Threshold-selecting strategy for best possible ground state detection with genetic algorithms

Jörg Lässig[*] and Karl Heinz Hoffmann

Institut für Physik, Technische Universität Chemnitz, D-09107 Chemnitz, Germany

(Received 19 January 2009; published 2 April 2009)

Genetic algorithms are a standard heuristic to find states of low energy in complex state spaces as given by physical systems such as spin glasses but also in combinatorial optimization. The paper considers the problem of selecting individuals in the current population in genetic algorithms for crossover. Many schemes have been considered in literature as possible crossover selection strategies. We show for a large class of quality measures that the best possible probability distribution for selecting individuals in each generation of the algorithm execution is a rectangular distribution over the individuals sorted by their energy values. This means uniform probabilities have to be assigned to a group of the individuals with lowest energy in the population but probabilities equal to zero to individuals which are corresponding to energy values higher than a fixed cutoff, which is equal to a certain rank in the vector sorted by the energy of the states in the current population. The considered strategy is dubbed *threshold selecting*. The proof applies basic arguments of Markov chains and linear optimization and makes only a few assumptions on the underlying principles and hence applies to a large class of algorithms.

DOI: 10.1103/PhysRevE.79.046702 PACS number(s): 02.60.Pn, 02.50.Ga, 05.10.−a

I. INTRODUCTION

The problem of finding the ground state of a complex system arises in many areas of modern sciences and related problems of global optimization have an ample number of applications in engineering and economics as for circuit design [1], process planning [2], and scheduling [3]. In the realm of physics, such problems occur in the study of systems such as spin glasses [4], neural networks [5], and protein structure prediction and simulation [6,7]. From the mathematical point of view, the global minimum in the domain of a real-valued function is requested.

Because of the NP completeness [8] of most problems of this class, there is no hope for deterministic methods which solve these problems in reasonable running times and hence, various heuristic methods for a near optimal solution have been developed, such as *simulated annealing* [9], *threshold accepting* [10], *particle swarm optimization* [11], or *genetic algorithms* (GAs) [12], which are in the focus of this paper. A general drawback of most of these methods is that they have many free parameters to be adjusted by the user and, even worse, that these methods often require different parameter setups for different problem domains and instances. Hence, much research has been done to choose different parameters possibly optimal (see, e.g., [13–17]).

In the particular case of designing a GA for a certain given problem, there are many degrees of freedom to be fixed, but as for the other methods mentioned above, the choice of certain parameters or operators often relies strongly on experimental studies and the experience of the programmer. Such choices are, e.g., (i) *representation of a solution* in the state space as an artificial genome, (ii) choice of a *crossover operator* to form a new population in each iteration, (iii) choice of a *mutation rate*, and (iv) choice of a *selection scheme* over the individuals of a population for crossover.

Today GAs are in broad successful application to problems in many different fields [18–23] and excellent experimental results have been obtained. Despite interesting theoretical progress in the last years [24–29], exact proves for optimal choices of design criteria are still missing. This paper focuses on the last of the design criteria above also called *parent selection*. In all variants of GAs, some form of the selection operator must be present [24]. A wide variety of selection strategies has been proposed in the literature. In general, m individuals of the current population of size n have to be selected for crossover into a *mating pool*. Individuals with higher fitness are more likely to receive more than one copy and less fit individuals are more likely to receive no copies. In different replacement schemes, the size of the pool differs. After selecting the mating pool, some crossover scheme takes individuals from that pool and produces new outcomes, until the pool is exhausted. No further structural restrictions are necessary for our considerations concerning the optimal choice of a selection strategy.

The behavior of the GA very much depends on how individuals are chosen to go into the mating pool [12]. Examples are given in Table I. The simplest approach is that the reproduction probability of an individual of the population is proportional directly to its fitness (*roulette-wheel selection*).

TABLE I. Comparison of different selection strategies.

Rank of the individuals	1	2	3	4
Raw fitness	50.0	25.0	15.0	10.0
Roulette-wheel	0.5	0.25	0.15	0.1
Windowing	$0.\bar{6}$	0.25	$0.08\bar{3}$	0.0
Exponential $(\sqrt{f+1})$	0.365	0.261	0.205	0.169
Linear transformation $(2f+1)$	0.495	0.25	0.152	0.103
Linear ranking selection	0.4	0.3	0.2	0.1
Binary tournament selection	0.438	0.312	0.188	0.062
Threshold Selecting $(\nu=3)$	$0.\bar{3}$	$0.\bar{3}$	$0.\bar{3}$	0.0

[*] joerg.laessig@physik.tu-chemnitz.de

1539-3755/2009/79(4)/046702(8)

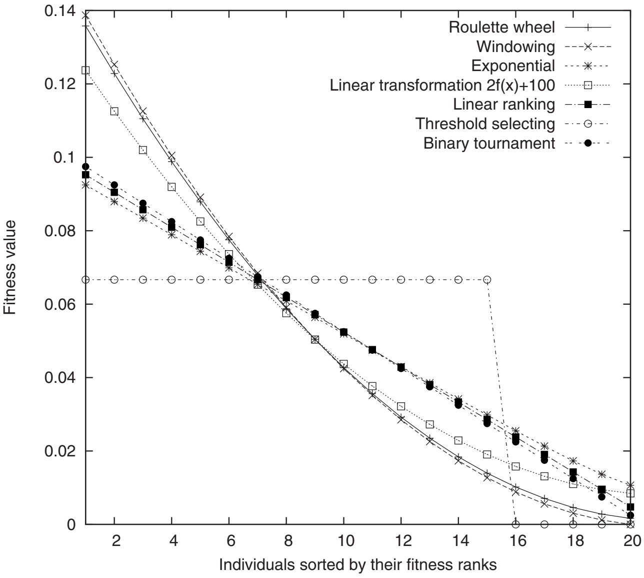

FIG. 1. Functions of different selection strategies.

