm* (MA). Let us define the d-dimensional fitness function f to be maximized. Our objective is to find the best values for the d-dimensional vectors of our problem under consideration. Thus our individuals/species are now $S_i \equiv \mathbf{p} \in \Omega \in \Re^d$, i. e. d-dimensional objects constrained to a subspace Ω. In this context $\mathbf{p}$ will be a good approximation if $\forall \mathbf{q} : f(\mathbf{q}) \leq f(\mathbf{p}) + \epsilon$ where $\mathbf{p}$ and $\mathbf{q}$ are individuals and $\epsilon > 0$ is a threshold. Thus each individual in MA is described by a d-input vector with fitness f. The domains for these inputs describe the search space where our fitness function is nothing but a (more or less) rugged landscape (Palmer, 1991).

As with GA, MA uses a constant population size of P individuals evolving in time by successive updates of the given operators. The main idea is that our system will choose, through network interactions, which are the individuals to be eliminated so as to guarantee exploration by new individuals and exploitation of better solutions by further generations. To this purpose, it is essential to correctly establish a relationship between individuals. This is described by the following criteria:

(a) each individual gathers information about the rest of the population through the strength and sign of its couplings W_{ij}. Individuals with higher input fields h_i will be favoured. Additionally, they will have a harmful effect on other less-fit solutions.

(b) In this version of the MA model, we define the $W_{i,j}$ as:

$$W_{i,j} = F(f(\mathbf{p}_i), f(\mathbf{p}_j))$$

where two different alternatives are considered, to be compared with standard genetic algorithms. Here $F(x, y)$ describes the specific definition of the coupling as a function of the local fitnesses. The two possibilities analyzed here are:

(i) $W_{i,j} = \begin{cases} \frac{f(\mathbf{p}_i)-f(\mathbf{p}_j)}{|p_i-p_j|} & \text{if } f(\mathbf{p}_i) \neq f(\mathbf{p}_j) \\ 0 & \text{otherwise} \end{cases}$ where the denominator is a normalization factor that weights the relative distance among solution (Marín and Solé, 1998).

(ii) the reduced version $W_{i,j} = f(\mathbf{p}_i) - f(\mathbf{p}_j)$. Here $\mathbf{p}_i = (p_i^1, ..., p_i^d)$ are the input parameters of the i-th individual.

Now we can define the most important ingredients that will be used in building the set of operators to be applied each generation:

1. *Selection operator*: it allows to calculate the surviving individuals through their relations, i.e. as a sum of penalties and benefits. The state of a given individual S_i will be given by:

$$S_i(t+1) = \begin{cases} 1 & \text{if } \sum_{j=1}^{P} W_{i,j}(t) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

   where t is generation number and $W_{i,j} = W(\mathbf{p}_i, \mathbf{p}_j)$ is calculated according to the previous possibilities (i-ii). In the following this rule will be indicated as $S_i(t+1) = \theta(h_i(t))$ where $\theta(z) = 1$ if $z \geq 0$ and zero otherwise. Additionally, for the (ii)-case, computing time can be reduced through mean of population fitness without computing W: $S_i(t+1) = f(\mathbf{p}_i) \geq \sum_{j=1}^{P} \frac{f(\mathbf{p}_j)}{P}$.

2. *Colonization operator*: it allows to fill vacant sites freed by extinct individuals (that is, those such that $S_i = 0$). This operator is applied to each extinct individual in two ways. With a probability τ, a totally new solution $\mathbf{p}_n \in \Omega$ will be generated. Otherwise exploitation of surviving solutions takes place through colonization. For a given extinct solution $\mathbf{p}_i$, we choose one of the surviving solutions as best-fitness solution, say $\mathbf{p}_b$. Now the extinct solution will be attracted towards $\mathbf{p}_b$.

