Annealing schedule from population dynamics

Stefan Bornholdt*
Institut für Theoretische Physik, Universität Kiel, Leibnizstrasse 15, D-24098 Kiel, Germany

(Received 23 June 1998; revised manuscript received 30 November 1998)

We introduce a dynamical annealing schedule for population-based optimization algorithms with mutation. On the basis of a statistical mechanics formulation of the population dynamics, the mutation rate adapts to a value maximizing expected rewards at each time step. Thereby, the mutation rate is eliminated as a free parameter from the algorithm. [S1063-651X(99)06004-3]

PACS number(s): 05.50.+q, 87.10.+e, 07.05.Mh, 02.60.Pn

I. INTRODUCTION

Population-based optimization algorithms [1–3] have been successfully applied to problems in physics [4,5] and beyond [6]. This class of algorithms is based on the simultaneous tracking of more than one point in search space (a ''population,'' in analogy to biological evolution [7]), in order to make trapping in local optima less likely during the process of optimization. In addition, stochastic noise is used to generate random displacements of the search points (''mutations''), performing a local optimization. A major problem in these algorithms is the adjustment of the noise level for a given optimization task. In the beginning of the search, high noise levels help to identify promising regions of the search space, while for a subsequent fine tuning low noise works best. This problem is well known from simulated annealing [8], an optimization algorithm where noise is introduced by means of a formal temperature. Lowering, or ''annealing,'' the temperature from high to low values in the course of the optimization leads to improved results compared to an optimization at fixed temperatures. However, there remains the problem of choosing a suitable annealing schedule for a given optimization problem [8]. The same problem occurs in population-based optimization algorithms, and will be addressed in the remainder of this paper.

For some population-based algorithms, heuristics have been proposed that adjust the noise rate during optimization. For example, when noise is implemented as random steps of fixed Euclidean distance in the search space, its step size can be adjusted according to an estimate for the most promising next step on the basis of the previous one [2]. Another approach, taken by Davis [9], uses a set of noise operators competing for high scores in producing low energy search points. Although such approaches work in practice, they have not been formally established on the basis of a dynamical formulation of the algorithm. A major problem is the enormous complexity of the dynamics of population-based algorithms.

Recently, this problem has attracted parts of the physics community, applying statistical mechanics methods to the algorithm dynamics. Prügel-Bennett and Shapiro [10] described the average population dynamics in terms of the distribution of energy values in the population at each time step. The observables are the cumulants of the energy distribution of the population. Selection of the ''fittest'' (the low energy members) is completely defined in terms of these variables. For certain energy functions, this enables a prediction of the algorithm dynamics to high accuracy over large numbers of generations [11]. In Ref. [12] it was proposed to use this formalism in order to determine an annealing schedule from the predicted dynamics. However, each of the two immediate routes faces a major obstacle: The analytical approach is only feasible for exactly known energy functions with simple properties, and involves a complicated maximum entropy calculation. The alternative way via measuring the current cumulants during evolution is spoiled by large sample-to-sample fluctuations. Here we propose a model which, though inspired by this, does not have to deal with these problems. For an *a priori* unknown energy function (being the usual case when dealing with optimization problems) a less formal framework is needed. This is supplied by a dynamical model based on energy correlation formulated in Ref. [13]. It will be used here to predict an optimal noise rate by maximizing the expected performance of the algorithm in each time step. In the following we will define two test functions and two algorithms to be considered. A model for the prediction of the mutation effects will then be given. An improved algorithm will be defined on the basis of this model. It is then applied to the test functions and its performance compared to standard versions of the respective algorithms.

II. TEST SYSTEM

Let us consider an optimization problem in terms of the minimization of a real valued function $E(S)$ on a binary search space. Its value is the energy of the test point S, which is to be minimized, equivalent to a search for the ground state energy of a physical system [14]. In biological terms this corresponds to the negative ''fitness'' of an organism to be maximized for survival. The discrete search space will be parametrized through the binary representation $S\in\{\pm 1\}^N$ of length N.