Other approaches scale the original fitness function first for better performance. Examples are *windowing*, where the fitness of the worst individual is subtracted from each individual fitness before calculating the selection probabilities $f' = f - f_w$; *linear transformation*, where a linear function of the fitness is computed, i.e., $f' = c_0 f + c_1$; or *exponential*, where a scaled fitness function $f' = (c_0 f + c_1)^{c_2}$ is computed (c_0, c_1 and c_2 are constants). Then the probabilities are chosen proportional to the values f', taking normalization into account. In *linear ranking selection*, a linear function over a fitness ranking of the individuals is applied and in *binary tournament selection* two individuals are selected with uniform probability in a preselection step and the individual with better fitness is then submitted to the mating pool. *Sigma truncation*, *dynamic scaling*, or *hyperbolic scaling* is also common. The strategy *threshold selecting* applies rectangular distributions with a certain cutoff rank ν, as introduced in [30]. Note that the rank ν can be different for different individuals which are chosen in the same iteration and of course also in different iterations. See [12,24,26] for an overview on different selection schemes.

Table I [12] introduces the different methods for a population of four individuals with exemplary fitness values 50, 25, 15, and 10. In Fig. 1 the different selection strategies are visualized for a larger population of 20 individuals. For a better readability, the exemplary fitness values of the individuals have been chosen according to a function, i.e., $f(x) = 2x^2 + x + 7$ for the xth individual.

Each crossover strategy is reported to have strengths and weaknesses. In general, the selection strategy has to be chosen such that the population evolves toward "better" overall fitness. For example, the fitness of the fittest individual in the final population might be required to be as high as possible. In Secs. V and VI, it is proven that *threshold selecting* is optimal in a certain sense defined below. This generalizes the results of [30] to a larger class of objective functions. In *threshold selecting*, the selection is based on fitness ranks, and the selection probability on the ranks is rectangular, i.e., it includes one or more individual(s) with the highest fitness value(s) with the same nonvanishing probability but introduces a cutoff rank ν, so that all individuals with higher ranks are selected with probability zero.

II. IDEA IN A NUTSHELL

The proof to show the optimality of *threshold selecting* for a large class of objective functions, which is the main contribution of this paper, is based on the fact that the selection probability distributions assigning probabilities to n-ordered objects can be seen as vectors in a n dimensional space. Special assumptions of the problem structure restrict the space of possible solutions to a simplex, which is described in detail in Sec. IV. Then, due do the linearity of applicable objective functions on the selection probabilities as described in Secs. V and VI, the problem reduces to the task to find the minimum of a linear function on a simplex, which must be a vertex in the general case by the fundamental theorem of linear programming. As will be shown, the vertices are exactly equivalent to the rectangular distributions mentioned above as *threshold selecting*.

This proof technique has been already applied to show for the acceptance rule in Monte Carlo methods, such as *simulated annealing* [9], *threshold accepting* [10], or Tsallis *statistics* [13,14,31], that threshold accepting is provably a best possible choice [15].

Furthermore, the stochastic optimization algorithm *extremal optimization* [32] has been investigated [16,17]. Extremal optimization also works by simulating random walkers as do the methods mentioned before, but it needs a special structure of the problem under consideration. Every state is specified by several degrees of freedom, each of which can be assigned a fitness. Each iteration chooses 1 degree of freedom to be changed based on fitness values. It has been shown that a rectangular distribution is the best choice in each iteration of extremal optimization.

III. DEFINITIONS

We consider problems with finite state spaces Ω of states $\alpha \in \Omega$. An *energy function* $E(\alpha)$ describes how desirable every single state α is and has to be minimized, i.e., the states of lower energy are better. Due to the finite state space, there is only a finite number of possible values for $E(\alpha)$. As stated, in genetic algorithms, it is very common to define a fitness function $f(\alpha)$ where states with lower energy have a higher fitness because energy has to be minimized. Hence, different mappings to fulfill this requirement would be possible but functions as defined by

$$f(\alpha) := \max\{E(\beta)|\beta \in \Omega\} - E(\alpha)$$

would be a considerable choice here. The energy notation is more common in physics; hence it is used predominantly in Secs. III–VII. GAs consider populations (or pools) of states. If there are n states in a population then each generation is equivalent to a generalized state vector $\boldsymbol{\alpha} := (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \Omega^n = \boldsymbol{\Omega}$ with n finite. A generalized energy function $E(\boldsymbol{\alpha})$

has to be defined for our considerations as well, which is done by

$$E(\boldsymbol{\alpha}) := \min\{E(\alpha_i)|i=1,2,\ldots,n\}. \tag{1}$$

To obtain low energies, GAs proceed by randomly selecting a start population and then evolving it by a selection and subsequent crossover operation. Mutations are also possible, but of no importance for our considerations. The reason is that the consecutive application of a crossover operator $\mathcal{C}$ with crossover probability p_c and a mutation operator $\mathcal{M}$ with mutation probability p_m can be considered as one application of a universal operator, i.e.,

$$\boldsymbol{\alpha} = \mathcal{M}[\mathcal{C}(\boldsymbol{\beta},p_c),p_m] = (\mathcal{M}\circ\mathcal{C})(\boldsymbol{\beta},p_m,p_c) =: \mathcal{U}(\boldsymbol{\beta}).$$

The application of $\mathcal{U}$ instead of a pure crossover operator $\mathcal{C}$ does not change the reasoning applied later on. In both cases, there is a rigorously defined probability to obtain a fixed state $\boldsymbol{\beta}$ from a given state $\boldsymbol{\alpha}$ if the operator is applied (only the probability values change if mutation is integrated, which does not matter for the general setup). So be aware that despite our focus on crossover, the following considerations include also operator applications with positive probabilities for mutations. Another assumption is that we here confine ourselves to selection steps, where the probability to enter the mating pool for the crossover operation is based on the ranks of the population members in a ranking where they are ordered by their energy value. Later on it gets clear that the proof covers also the distributions from Table I, which are not only rank based, because they all satisfy the conditions (A1) and (A2) regardless of other parameter choices.

The possible mating pools are described by vectors of chosen ranks $\boldsymbol{r}\in(\mathbb{N}_n^+)^m$, where $\mathbb{N}_n^+$ denotes the set $\{1,2,\ldots,n\}$ of integers and m denotes the number of chosen ranks for the crossover step.

