Ensemble-based control of evolutionary optimization algorithms

Axel Reimann and Werner Ebeling
Humboldt-University Berlin, Institute of Physics, D-10115 Berlin, Germany

(Received 27 July 2001; published 20 March 2002)

This paper presents a study on how the intrinsic search parameters of an evolutionary optimization algorithm can be automatically controlled. It will be shown that only a small search parameter window ensures good optimization results. This evolutionary window, enclosing effective values for the mutation rate and temperature, can be adapted to by carefully steering the ensemble's fitness dispersion. A control sensor based on an entropy measure is introduced to achieve this goal. The efficiency of the proposed control method will be tested by optimizing artificial sequences such as the well-known low autocorrelated binary strings and natural sequences including RNA.

DOI: 10.1103/PhysRevE.65.046106 PACS number(s): 02.50.Ey, 05.10.Gg

I. INTRODUCTION

Evolutionary algorithms have proven to be powerful instruments when it comes to complex and high-dimensional optimization problems. They are still not in widespread use, however, for the following reasons: Evolutionary algorithms are designed to find good solutions for a large class of problems, while there is no guarantee that the found solution is the best possible. What makes it particularly difficult to use evolutionary algorithms is the strong dependence of the solution quality on intrinsic search parameters such as ensemble size or mutation rate. So, in order to open a wider application field for these algorithms, the algorithm-internal (nonproblem related) search parameters should be hidden from a potential user and automatically controlled to ensure search efficiency. While there are many proposals regarding temperature control, for example, in "simulated annealing" strategies [1,2], this paper focuses on the mutation and selection rates.

After a short introduction to the mathematical model of a certain class of mixed evolutionary algorithms, this work proposes a possible way to adjust its search parameters and presents results obtained for frustrated periodic sequences, the LABS problem, and RNA secondary structures.

II. MODELING MIXED EVOLUTIONARY STRATEGIES

The mathematical model used has already been introduced in an earlier work [3]. Here, a short survey might suffice. Analyzing the mechanisms of natural evolution we find several basic strategies [4–9]. The most important are biological and thermodynamical strategies.

Biological strategies in the universe appear in the process of biogenesis only, i.e., about three to four billion years ago. The basic elements of a Darwin-type strategy are [4–7] (i) self-reproduction of good species that show maximal fitness, (ii) mutation processes due to error reproductions that change the phenotype's properties of the species.

Thermodynamical strategies rely on the second law. Macroscopic physical systems optimize certain thermodynamic functions in their course of evolution. Simplifying, we will call this strategy the Boltzmann strategy. The Boltzmann strategy has the following two basic elements: (i) motion along gradients to reach steepest ascent/descent of thermodynamic functions; (ii) stochastic processes including thermal and hydrodynamic fluctuations leading to random changes; thus avoiding locking in local maxima. In order to formulate a simple dynamic model of a mixed strategy, let us consider a numbered set of states $i=1,2,\ldots,s$—each characterized by a potential energy U_i that is to be minimized in the search and a relative frequency in the population $x_i(t)$ at time t. Then, the simplest model of a mixed strategy that tends to find minima of U_i is described by the following equation:

$$\frac{d}{dt}x_i(t)=\underbrace{\gamma(\langle U\rangle-U_i)x_i(t)}_{\text{selection term}}+\underbrace{m\sum_{j=1}^{s}[A_{ij}x_j(t)-A_{ji}x_i(t)]}_{\text{mutation term}}. \tag{1}$$

The matrix A_{ij} describes mutation processes that are modeled as transitions from state j to state i. Therefore, A_{ij} contains the respective transition rates, while the x_i denote the occupation numbers of the states i. For simplicity we demand

$$\sum_{i=1}^{s}x_i(t)=x_0=\text{const.}, \tag{2}$$

i.e. we set the seeker ensemble size constant throughout the optimization process. The selection strength in Eq. (1) is denoted by the parameter γ and the mutation rate is labeled m. The transition rates A_{ij} separate into a symmetrical component $A_{ij}^0\equiv A_{ji}^0$ describing a biological strategy and an asymmetrical term introducing thermodynamic components:

$$A_{ij}=A_{ij}^0\begin{cases}1 & \text{if } \Delta U\leqslant 0,\\ \exp[-\beta\,\Delta U] & \text{if } \Delta U>0.\end{cases} \tag{3}$$