Two functions serve as examples: one purely additive energy function, and another with many local optima. The first problem is a random field paramagnet

$$E_\alpha=\sum_{i=1}^{N} J_i S_i^\alpha+\kappa_1^0, \tag{1}$$

*Electronic address: bornholdt@theo-physik.uni-kiel.de with random couplings J_i taken from a Gaussian distribution with mean 0 and variance 1. The N spins S_i^α with $i=1,\ldots,N$ and $S_i^\alpha=\pm 1$ form the genetic string of the member α of the population. The second function is the NK model energy function [15] as an example for a hard search problem with many local minima. It is defined through

$$E_\alpha=\sum_{i=1}^{N} E_i(S_i^\alpha;S_{i_1}^\alpha,\ldots,S_{i_K}^\alpha), \tag{2}$$

with 2^{K+1} random energy values $E_i(S^\alpha)$ drawn from a uniform distribution over the interval [0,1] and a randomly chosen permutation of sites i_1 to i_K, both for each i. Originally, this function has been formulated for the study of evolution on tunably rugged energy landscapes with application to the evolution of the immune response [16]. These functions will be minimized by means of a population-based algorithm which is defined as follows: First, a random ensemble of search points S_α with $\alpha=1,\ldots,P$ and energies $E_\alpha=E(S_\alpha)$ is chosen (a ''population''). One time step of the algorithm consists of the following procedure: (1) Select the member with the lowest energy. (2) Reproduce it once. (3) Replace the member with the highest energy by the new copy. (4) ''Mutate'' all members except the original one with the lowest energy by inverting each spin with a small fixed probability γ. Repeating these steps then forms an evolutionary algorithm searching for low lying energy states. It is driven by selection lowering the mean energy of the population and mutation increasing the variance.

For comparison let us also look at a simplified algorithm, a stochastic gradient descent. After the initial population is chosen as above, the following steps are taken: (1) Select the member with the lowest energy. (2) Reproduce it $P-1$ times. (3) ''Mutate'' the new copies by inverting each spin with a small fixed probability γ. (4) Replace all members, except the one with the lowest energy, by the mutated copies. Here the offspring of the best member takes over the entire population in each time step.

III. MODELING THE ALGORITHM DYNAMICS

We will model the dynamics of these algorithms in terms of the energy distribution $\rho(E)$ of the population expressed as an expansion in cumulants. The energy distribution $\rho(E)$ of a population is the natural quantity for the selection operator, which solely acts on the energy values of the search points. The expansion in cumulants of the energy distribution $\rho(E)$ has been shown to be a useful approximation for population-based algorithms [11]. At each time step, the evolving population is then approximated by a set of these variables. When also modeling the mutation operator, one has to be more careful. Mutation, acting on the underlying representation instead of the energy values themselves, requires additional assumptions to model its dynamics in terms of $\rho(E)$. For example, one could obtain a maximum likelihood estimation for the underlying spin states corresponding to a given energy cumulant, and use this to calculate the expected effect of the mutation operator. However, such a procedure requires simple energy functions to allow for the calculation, and, of course, a complete knowledge of the energy function, which is usually not available for realistic optimization problems.

Here we use a different approach which is based solely on a phenomenological parameter that is accessible by measurement if the energy function is not known [13]. In particular, we will use a model for the lowest order average dynamics of the mutation operator on the basis of the energy correlation m of mutation on a given energy landscape $E(S)$,

$$m=\frac{\langle E_\alpha E_\alpha^m\rangle_{\alpha,\mathrm{mut}}-\langle E_\alpha\rangle_\alpha\langle E_\alpha^m\rangle_{\alpha,\mathrm{mut}}}{\langle E_\alpha^2\rangle_\alpha-\langle E_\alpha\rangle_\alpha^2}, \tag{3}$$