For the choice of the m individuals for the crossover step in the GA, m time-dependent probability distributions $d^{i,t}$, $i=1,2,\ldots,m$ are defined over the ranks r, where $d^{i,t}(r)$ denotes the probability that the individual with rank r is chosen in time step t as ith element of the mating pool. Given this structure at time t, exactly m ranks $\boldsymbol{r}=(r_1,r_2,\ldots,r_m)$ are chosen by the GA and hence, m individuals from the current population according to $d^{i,t}$, $i=1,2,\ldots,m$. Technically, each of the individual members β_i of the current population $\boldsymbol{\beta}$ is assigned a rank $r(\beta_i)$ based on its energy, i.e., the individuals in a population can be ordered according to their energy values in a ranking with

$$r(\beta_i)\le r(\beta_j) \Leftrightarrow E(\beta_i)\le E(\beta_j) \quad \forall \quad \text{pairs}(i,j),$$

where $r(\beta_i)$, $r(\beta_j)\in\mathbb{N}_n^+=\{1,2,\ldots,n\}$. The following assumptions are adopted for the selection probabilities $d^{i,t}(r)$. (A1) Each step of the algorithm is independent of the former steps. (A2) In each step t, $1\ge d^{i,t}(1)\ge d^{i,t}(2)\ge\cdots\ge d^{i,t}(n)\ge 0$ holds for $i=1,2,\ldots,m$, i.e., it is more probable to select individuals with lower rank (lower energy) than individuals with a higher rank (higher energy). (A3) $\sum_{r=1}^{n} d^{i,t}(r)=1$ for $i=1,2,\ldots m$, i.e., the distributions are normalized.

Due to the random nature of the selection process, there is a time-dependent probability to obtain the mating pool based on a vector of chosen ranks $\boldsymbol{r}=(r_1,r_2,\ldots,r_m)$,

$$\Lambda_{\boldsymbol{r}}^{S,t} = d^{1,t}(r_1)d^{2,t}(r_2)\ldots d^{m,t}(r_m). \tag{2}$$

In the crossover step, an operator $\mathcal{C}_{\boldsymbol{r}}$ is applied to the current population $\boldsymbol{\beta}$. The operator $\mathcal{C}_{\boldsymbol{r}}$ might not be deterministic but determines the fixed probabilities $\Lambda^C_{\boldsymbol{\alpha r\beta}}$ to obtain a new population $\boldsymbol{\alpha}\in\boldsymbol{\Omega}$ from $\boldsymbol{\beta}\in\boldsymbol{\Omega}$ with the chosen ranks $\boldsymbol{r}$ as intermediate step. For each fixed pair $\boldsymbol{r}$ and $\boldsymbol{\beta}$, we have

$$\sum_{\boldsymbol{\alpha}\in\boldsymbol{\Omega}} \Lambda^C_{\boldsymbol{\alpha r\beta}} = 1.$$

An exemplary crossover procedure could work as follows. After obtaining a mating pool of m states which are corresponding to the chosen ranks $\boldsymbol{r}$ of the old generalized state $\boldsymbol{\beta}$, m new states are created by recombination of these chosen states. Including the current n states then there are $n+m$ states available and n states are kept for the new generation $\boldsymbol{\alpha}$ applying some standard procedure (e.g., keep the best n of all $n+m$ states). In the special case of generation replacement, we have $n=m$ and $\boldsymbol{\beta}$ is replaced completely by the states from the recombination step.

One can think of the recombination itself that states from the mating pool are taken one after another. Each possible tuple of states for one crossover operation is chosen with the same probability, i.e., the probability is uniformly distributed among all possible tuples of desired size in the mating pool. Most commonly pairs are chosen and for each pair a split position for one point crossover or more than one split position for multipoint crossover procedures are determined—again uniformly distributed (the proof below is general enough that also other distributions or procedures are possible here).

Combining the selection step with probabilities $\Lambda_{\boldsymbol{r}}^{S,t}$ and the crossover step with probabilities $\Lambda^C_{\boldsymbol{\alpha r\beta}}$ leads to a transition probability $\Gamma^t_{\boldsymbol{\alpha\beta}}$ from one population $\boldsymbol{\beta}$ to the next population $\boldsymbol{\alpha}$. In summary, the dynamics of GAs can be described as a Markovian random walk in-state space. For the development of the probability $p^t_{\boldsymbol{\alpha}}$ to be in state $\boldsymbol{\alpha}$ (which means to have a certain population in the GA), the master equation

$$p^t_{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\beta}\in\boldsymbol{\Omega}} \Gamma^t_{\boldsymbol{\alpha\beta}} p^{t-1}_{\boldsymbol{\beta}} \tag{3}$$

is applicable. Here $\Gamma^t_{\boldsymbol{\alpha\beta}}$ is defined to be

$$\Gamma^t_{\boldsymbol{\alpha\beta}} = \sum_{\boldsymbol{r}\in(\mathbb{N}_n^+)^m} \Lambda^C_{\boldsymbol{\alpha r\beta}}\Lambda^{S,t}_{\boldsymbol{r}} \overset{(2)}{=} \sum_{\boldsymbol{r}\in(\mathbb{N}_n^+)^m} \Lambda^C_{\boldsymbol{\alpha r\beta}}\prod_{i=1}^{m} d^{i,t}(r_i). \tag{4}$$

In the next step, the dependence of the performance of the GA on the probability distributions $d^{1,t}$, $d^{2,t}$, $\ldots$, $d^{m,t}$ over the ranks in the population is investigated and we determine which choice is an optimal one for these distributions, considering an optimization run with S steps.

Most commonly, one of the following objectives is used [15] (here slightly adapted in the notation for GAs). (O1) The mean energy of the best individual in the final population (mean final energy) should be as low as possible. (O2) The probability of having a final population containing an individual with ground state energy should be as large as possible. (O3) The expected number of obtained populations during the execution of the algorithm which contain at least one ground state individual should be as large a possible. (O4) The probability of obtaining a population which contains a ground state during the execution of the algorithm should be as large as possible. (O5) The mean energy of the best so far obtained individual during the algorithm execution (mean best so far energy) should be as low as possible.