The introduced term ΔU depends on the problem type. For maximization tasks it is defined as $\Delta U:=U_j-U_i$. For minimization problems on the other hand it is defined as $\Delta U:=U_i-U_j$. Thus, the asymmetric term ensures that indifferent transitions and transitions leading to a solution improvement are always carried out—whereas "mis-steps" occur only with a small probability decreasing exponentially with the threshold's height. It is easy to see that Eq. (1) contains the pure Boltzmann strategy for $\gamma=0$ as a special case. The Darwin strategy on the other hand is obtained for $m=1$, $\beta\rightarrow 0$.

The selection is certainly not required to be linear (e.g., fitness proportional). In fact, hard selection as described below has turned out to yield better results in our simulations at equal simulation time and for the respective optimal parameter settings (mutation rate, temperature, and ensemble size).

In a generalization, we substitute the quasilinear selection term in Eq. (1) by a nonlinear selection and postulate:

$$\frac{d}{dt}x_i(t)=\gamma\sum_{j=1}^{s} F(\Delta U)x_i(t)x_j(t) + m\sum_{j=1}^{s}[A_{ij}(\beta)x_j(t)-A_{ji}(\beta)x_i(t)]. \tag{4}$$

Here, the nonlinear selection function $F(\Delta U)$ is monotonically decreasing. For numerical simulations we use

$$F(\Delta U)=\text{const}-\Theta(\Delta U), \tag{5}$$

where $\Theta(\Delta U)$ is the step function. This equation describes a selection scheme we call *tournament selection* since, in a selection process, the worst seeker in a randomly drawn tournament ensemble of size 2 gets replaced by the best one in the tournament ensemble.[1]

III. MODEL PROBLEMS

In order to make statements about an optimization strategy's efficiency it needs to be applied to test problems. Here, we will use the same test problems already introduced in Ref. [3]; so we will only briefly recall them again.

A. Frustrated periodic sequences

Frustrated periodic sequences (proposed by Engel [4]) are built using an alphabet of four letters, e.g., $l_i\in\{A,B,C,D\}$. A certain fitness value can be assigned to a given sequence by crediting alphabetic pairs, i.e.,

$$l^{\alpha}\in\{\{A,B\},\{B,C\},\{C,D\},\{D,A\}\}$$

or, additionally, periodic occurrences n^p of the same letter with period p:

$$f=\sum_i (l_i^{\alpha}+b\,n_i^p). \tag{6}$$

The problem becomes maximally frustrated—that is, we have two contradictory optimization goals—if the parameter b is chosen to be $b=1/p$. In our simulations these parameters were chosen to be $p=5$ and $b=1/5$.

TABLE I. The mutation operator used for RNA secondary structure optimization implements and allows single as well as multiple bind/dissolve operations.

Mutation operator	Operation
Connect	. . . .→ . . . (. . .) . . .
Disconnect	. . . (. . .) . . . → . . . .
Zip	. . . (. . .) . . . →.(((. . .)))
Unzip	.(((. . .)))→ . . . .

B. Low autocorrelation binary strings

This model is well known in literature and has already been studied by Golay *et al.* [10–12]. The binary strings S are composed of $+1$ and -1 bits

$$S=\{s_1,s_2,\ldots,s_L\},\quad s_i\in\{-1,+1\}. \tag{7}$$

The autocorrelation coefficient R for distance k is given by

$$R_k=\sum_{i=1}^{L-k} s_i\,s_{i+k}. \tag{8}$$

The aim is to minimize the quadratic sum E of all autocorrelation coefficients

$$E=\sum_{k=1}^{L-1} R_k^2 \tag{9}$$

or equivalently maximize the so-called merit factor F,

$$F=\frac{L^2}{2\,E}. \tag{10}$$

For most (but not all) odd length sequences, the highest merit factor is achieved by skew symmetric configurations. Skew symmetric sequences fulfill the relation

$$s_{\mu+i}=(-1)^i s_{\mu-i},\quad \mu=\frac{L+1}{2} \tag{11}$$

and therefore have $R_k=0$ for all odd k.