where $\langle\,\rangle_\alpha$ denotes an average over the population, and $\langle\,\rangle_{\mathrm{mut}}$ an average over all possible mutation events. The energy correlation m is a measure of how strongly, on average, the energy of a mutant is correlated to that of its parent, for a given mutation operator applied to a given energy function. Such correlations form the backbone on which the search process in mutation-based algorithms proceeds. Energy correlations can be measured for many hard optimization problems, as were recently classified in Ref. [17]. In this framework, the energy distribution of a population after mutation can be approximated by its cumulants as a function of m,

$$\begin{aligned}
\kappa_1^m&=m\kappa_1+(1-m)\kappa_1^0,\\
\kappa_2^m&=m^2\kappa_2+(1-m^2)\kappa_2^0,\\
\kappa_3^m&=m^3\kappa_3,\\
\kappa_4^m&=m^4\kappa_4,
\end{aligned} \tag{4}$$

where κ_1^0 and κ_2^0 are the energy mean and variance of a random initial population. This model was derived in Ref. [13]. The underlying assumption of this model is that the population of the algorithm (not the landscape itself) can be expanded in cumulants around a Gaussian. In fact, one observes that the initial random population for many real optimization problems fulfills this requirement well. In the framework of this expansion, the above model uses m to predict the expected energy distribution of the population in the next time step. Such a model, based on energy correlations, further assumes that the lowest order correlation m contains major information about the average effect of the mutation operator. It has been shown to be useful to describe the dynamics of a population-based algorithm over at least 200 generations, both for correlated and poorly correlated landscapes [13].

How can such a model be used to improve an optimization algorithm? Let us look at a numerical example for the dynamics of a stochastic gradient descent under a fixed mutation rate γ, as shown in Fig. 1. Optimization of the first test function (1) is shown with a stochastic gradient descent, searching for the minimal energy configuration of a random paramagnet of $N=128$ spins in an external field. For a large mutation rate γ one sees that the early gain is large, whereas for small γ, as shown by the solid curve, a poor early gain is balanced later by a slow but steady improvement. For optimization problems involving computationally costly energy evaluation, this behavior poses a severe problem. Knowledge of the latter stages of the dynamics would be needed at the

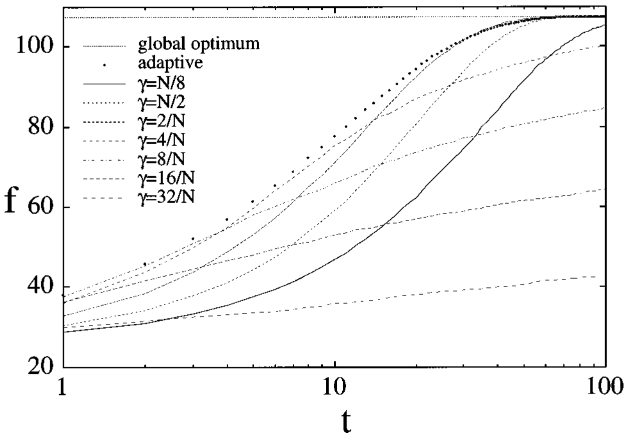

FIG. 1. The evolution of the member with maximum fitness $f = -E_{\min}$ is shown at different fixed mutation rates γ of a stochastic gradient descent for the random paramagnet. In comparison, the points show the dynamics of the adaptive mutation algorithm. In all simulations a quenched average over 200 runs is shown, with a random energy function chosen once. The dotted line denotes the global optimum of the function.

beginning in order to be able to choose an optimal γ. In the following, this problem will be addressed through a variable mutation rate $\gamma(t)$, that combines the advantages of both regimes of the mutation rate γ.