In practice the objectives (O4) and (O5) are the most common ones. To optimize according to (O1), one chooses

$$g_1(\boldsymbol{\alpha}) = E(\boldsymbol{\alpha}) \overset{(1)}{=} \min\{E(\alpha_i)|i=1,2,\ldots,n\},$$

which means essentially that the quality of a population is assumed to be equivalent to the quality of the best individual in the population. Here $\boldsymbol{\alpha}$ is assumed to be the final population. To optimize according to (O2), one chooses

$$g_2(\boldsymbol{\alpha}) = \begin{cases} 0 & | \quad \text{if } \boldsymbol{\alpha} \text{ contains a ground state} \\ 1 & | \quad \text{otherwise,} \end{cases}$$

i.e., only optimal states with ground state energy have objective values equivalent to zero. The other objectives are described in Sec. VI. The objectives (O1) and (O2) are linear in the final-state probabilities, which is the important fact for the proof as explained below.

The optimization process consists of a finite number of S steps ($t:=1,2,\ldots,S$). Note that $\Gamma^t_{\boldsymbol{\alpha\beta}}$ is linear in $d^{i,t}(r)$ for i fixed by definition in Eq. (4). The arguments below apply in general to any objective function which is linear in the final-state probabilities $p^S_{\boldsymbol{\alpha}}$ as, e.g., (O1) and (O2). The state probabilities at time t are considered as vector $\boldsymbol{p}^t$ with $\dim(\boldsymbol{p}^t) = |\boldsymbol{\Omega}| =: L$ and the linear objective function with values $g(\boldsymbol{\alpha})$ for each state $\boldsymbol{\alpha} \in \boldsymbol{\Omega}$ as vector $\boldsymbol{g}$, again with $\dim(\boldsymbol{g}) = L$. If $(\cdot)^{\text{tr}}$ denotes the transpose, the measure of performance is equivalent to

$$g(\boldsymbol{p}^S) = \boldsymbol{g}^{\text{tr}} \cdot \boldsymbol{p}^S = \sum_{\boldsymbol{\alpha} \in \boldsymbol{\Omega}} g(\boldsymbol{\alpha}) p^S_{\boldsymbol{\alpha}} \to \min. \tag{5}$$

IV. SETUP OF A VECTOR SPACE

In the following, the distributions $d^{i,t}(r)$, $r=1,2,\ldots,n$, are considered to be n dimensional vectors $\boldsymbol{d}^{i,t}$ with entries $d^{i,t}_r := d^{i,t}(r) \in [0,1]$. Assume without loss of generality $m-1$ of these distributions $\boldsymbol{d}^{i,t}$, $i \in 1,2,\ldots,m$ at time step t, to be fixed. Only one remaining distribution denoted by $\boldsymbol{d}^{c_t,t}$ is open to optimize. The question is then how to choose $\boldsymbol{d}^{c_t,t}$ to minimize Eq. (5). As a consequence of the assumptions (A2) and (A3), the region $\mathcal{F}$ of feasible vectors $\boldsymbol{d}^{c_t,t}$ is defined by the $n+1$ linear inequalities in (A2) and one linear equation in (A3), where the first inequality $1 \geq d^{c_t,t}_1$ follows from the others. Of the remaining n inequalities, $n-1$ must be set to equations to find extreme points (vertices) in the region $\mathcal{F}$. Letting V denote the set of extreme points of $\mathcal{F}$, the elements of V are exactly those vectors $\boldsymbol{d}^{c_t,t}$ which have the initial sequence of i entries equal to $1/i$ followed by a sequence of $n-i$ entries equal to zeros. Explicitly, $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$, where $\boldsymbol{v}_1 = (1,0,0,\ldots,0)^{\text{tr}}$, $\boldsymbol{v}_2 = (1/2,1/2,0,0,\ldots,0)^{\text{tr}}$, $\boldsymbol{v}_i = (1/i,1/i,\ldots,1/i,0,0,\ldots,0)^{\text{tr}}$, and $\boldsymbol{v}_n = (1/n,1/n,\ldots,1/n)^{\text{tr}}$. Note that the elements of V are linearly independent. Then by an argument from [15] $\mathcal{F}$ is exactly the convex hull $C(V)$ of V, which is a simplex (see Fig. 2). The equivalence of $C(V)$ and the region $\mathcal{F}$ defined by (A1) to (A3) can be seen as follows. For the lth row of an arbitrary element,

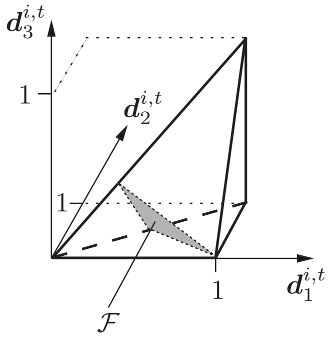

FIG. 2. Simplex in three dimensions.

$$\boldsymbol{d} = \sum_{i=1}^{n} a_i \boldsymbol{v}_i, \quad \forall\, i=1,2,\ldots,n{:}a_i \in [0,1], \quad \sum_{i=1}^{n} a_i = 1$$

of $C(V)$ the relation

$$d_l = \sum_{i=l}^{n} a_i \frac{1}{i} = \sum_{i=l+1}^{n} a_i \frac{1}{i} + a_l \frac{1}{l} = d_{l+1} + a_l \frac{1}{l} \geq d_{l+1},$$

holds, which means the assumption (A2) is fulfilled. Now summing up the rows of $C(V)$ gives

$$\sum_{l=1}^{n} d_l = \sum_{l=1}^{n} \sum_{i=l}^{n} a_i \frac{1}{i} = \sum_{l=1}^{n} l a_l \frac{1}{l} = \sum_{l=1}^{n} a_l = 1,$$

showing also that (A3) is fulfilled. Thus, $C(V) \subseteq \mathcal{F}$. Conversely, consider an arbitrary point $\boldsymbol{d} \in \mathcal{F}$. Since the vertices $\boldsymbol{v}_i$ are linearly independent, they can be used as a basis and $\boldsymbol{d}$ can be written as a linear combination

$$\boldsymbol{d} = \sum_{i=1}^{n} b_i \boldsymbol{v}_i.$$

Then, for the lth component d_l, this gives

$$d_l = \sum_{i=l}^{n} b_i \frac{1}{i} = d_{l+1} + b_i \frac{1}{l},$$

which by (A2) implies

$$d_l \geq d_{l+1} \Rightarrow d_l - d_{l+1} = b_l \frac{1}{l} \geq 0 \Rightarrow b_l \geq 0.$$

Summing up all values d_l and using (A3) gives

$$\sum_{l=1}^{n} d_l = \sum_{l=1}^{n} l b_l \frac{1}{l} = \sum_{l=1}^{n} b_l = 1 \Rightarrow b_l \leq 1.$$

Hence, the constraints $b_l \geq 0$ and $b_l \leq 1$ hold and therefore $\boldsymbol{d} \in C(V)$ for all $\boldsymbol{d} \in \mathcal{F}$, i.e., $\mathcal{F} \subseteq C(V)$, $C(V) = \mathcal{F}$.