C. RNA secondary structure

We investigate four-letter $\{A,C,G,U\}$ RNA sequences. The different bases can form pairings known as Crick-Watson pairs (A,U), (G,C) and a so-called wobbled pair (G,U). The minimum free-energy formation of pairings is sought after as the "optimal" secondary structure. (A detailed introduction to RNA landscapes can be found e.g., in Ref. [13].) To evaluate foldings we make use of the RNAeval subroutine contained in the publicly available VIENNA-RNA package, version 1.4. The sequence used for testing are the first 100 base pairs of polio virus type 1 Mahoney (AC V01148) as used e.g. in Ref. [14].

The mutation operator used in the numerical simulations was designed to allow single, as well as multiple, connect/

[1] In our simulations we use a generalization to four seekers.

disconnect operations at once (cf. Table I), while avoiding the generation of not yet tractable pseudoloops.

IV. SEARCH PARAMETERS AND ENSEMBLE DISPERSION

The mixed evolutionary algorithms introduced in Sec. II basically have three intrinsic search parameters: ensemble size N (a.u.), temperature T (a.u.), and a mutation probability P_{mut} (%). The latter is a translation of the selection pressure γ and mutation rate m as introduced in Eqs. (1) and (4). For infinite computation time both parameters are truly independent. For real world applications, however, long simulation times are costly, and often a quickly obtained suboptimal solution is valued higher than the optimal solution gained in an indefinite time span. For short simulation times we have to take into account that basically all common computing devices[2] work sequentially and can thus execute algorithms only one step at a time.

Since the total number of steps is fixed (by the limited simulation time), it is obvious that, for example, an increased number of mutation steps necessarily allows fewer selection operations and vice versa. We define, therefore, the mutation rate as follows:

$$P_{mut}(\%)=100\,m, \qquad m+\gamma\overset{!}{=}1. \tag{12}$$

As one possible numerical realization, the algorithm used in our simulations follows these three steps (parameter control switched off):

```
(1) Initialize the $N$ seekers with random initial conditions.
(2) Set temperature $T$.
(3) while [run time $<$ final time].
    (i) Draw a uniformly distributed random number $\xi \in [0,1]$.
    (ii) if $[\xi < m] \rightarrow$ mutation step:
        (a) Mutate randomly drawn seeker.
        (b) Decide about mutant's acceptance according to Eq. (3).
        (c) If accepted, replace original with mutant. Discard mutant otherwise.
    (iii) else $\rightarrow$ selection step:
        (a) Pick some $2 \leqslant n \ll N$ seekers at random (e.g., $n=4$).
        (b) Determine the best and the worst from these $n$ seekers.
        (c) Replace the worst seeker by the best one.
```

As can be seen here, mutation and selection operate differently on the seeker ensemble. In contrast to genetic algorithms, both operations cannot be carried out simultaneously, causing the initially independent parameters m and γ [cf. Eq. (4)] to be coupled as stated above.