IV. ANNEALING THE MUTATION RATE

For this purpose, the expected best member of a population after mutation $\langle E_{\min}\rangle$ is evaluated on the basis of the energy distribution $\rho^m(E)$ of the population after mutation given in terms of cumulants κ_i^m. The expectation value for the lowest energy occurring in a set of P samples [18] drawn from the post-mutation distribution $\rho^m(E)$ is

$$\langle E_{\min}\rangle = P\int_{-\infty}^{\infty} dE_1 E_1 \rho^m(E_1) \prod_{n=2}^{P} \int_{E_1}^{\infty} dE_n \rho^m(E_n). \quad (5)$$

In the Gaussian approximation, a saddle point expansion yields, to leading order,

$$\langle E_{\ \min}\rangle = \kappa_1^m - \sqrt{2\kappa_2^m \ln P}. \quad (6)$$

Inserting the post-mutation distribution (4), the expected energy of the best member after mutation $\langle E_{\min}\rangle$ can be minimized in terms of m. The resulting mutation correlation m_{opt} is then used to choose the mutation rate γ in the forthcoming mutation step, thereby optimizing the expected best member of the next generation. Unfortunately, this method is plagued by large fluctuations in the measured moments of the energy distribution.

Therefore, let us first look at the expected dynamics of the stochastic gradient descent where this problem does not occur. Following Eq. (4), the energy distribution after the mutation step is given in the Gaussian approximation by

$$\kappa_1^m = mE_{\min}(t) + (1-m)\kappa_1^0,$$

$$\kappa_2^m = (1-m^2)\kappa_2^0. \quad (7)$$

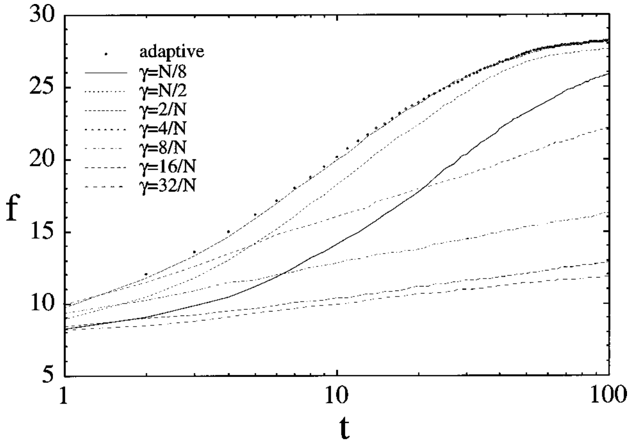

FIG. 2. Same as Fig. 1 for an evolution on the rugged landscape of the NK-model energy function with $K = 8$.

Inserting this into Eq. (6), and minimizing the expected best member of the next generation $\langle E_{\min}(t+1)\rangle$ with respect to m, yields an estimate for an optimal correlation:

$$m_{\text{opt}} = \sqrt{\frac{(E_{\min}(t) - \kappa_1^0)^2}{2\kappa_2^0 \ln(P) + (E_{\min}(t) - \kappa_1^0)^2}}. \quad (8)$$

This is subsequently translated into an optimal mutation rate γ_{opt} via

$$m = 1 - 2\gamma, \quad (9)$$

which is derived from the known energy function. Each time step of the modified algorithm can now be described as follows: (1) Determine the lowest energy $E_{\min}$ in the population. (2) Calculate the optimal correlation m_{opt} from Eq. (8) and calculate the mutation rate γ_{opt} from this. (3) Select the member with the lowest energy. (4) Reproduce it $P-1$ times. (5) ''Mutate'' the new copies by inverting each spin with the mutation rate γ_{opt}. (6) Replace all members, except the one with the lowest energy, by the mutated copies. Starting from an initial condition as above and iterating this step results in an algorithm with an adaptive mutation rate. How it applies to the above test function is shown in Fig. 1. At each time scale, the evolution of the lowest energy member of the evolving population compares well to the respective best ''fixed mutation rate algorithm.'' No explicit knowledge of favorable ranges of the mutation rate γ is used, thus removing the free parameter γ from the algorithm. Applying the formalism to the NK-model function (2), and using the relation between parent child correlation m and mutation rate γ, derived as

$$m = (1-\gamma)^{K+1}, \quad (10)$$

a comparable result is obtained (Fig. 2).