V. PROVING OPTIMALITY

In the following, it is proven that threshold selecting is the best possible choice for each distribution $d^{i,t}$, $i=1,2,\ldots,m$ in the time steps $t=1,2,\ldots,S$. First the final step S of the optimization process is considered. The output of the last step is $\boldsymbol{p}^S$ and used to determine the value of the optimality criterion as in Eq. (5). In step S, one has to solve the optimization problem in Eq. (5) for the given input $\boldsymbol{p}^{S-1}$. Using Eq. (3) one gets

$$g(\boldsymbol{p}^S)=\sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\boldsymbol{\Omega}} g(\boldsymbol{\alpha})\Gamma^S_{\boldsymbol{\alpha\beta}}\,p^{S-1}_{\boldsymbol{\beta}} \rightarrow \min, \tag{6}$$

with $\Gamma^S_{\boldsymbol{\alpha\beta}}$ given by Eq. (4). Note that in this stage, $p^{S-1}_{\boldsymbol{\beta}}$ is fixed in any case to a value which has been determined by the first $S-1$ steps of the algorithm. Now the distributions in $\Gamma^S_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ defined by

$$\begin{aligned} g(\boldsymbol{p}^S) &= \sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\boldsymbol{\Omega}} g(\boldsymbol{\alpha})\Gamma^S_{\boldsymbol{\alpha\beta}}\,p^{S-1}_{\boldsymbol{\beta}} \\ &\overset{(4)}{=} \sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\boldsymbol{\Omega}} g(\boldsymbol{\alpha})\left[\sum_{\boldsymbol{r}\in(\mathbb{N}_n^+)^m} \Lambda^C_{\boldsymbol{\alpha r\beta}}\prod_{i=1}^m d^{i,S}(r_i)\right]p^{S-1}_{\boldsymbol{\beta}} \\ &= \sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\boldsymbol{\Omega}} g(\boldsymbol{\alpha})\boldsymbol{d}^{c_S,S}\cdot\boldsymbol{h}(\mathcal{D}_S^-)p^{S-1}_{\boldsymbol{\beta}} \\ &= \boldsymbol{d}^{c_S,S}\cdot\sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\boldsymbol{\Omega}} g(\boldsymbol{\alpha})\boldsymbol{h}(\mathcal{D}_S^-)p^{S-1}_{\boldsymbol{\beta}} \rightarrow \min \end{aligned} \tag{7}$$

have to be optimized. Here $\mathcal{D}_S^-$ denotes the set of distributions $\boldsymbol{d}^{1,S}$, $\boldsymbol{d}^{2,S},\ldots,\boldsymbol{d}^{c_S-1,S}$, $\boldsymbol{d}^{c_S+1,S},\ldots,\boldsymbol{d}^{m,S}$ and $\boldsymbol{h}(\mathcal{D}_S^-)$ is the vector which is obtained if $\boldsymbol{d}^{c_S,S}$, $c_S\in\{1,2,\ldots,m\}$ gets factored out. Obviously this is possible for each distribution $\boldsymbol{d}^{i,S}$ and hence the constant c_S can be chosen arbitrarily from $\{1,2,\ldots,m\}$. This means $g(\boldsymbol{p}^S)$ depends only linearly on each single distribution $\boldsymbol{d}^{i,S}$, $i=1,2,\ldots,m$, which selects the ith individual for crossover. Consequently by the fundamental theorem of linear optimization, the distributions can be chosen optimally as vertices $\boldsymbol{v}\in V$. But, because the distributions of the previous steps are not fixed in the current stage of the proof, the optimal values p^{S-1}_{β} are currently unknown. Hence, all possible combinations of vertices from V are considered for the distributions $\boldsymbol{d}^{i,S}$, $i=1,2,\ldots,m$. These are n^m possible choices. This finishes step S. Defining $g^{S-1}(\boldsymbol{\alpha})=\Sigma_{\boldsymbol{\gamma}\in\boldsymbol{\Omega}}g(\boldsymbol{\gamma})\Gamma^S_{\boldsymbol{\gamma\alpha}}$ as a new objective function (this has to be done in each of the n^m search branches independently) and considering now the step before, i.e., step $S-1$, one obtains

$$\boldsymbol{g}^{\mathrm{tr}}\cdot\boldsymbol{p}^S=\sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\boldsymbol{\Omega}} g^{S-1}(\boldsymbol{\alpha})\Gamma^{S-1}_{\boldsymbol{\alpha\beta}}p^{S-2}_{\boldsymbol{\beta}} \rightarrow \min. \tag{8}$$

Obviously the same transformation as above can be applied to factor out $\boldsymbol{d}^{c_{S-1},S-1}$, where $c_{S-1}\in\{1,2,\ldots,m\}$ and by the same arguments as above the optimal transition probabilities are found by taking $\boldsymbol{d}^{c_{S-1},S-1}$ to be an element of V. Again this reasoning is valid for each distribution $\boldsymbol{d}^{i,S-1}$, $i=1,2,\ldots,m$. For all other steps $S-2$, $S-3,\ldots,1$ the same argument holds as well, i.e., $\boldsymbol{d}^{i,t}$, $i=1,2,\ldots,m$ are all elements of the vertex set V. In the last step, the search tree has $(n^m)^S$ branches and one of these branches is equivalent to the optimal choice of vertices of the complete iterative search process of the GA. Equivalently, this process can be also imagined by applying the Bellman principle of dynamic programming [33], working the way backward starting with the last step.