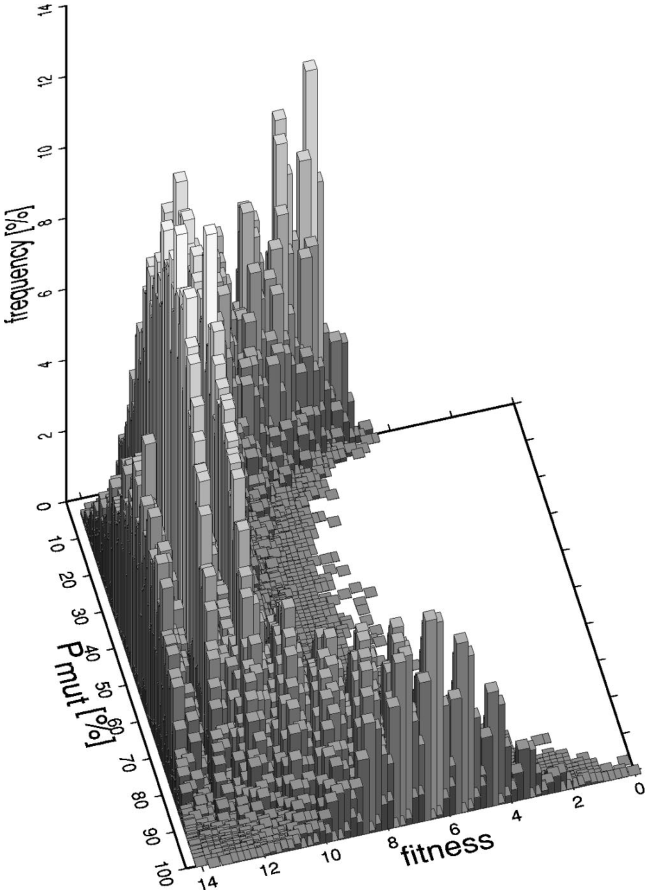

FIG. 1. Frustrated periodic sequence of length $L=15$: Frequency histogram for the different fitness levels depending on the mutation rate m, simulation time $t=500$, temperature $T=1$, tournament selection $r=4$, ensemble size $N=25$, random initial sequences, averaged over 1000 runs.

While the ensemble size is practically limited by time and computer resources, temperature and mutation probability can be chosen at will. As laid out in our previous work [3], both parameters have a strong influence on the ensemble dispersion in search space and, henceforth, on the search efficiency. A good parameter choice ensures that the seekers will cover a certain area of the search space. We will show now that controlling the ensemble's fitness dispersion can drastically improve the search efficiency, since very low mutation rates cause a strategy to resemble gradient search methods (thus leading to a higher likelihood of getting stuck), and too high mutation rates turn the search into a random walk, destroying information already gathered while searching. Figure 1 shows the seeker ensemble distribution to illustrate this problem for frustrated periodic sequences. At low mutation rates ($P_{mut}<20\%$) the ensemble is centered around mediocre fitness values. The ensemble mean improves with an increasing mutation rate—until beyond a certain threshold ($P_{mut}>70\%$) the ensemble completely deteriorates.

An efficient adaptive optimization algorithm needs a numerical measure capable of detecting this error threshold in order to drive the running optimization process as close as possible to the threshold. The ensemble variability intro-

[2] This statement holds with respect to the underlying principle of Turing machines and does not refer to technical implementations such as multiprocessor machines or SIMD instruction sets e.g., on single processor machines. A true exception would be quantum computing devices that are explicitly not considered here.

duced in Ref. [3], defined as the ratio of the number of different seekers to ensemble size *N*:

$$v := \frac{N_{diff}}{N}, \tag{13}$$

though easy to calculate, has turned out to be insufficient for a number of applications. The main problem here is the strong bias towards small ensemble sizes. The relative dispersion measure introduced in Ref. [15] circumvents this problem but requires problem-dependent parameter tuning.

In this paper, we are introducing a nonlinear sensor that avoids bias and the necessity of problem-dependent exhaustive parameter scans in order to determine optimal control values: The entropy *H*, as defined e.g., in information theory can be calculated as

$$H = -\sum_i P_i \ln P_i . \tag{14}$$

By gathering information from the seeker ensemble by generating a histogram as seen in Fig. 1, it is straightforward to derive a normalized entropy measure:

$$H = \sum_i \frac{x_i}{N} \log_N \frac{x_i}{N} . \tag{15}$$

As shown in Fig. 2, the ensemble entropy, varying from *zero* to *one*, marks areas of different fitness values indifferent of the ensemble size. This behavior is observed for all model problems investigated. Now it is easy to pick an ensemble entropy value that accompanies best optimization results (an optimal entropy, that is). The optimal entropy interval reads

$$H_{opt} = 0.17 \pm 0.05 \tag{16}$$

for all tested model problems. This value is even fairly independent of the temperature parameter, as can be seen in Table II. We are not yet sure about the causes for the observed temperature independence.