   Mathematically, a possible (but not unique) choice for this colonization of extinct solutions reads:

$$\mathbf{p}_i(t+1) = \begin{cases} \mathbf{p}_b(t) + \rho\lambda\big(\mathbf{p}_b(t) - \mathbf{p}_i(t)\big) & \text{if } \xi > \tau \\ \mathbf{p}_n & \text{if } \xi \leq \tau \end{cases} \tag{4}$$

   where $\xi \in [0,1]$ is a random number, $\lambda \in [-1,+1]$ (both with uniform distribution) and ρ and τ are given constants of our algorithm. So we can see that ρ describes a maximum radius around surviving solutions and τ acts as a temperature parameter. Other alternatives gave similar results. For example, one can take $\mathbf{p}_i(t+1) = \mathbf{p}_s(t)(1+\rho\lambda)$ where $\mathbf{p}_s$ is one of the surviving solutions. In figure 2, we show an example of a typical run of the reduced MA.

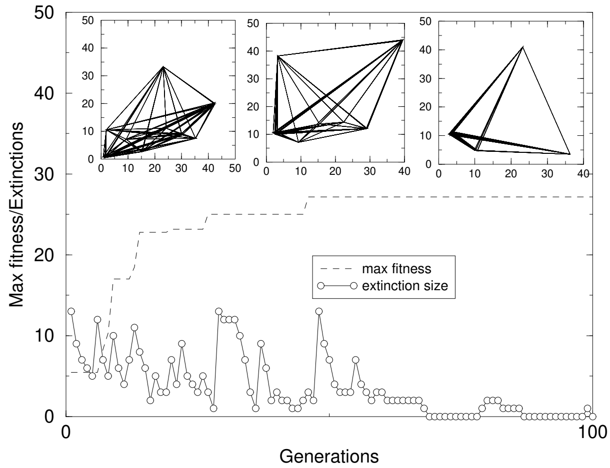

Figure 2: *An example of the time dynamics of the MA (reduced version). Here a two-dimensional landscape has been used. The function is of the form $f(x,y) = \sum_{j=1}^{S} A_j G_j(x,y)$ where $A_j \in (0.8, 1)$ and $G_j(x,y) = \exp(-((x-x_0)^2 + (y-y_0)^2)\sigma_j)$. The points (x_0, y_0) are randomly scattered in the $[0,50] \times [0,50]$ and $\sigma_j \in [10,100]$. The maximum is located at $(23,40)$ and is reached at the $G \approx 50$ generation using a small population of $P = 20$ solutions and $\rho = 0.2, \tau = 0.2$. The maximum fitness and the size of the extinction events are shown. Three examples of the location of the solutions at $t = 5, 25$ and 55 are shown (insets, from left to right). We can see that large extinctions are linked to increases in fitness.*

Although all essential rules defining the MA have been presented, several improvements and additional rules have been explored. In particular, we can decrease τ with time as in simulated annealing (Kirkpatrick et al., 1983) to get a good convergence. In this context, the "temperature" τ, when lowered, provides a decrease in randomness which favours the exploitation around the best individual found. In order to lower τ in each generation, we can use a given decreasing function. In our analysis, we have used a linear relation:

$$\tau(t;G) = 1 - \frac{t}{G} \tag{5}$$

where G is number of generations. The results of using this linear annealing procedure do not strongly differ from other choices of $\tau(t)$.

4 NUMERICAL RESULTS

A number of systematic explorations using MA, both (i) and (ii), have been performed [1] and compared with standard GA with tournament selection (Mitchell, 1996). In order to obtain a standard comparison with previous studies, we have introduced examples of d-input functions (proposed for real valued spaces in the contest helded during the 1996 IEEE International Conference on Evolutionary Computation):

(a) Griewank's function where $x_i \in [-600, 600]$ with $d = 10$:

$$f_1(\vec{x}) = -\frac{1}{4000}\sum_{i=1}^{d}(x_i - 100)^2 + \prod_{i=1}^{d}\cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1 \tag{6}$$

(b) Michalewicz's function where $x_i \in [0, \pi]$ with $d = 10$:

$$f_2(\vec{x}) = \sum_{i=1}^{d}\sin(x_i)\sin^{20}(\frac{ix_i^2}{\pi}) \tag{7}$$

(c) Rotated version of Michalewicz's function where $x_i \in [0, \pi]$ with $d = 10$ and $\alpha = \frac{\pi}{4}$:

$$f_3(\vec{x}) = f_2(Rotation(\vec{x})) \tag{8}$$

where *Rotation* means to perform $d-1$ rotations of α radians centered at point $(\frac{\pi}{2}, ^{(d}\ldots, \frac{\pi}{2})$ (see figure 3).