Hence, the proof shows that a rectangular distribution over the individuals with the lowest-energy values in each generation in the iterations $t=1,2,\ldots,S$ in GAs gives the best implementation of the selection step for each individual used for the crossover step in iteration t.

VI. GENERALIZATION

The arguments above are now generalized to cover besides (O1) and (O2) also the objectives (O3) to (O5). To obtain the probability $P^t_{\cup,E}$ of a generalized state of having obtained the energy value E or lower during the execution of t steps of the genetic algorithm, one first has to introduce generalized transition probabilities

$$\hat{\Gamma}^t_{\boldsymbol{\alpha\beta},E}=\begin{cases}\delta(\boldsymbol{\alpha},\boldsymbol{\beta}) & \mid \quad \text{if } E(\boldsymbol{\beta})\le E\\ \Gamma^t_{\boldsymbol{\alpha\beta}} & \mid \quad \text{otherwise,}\end{cases} \tag{9}$$

where $\delta(\boldsymbol{\alpha},\boldsymbol{\beta})$ denotes the Kronecker delta. By construction, the probability to leave a generalized state with energy value at most E is zero. Obviously, all values $\hat{\Gamma}^t_{\boldsymbol{\alpha\beta},E}$ are still linear in the selection probabilities of individuals into the mating pool $\boldsymbol{d}^{1,t}$, $\boldsymbol{d}^{2,t},\ldots,\boldsymbol{d}^{m,t}$.

It is now possible to specify a *master equation* for the development of a probability distribution $p^t_{\boldsymbol{\alpha},E}$ to be in the generalized state $\boldsymbol{\alpha}$ after t time steps,

$$p^t_{\boldsymbol{\alpha},E}=\sum_{\boldsymbol{\beta}\in\boldsymbol{\Omega}}\hat{\Gamma}^t_{\boldsymbol{\alpha\beta},E}\,p^{t-1}_{\boldsymbol{\beta},E}. \tag{10}$$

If $\boldsymbol{\alpha}$ has an energy value $E(\boldsymbol{\alpha})\le E$ then the random walk is trapped in this state but if $E(\boldsymbol{\alpha})>E$, no state with energy value E or lower has been obtained so far. The accumulated probability to be in some generalized state with energy value at most E is given by

$$P^t_{\cup,E}=\sum_{\boldsymbol{\alpha}\in\boldsymbol{\Omega},E(\boldsymbol{\alpha})\le E}p^t_{\boldsymbol{\alpha},E}. \tag{11}$$

Because $\boldsymbol{\Omega}$ is a finite state space, i.e., $|\boldsymbol{\Omega}|<\infty$, there is only a finite set of possible energy values $E(\boldsymbol{\alpha})$. More specific, the number of different possible energy values is equivalent to the different values $E(\alpha)$ with $\alpha\in\Omega$ and the possible values are here denoted by $E_k\in\{E_1,E_2,\ldots,E_K\}$ with $E_1<E_2<\cdots<E_K$. Hence, for each possible value $k=1,2,\ldots,K$, the value $P^t_{\cup,E_k}$ is clearly defined. The probability that the lowest so far obtained energy value is exactly E_k is given by

$$P^t_{\cap,E_k}=P^t_{\cup,E_k}-P^t_{\cup,E_{k-1}}, \tag{12}$$

where we introduced an energy value $E_0<E_1$ with $P^t_{\cup,E_0}=0$ for the convenience of the notation. To address the objectives (O3) to (O5), it is not sufficient to multiply the final-state probabilities with an objective vector $\boldsymbol{g}$ of $\dim(\boldsymbol{g})=|\boldsymbol{\Omega}|=L$. Instead a tuple

$$\mathcal{G}=(\boldsymbol{G}^S,\boldsymbol{G}^{S-1},\ldots,\boldsymbol{G}^1)$$

of objective vectors

$$\boldsymbol{G}^t = [(\boldsymbol{g}^t_{E_0})^{\mathrm{tr}}, (\boldsymbol{g}^t_{E_1})^{\mathrm{tr}}, \ldots, (\boldsymbol{g}^t_{E_K})^{\mathrm{tr}}]^{\mathrm{tr}}$$

with $\dim(\boldsymbol{G}^t) = L(K+1)$ has to be defined to express a certain objective, where $\boldsymbol{g}^t_{E_k}$ is the objective vector to be multiplied with the probability vector $\boldsymbol{p}^t_{E_k}$ consisting of all probabilities $p^t_{\boldsymbol{\alpha},E_k}$, $\boldsymbol{\alpha} \in \boldsymbol{\Omega}$. First, Eq. (10) can be expressed in matrix notation equivalently, integrating all modified transition probabilities $\hat{\Gamma}^t_{\boldsymbol{\alpha\beta},E}$ in matrices $\hat{\boldsymbol{\Gamma}}^t_E$, controlling the dynamics by

$$\boldsymbol{p}^t_E = \hat{\boldsymbol{\Gamma}}^t_E \cdot \boldsymbol{p}^{t-1}_E$$

and combining all possible different energy values $E_0, E_1, E_2, \ldots, E_K$ in one equation by

$$\boldsymbol{q}^t = \begin{pmatrix} \boldsymbol{p}^t_{E_0} \\ \boldsymbol{p}^t_{E_1} \\ \vdots \\ \boldsymbol{p}^t_{E_K} \end{pmatrix} = \begin{pmatrix} \hat{\boldsymbol{\Gamma}}^{t-1}_{E_0} & 0 & \cdots & 0 \\ 0 & \hat{\boldsymbol{\Gamma}}^{t-1}_{E_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\boldsymbol{\Gamma}}^{t-1}_{E_K} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{p}^{t-1}_{E_0} \\ \boldsymbol{p}^{t-1}_{E_1} \\ \vdots \\ \boldsymbol{p}^{t-1}_{E_K} \end{pmatrix}$$
$$= \hat{\boldsymbol{\Gamma}}^t \cdot \boldsymbol{q}^{t-1}, \qquad (13)$$

where $\dim(\boldsymbol{q}^t) = L(K+1)$. Assuming that the generalized states $\boldsymbol{\alpha}$ are ordered according to some sorting in $\boldsymbol{\Omega}$ and the position of a certain state $\boldsymbol{\alpha} \in \boldsymbol{\Omega}$ in this sorting is given by $s = s(\boldsymbol{\alpha})$, the following equivalence is given:

$$q^t_{|\boldsymbol{\Omega}|k+s(\boldsymbol{\alpha})} = q^t_{Lk+s(\boldsymbol{\alpha})} = p^t_{\boldsymbol{\alpha},E_k}. \qquad (14)$$