How sensitive this measure indeed is to mutation rate variations can be seen in Fig. 3. The gradient within the interesting region ($50\% \leqslant P_{mut} \leqslant 100\%$) is very high, so that small deviations from the optimal interval can easily be detected. As stated above, the optimal mutation rate is found close below the error threshold. Comparing Fig. 1 with Fig. 2(a) and 2(b) for example, one can clearly see that indeed for the $N = 25$ seekers used in Fig. 1 the ensemble distribution suddenly spreads for mutation rates above $P_{mut} > 70\%$ as predicted by the optimal ensemble entropy $H_{opt} \approx 0.2$. This coincides with the upper bound of the search parameter interval yielding best results (the evolutionary window) as shown in Fig. 2(a).

V. OPTIMIZATION WITH ADAPTIVE PARAMETER CONTROL

It is intuitively clear that one constant mutation/selection ratio cannot be optimal for all given simulation times. If the computer time is clearly insufficient, the best optimization

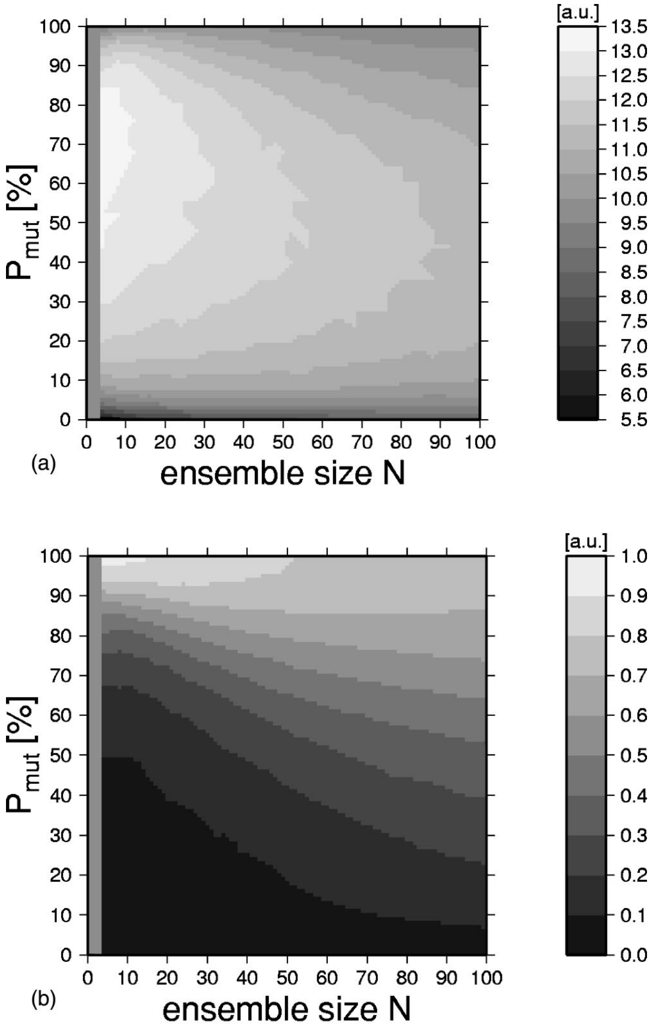

FIG. 2. Mean fitness and ensemble entropy for a frustrated periodic sequence of length $L = 15$; The entropy measure nicely redraws the areas of different fitness values independent of the ensemble size and may thus server as a numerical sensor. The temperature was kept constant at $T = 1$; random initial sequences were used; the simulation time was $t = 500$; the results averaged over 1000 runs. The best fitness values are obtained for an entropy around 0.20.