A similar procedure can also be carried out for the full population-based algorithm with sparse replication. Again, Eq. (4) is used to adjust the mutation rate in the next generation to a value that maximizes the expected gain. In order to avoid large fluctuations, which would be incompatible with a smooth evolution, we do not base the prediction on the cumulants of the current energy distribution in the population, but rather on $E_{\rm min}$ alone. This is done in the spirit of Eq. (7), which is less likely to fluctuate than the prediction based on the full cumulants κ_i. However, $E_{\rm min}$ still relates to the dynamics of a mixed population, and proves to be useful in modeling the population dynamics under mutation. Depending on the mutation strength, a number of former mutants are still correlated with the new offspring, in addition to the one copy of $E_{\rm min}$ made per generation. Let us assume that a number of M members of the population are strongly correlated with the new offspring. For simplicity we further assume that the remaining members are completely uncorrelated and treat them as random. In this approximation, the integral for the expected best member of a population can be written as

$$\langle E_{\rm min}\rangle = M\int_{-\infty}^{\infty} dE_1 E_1 \rho^m(E_1)\times\left[\int_{E_1}^{\infty} dE_2\rho^m(E_2)\right]^{M-1}\left[\int_{E_1}^{\infty} dE_3\rho^0(E_3)\right]^{P-1-M} + (P-1-M)\int_{-\infty}^{\infty} dE_1 E_1\rho^0(E_1)\times\left[\int_{E_1}^{\infty} dE_2\rho^m(E_2)\right]^{M}\left[\int_{E_1}^{\infty} dE_3\rho^0(E_3)\right]^{P-2-M}. \tag{11}$$

It is solved using a saddle point expansion in the Gaussian approximation, considering the limit where the distributions ρ^m and ρ^0 move sufficiently apart from each other (due to $E_{\rm min}$ moving away from the random population distribution), where one can neglect their mutual variations. One obtains

$$\langle E_{\rm min}\rangle = \kappa_1^m - \sqrt{2\kappa_2^m \ln(M-1)} + \kappa_1^0 - \sqrt{2\kappa_2^0 \ln(P-2-M)}. \tag{12}$$

The expected $\langle E_{\rm min}(t+1)\rangle$ of the next generation based on Eq. (7) is then minimized by the mutation rate

$$m_{\rm opt} = \sqrt{\frac{(E_{\rm min}(t)-\kappa_1^0)^2}{2\kappa_2^0\ln(M-1) + (E_{\rm min}(t)-\kappa_1^0)^2}}. \tag{13}$$

Finally, the number of correlated members M in the population remains to be specified. For a lowest order estimate let us consider a member with energy $E_{\rm min}$ and mutate it k times. We then require that its energy does not, on average, move away more than $\sqrt{2\kappa_2^m}$ from the current value of $E_{\rm min}$, i.e.,

$$E_{\rm min} + \sqrt{2\kappa_2^m} > m^k E_{\rm min} + (1-m^k)\kappa_1^0. \tag{14}$$

The exact limit for the number of subsequent mutations k depends on the current details of the energy values in the population. However, when using Eq. (13) as an estimate for the current value of m, the energy value of a mutant decorrelates after only a few mutation steps. Therefore, $\ln(M-1)$ is estimated to be of the order of 1 and we determine the optimal mutation rate in the algorithm using

$$m_{\rm opt} = \sqrt{\frac{(E_{\rm min}(t)-\kappa_1^0)^2}{2\kappa_2^0 + (E_{\rm min}(t)-\kappa_1^0)^2}}. \tag{15}$$