In experiments performed with GA's, we have used a *crossover probability* of 0.7 and calculated τ according to equation (5), and we have used a constant value for ρ: 0.3 for example (a), and 0.5 for exemples (b) and (c). Other parameters for these experiments are summarized in table 1.

The results of these experiments are shown in figure 4. Example (a) is a typical case in the standard performance of the MA's in relation to GA's. We can see a very rapid convergence of the MA's in relation with the smooth, slow approach of the GA. Both types of MA's show similar behavior, although the reduced version is typicaly faster and more efficient in terms of computational time. Example (b) is a special type of test function involving geometric features favouring GA methods. In this case GA's work better than MAs, but in fact a rotation of this function (c) leads to a different relation of efficiencies: GA's performance fall considerably.

[1]For all the numerical experiments, a Sun Ultra-1 workstations with SunOS operating system has been used

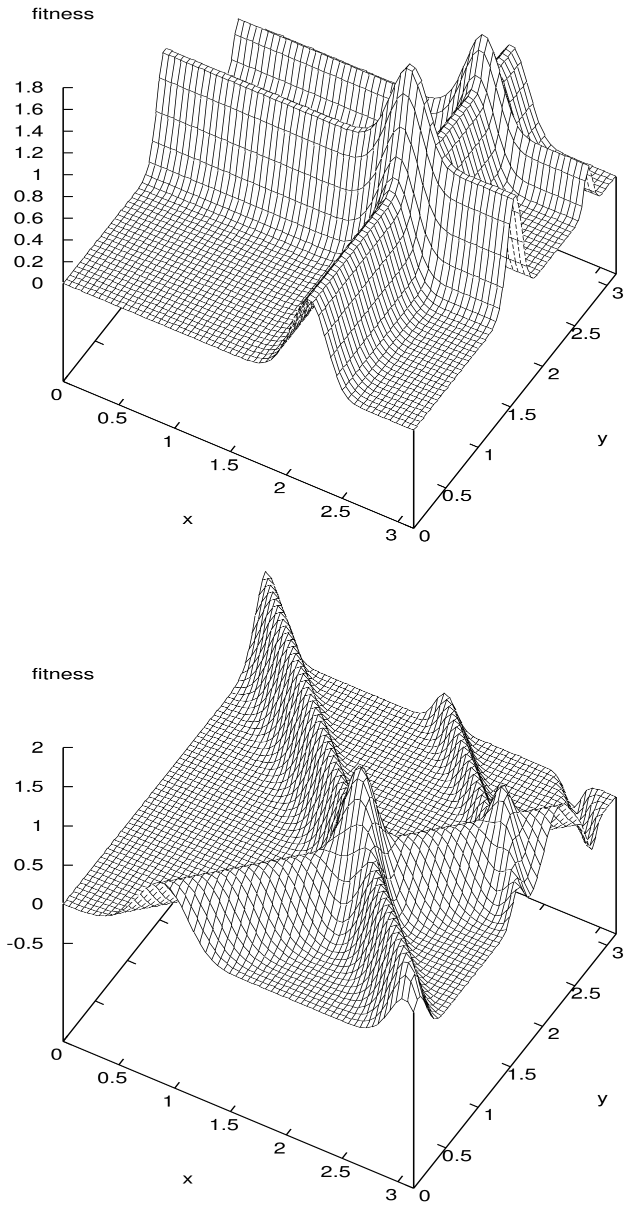

Figure 3: *Up: Graphical representation for $d = 2$ of Michalewicz's function —example (b). Down: Rotated version of Michalewicz's function —example (c).*