TABLE II. Optimal entropies for the different model problems in dependence of the temperature. All entropy values have a tolerance of about ±0.05.

	Engel	LABS	RNA
T	Optimal entropy	Optimal entropy	Optimal entropy
0	0.15	0.12	0.12
1	0.20	0.10	0.12
2	0.14	0.13	0.15
3	0.14	0.10	0.12
⋮	⋮	⋮	⋮
10	0.15	0.12	0.16

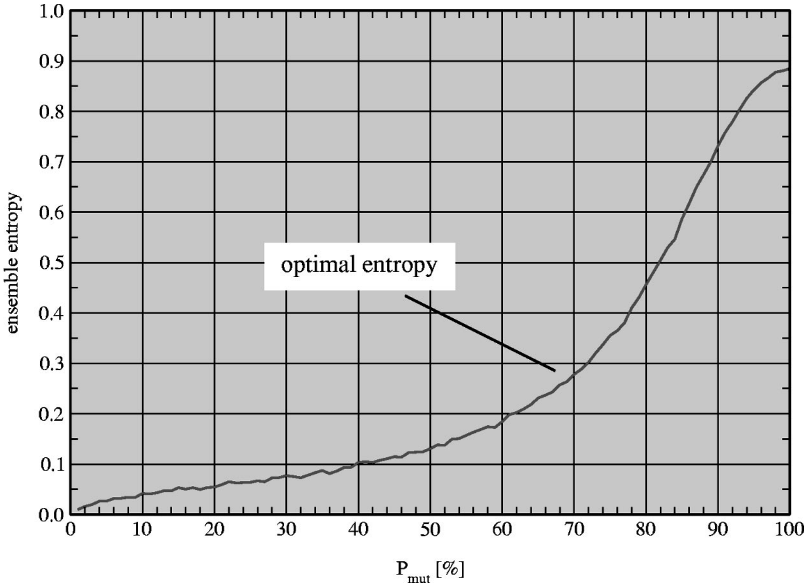

FIG. 3. Frustrated periodic sequence of length $L=15$; ensemble entropy in dependence of the mutation rate *m*; simulation time $t=500$, temperature $T=1$; ensemble size $N=20$; and averaged over 1000 runs.