This expression is now used for annealing the mutation rate in the population based algorithm. The modified time step of the algorithm is defined by the following procedure: (1) Determine the lowest energy $E_{\rm min}$ in the population. (2) Calculate the optimal correlation $m_{\rm opt}$ from Eq. (15), and calculate the mutation rate $\gamma_{\rm opt}$ from it. (3) Select the member with the lowest energy. (4) Reproduce it once. (5) Replace the member with the highest energy by the new copy. (6) ''Mutate'' all members except the original one with the lowest energy by inverting each spin with the probability $\gamma_{\rm opt}$. Again starting from an initial condition as above and iterating this step results in an algorithm with annealed mutation rate. In Fig. 3 the evolution of the best population member on the basis of this algorithm is compared to runs with fixed mutation rates. The algorithm adjusting the mutation rate compares well to the fixed mutation rate cases at each stage of evolution. In Fig. 4 the algorithm is applied to the NK-model function with similar results. For any given resource of CPU time, one reaches a level of performance comparable to an optimum fixed mutation rate (at the given total evolution time). This is helpful in optimization when the relationship between mutation rate γ and the algorithm dynamics at later times is *a priori* unknown.

V. DISCUSSION

For both algorithms considered above, we have seen how annealing the mutation rate can be based on a simple dynamical model based on the energy correlation of the muta-

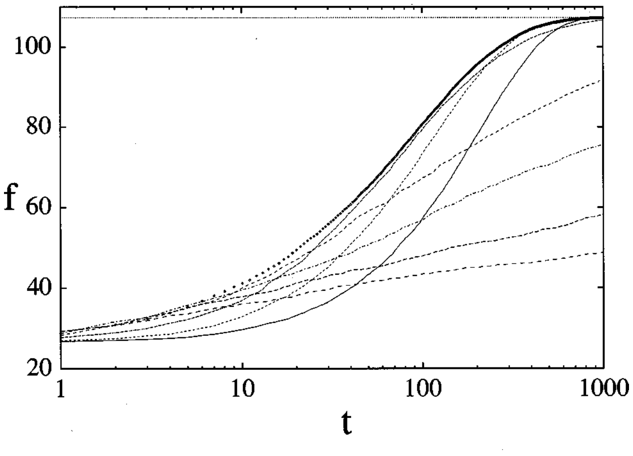

FIG. 3. Adaptive mutation in the population-based algorithm compared to the fixed mutation case for a random paramagnet, with conventions chosen as in the previous figures. The dotted line denotes the global optimum of the function.

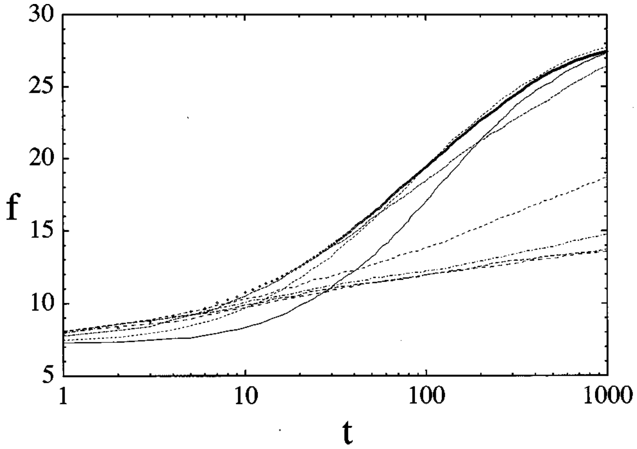

FIG. 4. Adaptive mutation in the population-based algorithm compared to a fixed mutation rate for the NK model.

tion operator. In the presented examples, functions with known analytical properties have been considered, enabling a direct calculation of the mutation correlation $m(\gamma)$. However, when applying the above method to general optimization problems, this functional dependence remains to be established. For many realistic optimization problems it is well approximated by a monotonic function with a simple decay law in the small γ regime, as classified in Ref. [17] for a number of different optimization problems. For many problems it can be modeled using the simple linear approximation $\gamma(m)=1-xm$. In order to apply the above algorithms to optimization problems where the energy function is not known, a heuristics that measures this relation for a given problem has been defined. One possibility is to measure m and improve the estimate for x during the run of the above algorithms. This procedure can be defined as follows: (1) Start from an initial estimate for x. (2) Measure the mutation correlation m during each time step of the algorithm using Eq. (3). (3) Use the measured m to improve the estimate for x in the linear approximation (taken as the average over all measured values of x so far). This allows one to apply the method to energy functions with no *a priori* knowledge of their correlation structure. This method has been sucessfully tested using the two energy functions of this study.