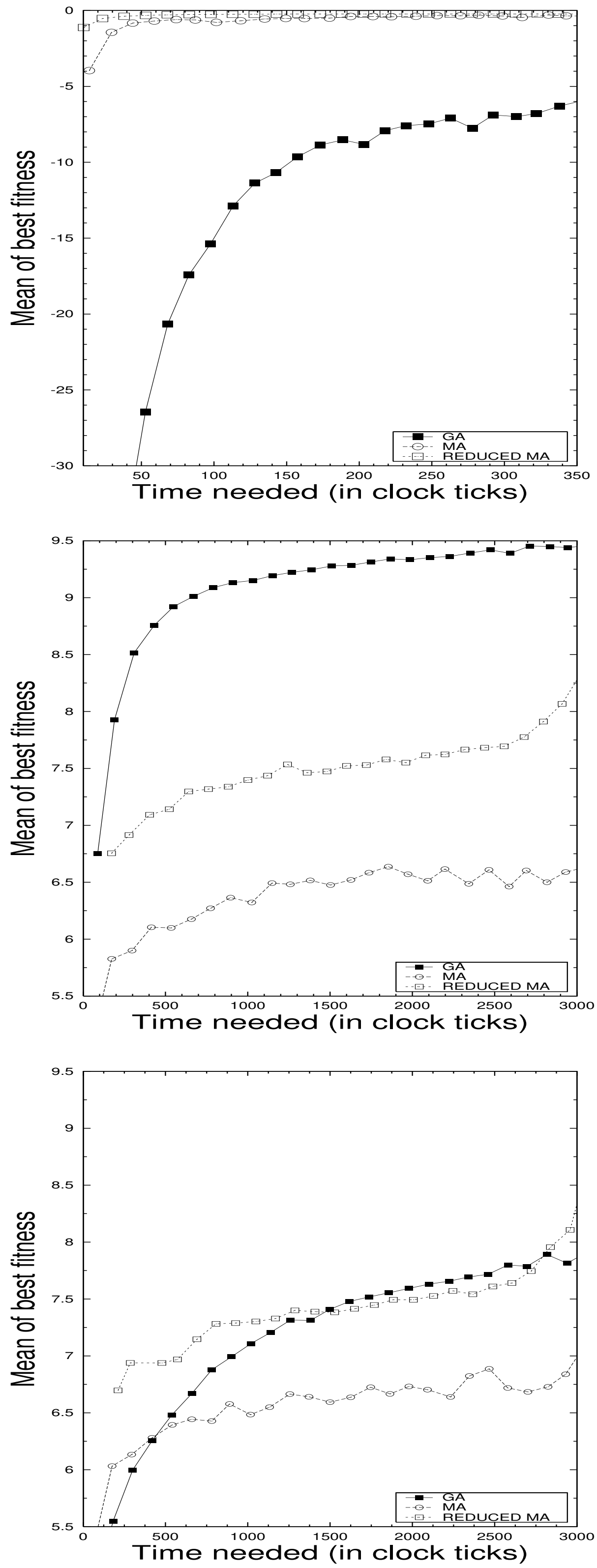

Figure 4: *Relation between fitness value reached and time needed, applying GA, MA and reduced MA to f_1, f_2 and f_3 (from top to bottom).*

Table 1: *Some parameter values for the performed experiments. Here P is population size, G is number of generations used, R number of runs and, ΔT is interval of time choosen to calculate the mean of fitness value. For each pair P and G value, 25 runs were performed.*

FUNC.	**ALG.**	**P**	**G**	**ΔT**
f_1	GA	20 to 150	10 to 500	15
f_1	MA	10 to 90	100 to 1200	15
f_1	Red.MA	20 to 800	100 to 2100	15
f_2,f_3	GA	20 to 330	50 to 250	120
f_2,f_3	MA	10 to 250	10 to 1250	120
f_2,f_3	Red.MA	70 to 400	500 to 7500	120