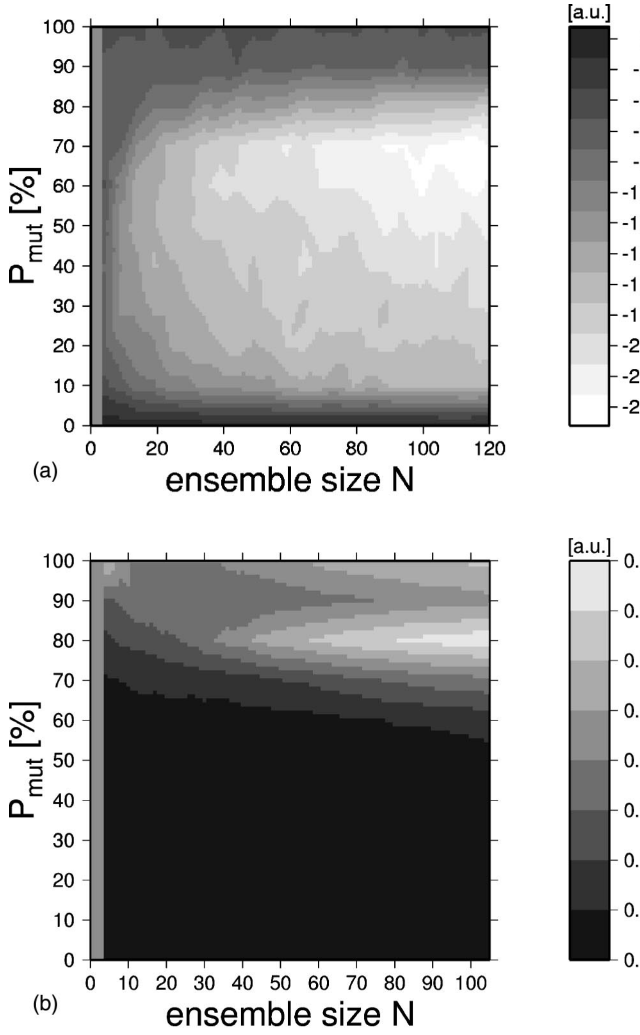

FIG. 4. RNA sequence of length $L=100$; The temperature was kept constant at $T=1$; the simulation time was $t=150$; random initial secondary structures; the results are averaged over 200 runs. The best fitness values are obtained for an entropy around $H_{opt}=0.20–0.40$.

strategy surely is to guess possible solutions. This corresponds to missing selection steps, i.e., $m=1$.

Contrary to what we expected, however, the optimal ensemble entropy is robust, even against variations of the computation time. Taking, for example, the RNA folding problem, too short computation times simply widen the optimal entropy interval. As can be seen in Fig. 4, the best results are achieved for large ensembles with $N>50$ seekers for a computation time of only $t=150$—while for long computation times $t>1000$ ensembles of $N\approx 20$ are much more efficient. Of course, the best optimization result of $F=-32.0$ kcal/mol is not yet reached, but as Fig. 4 demonstrates the best results gained so far are still achieved with ensemble entropies of $0.20\leqslant H_{opt}\leqslant 0.40$, i.e., well below $H=1$. It is also interesting to note that above this interval the functional correspondence between mutation rate P_{mut} and entropy *H* is lost.

Using the information laid out so far, it is possible to derive an advanced adaptive optimization schedule based on our work published in Ref. [15]. Extending the procedure introduced in the previous section, we propose the following algorithm:

```
(1) Initialize the N seekers with random initial conditions.
(2) Start with a high temperature ($\beta \ll 1$; e.g., $T=10^3$).
(3) Initially, disable selection ($m=1$).
(4) while [run time < final time]
    (i) Draw a uniformly distributed random number $\xi \in [0,1]$.
    (ii) if [$\xi < m$]→ mutation step:
        (a) Mutate randomly drawn seeker.
        (b) Decide about mutant's acceptance according to Eq. (3).
        (c) If accepted, replace original with mutant. Discard mutant otherwise.
    (iii) else → selection step:
        (a) Pick some $2\leqslant n \ll N$ seekers at random (e.g., $n=4$).
        (b) Determine the best and the worst from these n
```

seekers.

(c) Replace the worst seeker by the best one.

(iv) After a short computation time, the ensemble statistics yields enough information to increase and control the selection (i.e., reduce the mutation rate $m<1$), keeping the ensemble entropy at H_{opt}.

(v) Follow a standard deviation schedule[3] (SDS) to adapt the temperature parameter:

$$\frac{d\beta(t)}{dt}=\frac{c}{\sigma_f}. \tag{17}$$

The small constant c in Eq. (17) determines the annealing speed; the term σ_f denotes the ensemble fitness' standard deviation.

VI. SUMMARY

In this paper, we investigated mixed evolutionary algorithms. Numerical simulations reveal the existence of a small search parameter interval—the evolutionary window—which the algorithm needs to adapt to in order to be efficient. This paper introduces an entropy-based control sensor, which enables mixed evolutionary algorithms to automatically adapt themselves as necessary.

The numerical results obtained while using the proposed schedule are convincing, as they are absolutely comparable to manually tuned search parameters following an exhaustive parameter scan. In Fig. 5 the comparison is shown for the LABS problem. Within the standard deviation interval, the adaptive optimization gives the best possible results and definitely surpasses the ensemble dispersion-based adaption schedule published in Ref. [15]. The fact that the absolutely best result found in one particular single run (fitness $F=8$) is much better than the expected result ($F=3.6\pm0.5$) indicates that one can achieve better optimization results, granting more simulation time ($t>500$). It still remains to be seen, however, how our results can be generalized with respect to other optimization problems. Also, a profound theoretical motivation for the entropy-based sensor is still missing and must be subject to future investigations. To our disappointment we did not yet succeed to use a simple test function to analytically verify our findings. The well-known, linear onemax problem for example is simple, but not appropriate to study ensemble-based search strategies. A slight modification of the problem to switch from a linear to a parabolic potential, however, has already rendered it unsolvable.