Note that the unmodified chain with transition probabilities $\boldsymbol{\Gamma}^t$ is contained in this generalized chain at the positions $q^t_1, q^t_2, \ldots, q^t_L$. Now it is possible to optimize the tuple of vectors $\mathcal{Q} = (\boldsymbol{q}^S, \boldsymbol{q}^{S-1}, \ldots, \boldsymbol{q}^1)$ according to arbitrary objective functions $\mathcal{G}(\mathcal{Q})$:

$$\mathcal{G}(\mathcal{Q}) = \sum_{t=1}^{S} (\boldsymbol{G}^t)^{\mathrm{tr}} \cdot \boldsymbol{q}^t = \sum_{t=1}^{S} \sum_{i=1}^{L(K+1)} G^t_i q^t_i \to \min. \qquad (15)$$

In this framework, it is possible to optimize according to all objectives (O1) to (O5), choosing $\mathcal{G}$ adequately. (O1) It is sufficient to choose $\boldsymbol{G}^t = \boldsymbol{0}$ for time steps $t < S$ and $\boldsymbol{G}^S = (\boldsymbol{g}_1, \boldsymbol{0}, \boldsymbol{0}, \ldots, \boldsymbol{0})$. (O2) The same holds if g_1 is replaced by g_2, i.e., $\boldsymbol{G}^t = \boldsymbol{0}$ for time steps $t < S$ and $\boldsymbol{G}^S = (\boldsymbol{g}_2, \boldsymbol{0}, \boldsymbol{0}, \ldots, \boldsymbol{0}\ )$. (O3) The objective can be expressed by $G^t_i = 0$ unless $i \leq L$, $\boldsymbol{\alpha}_i := s^{-1}(i)$ and $E(\boldsymbol{\alpha}_i) = E_1$, in which case $G^t_i = -1$. (O4) To express this objective, it is sufficient to maximize $P^S_{\cap,E_1}$, which can be achieved by choosing $\boldsymbol{G}^t = \boldsymbol{0}$ for time steps $t < S$ but $G^S_i = 1$ for $L < i \leq 2L$, $\boldsymbol{\alpha}_{i-L} := s^{-1}(i-L)$ unless $E(\boldsymbol{\alpha}_{i-L}) = E_1$, in which case $G^S_i = 0$. (O5). Also this objective can be expressed within the framework, choosing (i) $\boldsymbol{G}^t = \boldsymbol{0}$ for $t < S$, (ii) $G^S_{Lk+s(\boldsymbol{\alpha})} = 0$ for $k \in \{0, 1, \ldots, K-1\}$, $\boldsymbol{\alpha} \in \boldsymbol{\Omega}$, and $E(\boldsymbol{\alpha}) > E_k$, and (iii) $G^S_{Lk+s(\boldsymbol{\alpha})} = E_k - E_{k+1}$ for $k \in \{0, 1, \ldots, K-1\}$, $\boldsymbol{\alpha} \in \boldsymbol{\Omega}$ and $G^S_{LK+s(\boldsymbol{\alpha})} = E_K$ for $\boldsymbol{\alpha} \in \boldsymbol{\Omega}$ and $E(\boldsymbol{\alpha}) > E_k$.

This can be obtained by expressing the *mean final best so far energy value* $\langle E^S_{\mathrm{BSF}} \rangle$ using different values of $P^S_{\cap,E_k}$ and a number of given equivalences as described above to obtain Eq. (15), which finally has to be minimized,

$$\langle E^S_{\mathrm{BSF}} \rangle = \sum_{k=1}^{K} P^S_{\cap,E_k} E_k \overset{(12)}{=} \sum_{k=1}^{K} (P^S_{\cup,E_k} - P^S_{\cup,E_{k-1}}) E_k$$
$$\overset{(11)}{=} \sum_{k=1}^{K} E_k \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_k}} p^S_{\boldsymbol{\alpha},E_k} - \sum_{k=1}^{R} E_k \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_{k-1}}} p^S_{\boldsymbol{\alpha},E_{k-1}}$$
$$\overset{(14)}{=} \sum_{k=1}^{K} E_k \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_k}} q^S_{Lk+s(\boldsymbol{\alpha})} - \sum_{k=1}^{K} E_k \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_{k-1}}} q^S_{L(k-1)+s(\boldsymbol{\alpha})}$$
$$= \sum_{k=1}^{K} E_k \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_k}} q^S_{Lk+s(\boldsymbol{\alpha})} - \sum_{k=0}^{K-1} E_{k+1} \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_k}} q^S_{Lk+s(\boldsymbol{\alpha})}$$
$$= \sum_{k=0}^{K-1} (E_k - E_{k+1}) \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_k}} q^S_{Lk+s(\boldsymbol{\alpha})} + E_K \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_K}} q^S_{LK+s(\boldsymbol{\alpha})}$$
$$\overset{\text{(iii)}}{=} \sum_{k=0}^{K-1} \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_k}} G^S_{Lk+s(\boldsymbol{\alpha})} q^S_{Lk+s(\boldsymbol{\alpha})} + \sum_{\substack{\boldsymbol{\alpha} \in \boldsymbol{\Omega}, \\ E(\boldsymbol{\alpha}) \leq E_K}} G^S_{LK+s(\boldsymbol{\alpha})} q^S_{LK+s(\boldsymbol{\alpha})}$$
$$\overset{\text{(ii)}}{=} \sum_{k=0}^{K} \sum_{\boldsymbol{\alpha} \in \boldsymbol{\Omega}} G^S_{Lk+s(\boldsymbol{\alpha})} q^S_{Lk+s(\boldsymbol{\alpha})} \overset{\text{(i)}}{=} \sum_{t=1}^{S} \sum_{i=1}^{L(K+1)} G^t_i q^t_i$$
$$\overset{(15)}{=} \sum_{t=1}^{S} (\boldsymbol{G}^t)^{\mathrm{tr}} \cdot \boldsymbol{q}^t = \mathcal{G}(\mathcal{Q}) \to \min.$$