5 CONCLUSIONS

In this paper a novel optimization technique, which we have called macroevolutionary algorithm, has been explored using two different versions of the MA model. The MAs are based in a simple procedure inspired in macroevolutionary dynamics and extinction events. In the original model extinctions removed some species and new ones were generated by diversification of the survivors. In the MA approach, the basic ecology-like structure is also preserved, but now applied to a set of candidate solutions to a given optimization problem on a fitness landscape. The survival of solutions is linked with the fitness $f(\mathbf{x})$ of each species in relation with all the other species. If the total sum of input connections to a given species h_i is positive, it survives. If negative, it becomes extinct. Extinction events are typically linked with the finding of new, high-fit solutions and large (mass-) extinction events take place when a very good solution is found. The replacement process guarantees both the exploitation of the high-fit solutions as well as further, random exploration of other domains of the landscape. Because of the connection matrix, the whole population is able to obtain a rather accurated map of the relative importance of the solutions being explored in the landscape.

These algorithms are easily extended to optimization problems in high-dimensional parameter spaces. In a recent study, we have shown that its efficiency in neural network training problems (Marín and Solé, 1999). A specific example is shown in figure 4 for the two-spirals problem (Peretto, 1991). Since several standard combinatorial optimization problems as the traveling salesman can be mapped to a multidimensional fitness landscape (in terms of an energy/cost function

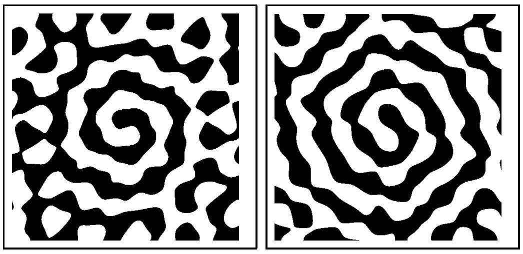

Figure 5: *Generalization performed from two-spirals problem with a 2-16-1 neural networks architecture and sinusoidal function in hidden layer. Left: Using back-propagation. Right: Using MA.*

plus constraints) the MAs also offers a new approach to such area of research.

Acknowledgements

The authors would like to thank Melanie Mitchell and the members of Complex Systems Research Group in Barcelona for useful comments and suggestions. This work has been supported by a grant PB97-0693 and by the Santa Fe Institute (RVS).

References

T. Bäck, U. Hammel and H-P Schwefel, *Evolutionary Computation: Comments on the History and Current State. IEEE Transactions on Evolutionary Computation*, Vol. 1, N. 1, April 1997.

D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

J. H. Holland, *Adaptation in Natural and Artificial Systems.* MIT Press, 1992.

S. Kirkpatrick, C. D. Gelatt and M.P. Vecchi. Science 220, 671, 1983.

J. Marín and R. V. Solé, *Macroevolutionary Algorithms: A New Optimization Method on Fitness Landscapes.* Santa Fe Institute working paper 98-11-108, 1998. To appear in IEEE Transactions on Evolutionary Computation.

J. Marín and R. V. Solé, *Macroevolutionary Algorithm as Neural Network Training Method.* In preparation, 1999.

M. Mitchell, *An Introduction to Genetic Algorithms.* MIT Press, 1996.

R. Palmer, *Optimization on Rugged Landscape in: Molecular Evolution on Rugged Landscape: Proteins, RNA and the Immune System.* Addison-Wesley, 1991.

P. Peretto, *An Introduction to the Modeling of Neural Networks.* Cambridge University Press, 1991.

R. V. Solé and S. C. Manrubia, Phys. Rev. E 54, R 42, 1996.

R. V. Solé, J. Bascompte and S. C. Manrubia. Proc. Roy. Soc. London B., 263 1407, 1996.

R. V. Solé, S. C. Manrubia, M. Benton and P. Bak, *Self-similarity of Extinction Statistics in the Fossil Record.* Nature, Vol. 338, 21 August 1997, pp 764-767.

G. Weisbuch, *Complex Systems Dynamics.* Santa Fe Studies Series II, Addison-Wesley, 1991.