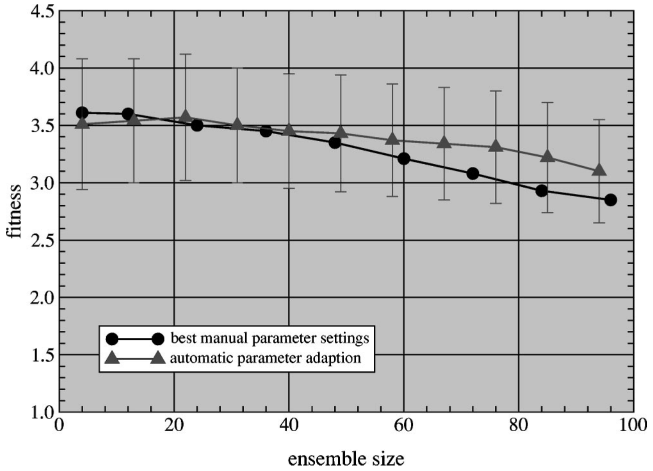

FIG. 5. Expectation value for the ensemble's best seeker; LABS problem of length $L=32$; comparison between exhaustive parameter scan and automatic parameter adaption with random initial strings and initial conditions $m=1$ and $T=10^3$; computation time $t=500$; averaged over 1000 runs; absolutely best result found is the string $S=01010100000111111011011001110011$ with fitness $F=8$.

[3] The SDS was inspired by Andresen [2], used in a previous work [3] and also by Mahnig and Mühlenbein [16].

[1] B. Andresen and J.M. Gordon, Open Sys. Inf. Dyn. Phys. Life Sci. **2(1)**, 1 (1993).

[2] B. Andresen and J.M. Gordon, Phys. Rev. E **6**, 4346 (1994).

[3] Werner Ebeling, Lutz Molgedey, and Axel Reimann, Physica A **287**, 599 (2000).

[4] W. Ebeling, A. Engel, and R. Feistel, *Physik der Evolutionsprozesse* (Akademie-Verlag, Berlin, 1990).

[5] T. Asselmeyer, W. Ebeling, and H. Rosé, Biosystems **39**, 73 (1996).

[6] T. Boseniuk and W. Ebeling, Europhys. Lett. **6**, 107 (1988).

[7] T. Boseniuk, W. Ebeling, and A. Engel, Phys. Lett. A **125**, 307 (1987).

[8] W. Ebeling and A. Engel, Syst. Anal. Model. Simul. **3**, 377 (1986).

[9] W. Ebeling, A. Engel, and V.G. Mazenko, Biosystems **19**, 213 (1986).

[10] M.J.E. Golay and D.B. Harris, IEEE Trans. Inf. Theory **36**, 1163 (1990).

[11] Stephan Mertens, J. Phys. A **29**, L473 (1996).

[12] Burkhard Militzer, Michele Zamparelli, and Dieter Beule, IEEE Trans. Evol. Comp. **2**, 34 (1998).

[13] Walter Fontana, Peter F. Stadler, Pedro Tarazona, Edward D. Weinberger, and Peter Schuster, Phys. Rev. E **47**, 2083 (1993).

[14] Helge Rosé, in *Proceedings of the International Conference on Complex Systems, Nashua, NH, 1997*, edited by Yaneer Bar-Yam (Perseus, Cambridge, MA, 2000).

[15] Werner Ebeling and Axel Reimann, Z. Phys. Chem. (Munich) **216**, 65 (2002).

[16] T. Mahnig and H. Mühlenbein, in *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, Seoul, Korea (IEEE Press, New York, 2001), pp. 183–190.