Several extensions remain to be studied, e.g., algorithms where recombination, or ''crossover,'' is present. In such algorithms, the annealed mutation as described here is expected to work equally well as long as the mutation step does not strongly interact with the crossover. Whether the recombination strength can be adapted in a similar way is an open question. Another free parameter is introduced by selection, namely, selection strength. Here a one parameter model exists [10], and an adaptive adjustment could be discussed as well.

To summarize, we proposed a mechanism for annealing the mutation rate in population-based algorithms. It is based on a statistical mechanics model of the population dynamics and a correlation measure of the mutation operator. The mutation rate γ thereby drops out as a free parameter of the algorithm.

ACKNOWLEDGMENTS

The author thanks J. L. Shapiro for a useful discussion on this subject, and, for further stimulating discussions, M. Mitchell and R. G. Palmer. The hospitality and support of the Santa Fe Institute as well as the financial support of the Deutsche Forschungsgemeinschaft is greatly acknowledged.

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975).

[2] I. Rechenberg, *Evolutionsstrategie 94* (Frommann-Holzboog, Stuttgart, 1994).

[3] D. B. Fogel, *Evolutionary Computation* (Addison-Wesley, Reading, MA, 1995); M. Mitchell, *An Introduction to Genetic Algorithms* (MIT Press, Cambridge, MA, 1995).

[4] P. Sutton, D. L. Hunter, and N. Jan, J. Phys. I **4**, 1281 (1994).

[5] F. Starrost, S. Bornholdt, S. Solterbeck, and W. Schattke, Phys. Rev. B **53**, 12 549 (1996).

[6] See, for instance, *Proceedings of the 1997 IEEE Conference on Evolutionary Computation (ICEC 1997)* (IEEE, Piscataway, 1997); *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA, 1997)*, edited by T. Baeck (Morgan Kauffman, San Francisco, 1997).

[7] C. R. Darwin, *The Origin of Species* (Penguin, Harmondsworth, 1968; original edition 1859).

[8] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines—A Stochastic Approach to Combinatorial Optimization and Neural Computing* (Wiley, New York, 1989).

[9] *Handbook of Genetic Algorithms*, edited by L. Davis (Van Nostrand Reinhold, New York, 1991).

[10] A. Prügel-Bennett and J. L. Shapiro, Phys. Rev. Lett. **72**, 1305 (1994).

[11] A. Prügel-Bennett and J. L. Shapiro, Physica D **104**, 7451 (1997).

[12] J. L. Shapiro and A. Prügel-Bennett, Lect. Notes Comput. Sci. **993**, 14 (1995).

[13] S. Bornholdt, Phys. Rev. E **57**, 3853 (1998).

[14] In general, one has to be careful to carry the analogy further to a physical system, since the dynamics does not have to follow all major principles of known dynamics in a physical system. That is, detailed balance may well be violated in optimization, while ergodicity remains essential.

[15] S. Kauffman and E. Weinberger, J. Theor. Biol. **141**, 211 (1989).

[16] S. Kauffman, E. Weinberger, and A. S. Perelson, in *Theoretical Immunology*, edited by A. S. Perelson (Addison-Wesley, Reading, MA, 1988).

[17] P. Stadler, in *Complex Systems and Binary Networks*, edited by R. Lopez-Pena, R. Caporilla, R. Garcia-Pelazo, H. Waelbroeck, and F. Zertude (Springer-Verlag, Berlin, 1995).

[18] L. M. Rattray, Complex Syst. **9**, 213 (1995).