Note that the objectives (O1) to (O5) are linear functions of the probabilities q^t_i, $i \in \{1, 2, \ldots, L(K+1)\}$, and $t \in \{1, 2, \ldots, S\}$. Now the following can be proven. Considering the randomized *selection* of an individual for the *crossover operator* in iteration t of a *genetic algorithm*, a *rectangular probability distribution* over the fittest individuals according to the *energy ranking* of all individuals of the current population, which is equivalent to a node of the set V as defined in Sec. IV gives the best implementation of this *selection step* in each generation, equivalent to the iterations $t = 1, 2, \ldots, S$.

Again, as in the proof for (O1) and (O2), the last step S has to be optimized first. The optimization problem as mentioned in Eq. (15) can be also expressed as

$$\mathcal{G}(\mathcal{Q}) = \sum_{t=1}^{S} (\boldsymbol{G}^t)^{\mathrm{tr}} \cdot \boldsymbol{q}^t$$
$$= \sum_{t=1}^{S} (\boldsymbol{G}^t)^{\mathrm{tr}} \cdot \hat{\boldsymbol{\Gamma}}^t \cdot \boldsymbol{q}^{t-1}$$
$$= (\boldsymbol{G}^S)^{\mathrm{tr}} \cdot \hat{\boldsymbol{\Gamma}}^S \cdot \boldsymbol{q}^{S-1} + c \to \min$$

by Eq. (13). We again focus on one arbitrary distribution $\boldsymbol{d}^{c_S,S}$, $c_s \in \{1, 2, \ldots, m\}$ to be optimized. The values $\hat{\Gamma}^S_{\boldsymbol{\alpha\beta},E}$ as defined in Eq. (9) still depend only linearly on the probabilities $\Gamma^S_{\boldsymbol{\alpha\beta}}$ and hence also only linearly on $\boldsymbol{d}^{c_S,S}$. The possible distributions according to (A2) and (A3) form a simplex just

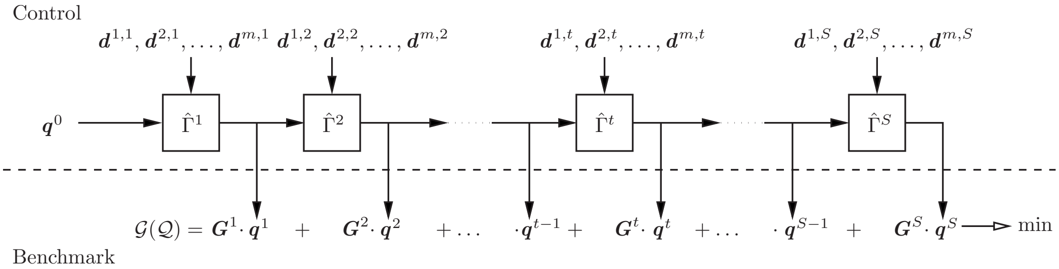
Control
$d^{1,1}, d^{2,1}, \ldots, d^{m,1}$
$d^{1,2}, d^{2,2}, \ldots, d^{m,2}$
$d^{1,t}, d^{2,t}, \ldots, d^{m,t}$
$d^{1,S}, d^{2,S}, \ldots, d^{m,S}$
q^0
$\hat{\Gamma}^1$
$\hat{\Gamma}^2$
$\hat{\Gamma}^t$
$\hat{\Gamma}^S$
$\mathcal{G}(\mathcal{Q}) = G^1 \cdot q^1 + G^2 \cdot q^2 + \ldots \cdot q^{t-1} + G^t \cdot q^t + \ldots \cdot q^{S-1} + G^S \cdot q^S$ ⟶ min
Benchmark

FIG. 3. Dynamic optimization process, transforming vectors $\boldsymbol{q}^{t-1}$ in $\boldsymbol{q}^t$ depending on the distributions $\boldsymbol{d}^{i,t}$, $i=1,2,\ldots,m$ for the time steps $t=1,2,\ldots,S$.

as argued already in the proof in Sec. V and hence, independently from the choice of the other distributions $\boldsymbol{d}^{i,S}$, $i \neq c_S$, an optimal value of $\mathcal{G}(\mathcal{Q})$ can be obtained for a vertex $\boldsymbol{v}^{c_S,S} := \boldsymbol{v}_i \in V$, which is a rectangular distribution as considered in the threshold-selecting approach.

Because c_S has been chosen arbitrary from $\{1,2,\ldots,m\}$, this holds for all other values of c_S as well; i.e., in an optimal combination of m distributions $\boldsymbol{d}^{1,S}$, $\boldsymbol{d}^{2,S},\ldots,\boldsymbol{d}^{m,S}$, the distributions are equivalent to vertices $\boldsymbol{v}^{1,S}$, $\boldsymbol{v}^{2,S},\ldots,\boldsymbol{v}^{m,S} \in V$. Which of these vertices are chosen depends only on the input vector $\boldsymbol{q}^{S-1}$ and this is determined by the pervious iterations of the algorithm. In stage of the proof, the vector is unknown because the distributions of the former steps still have to be fixed. Thus, we again have n^m possible combinations of the distributions $\boldsymbol{d}^{i,S}$, $i=1,2,\ldots,m$ but at least one of them must be optimal. Now this construction can be continued in each of the n^m search branches with the step $S-1$, where the objective can be determined by considering the m probability distributions in time step S to be chosen. In step $S-1$, we have the minimization

$$\begin{aligned}\mathcal{G}(\mathcal{Q}) &= (\boldsymbol{G}^S)^{\mathrm{tr}} \cdot \hat{\boldsymbol{\Gamma}}^S(\boldsymbol{v}^{1,S},\boldsymbol{v}^{2,S},\ldots,\boldsymbol{v}^{m,S}) \cdot \hat{\boldsymbol{\Gamma}}^{S-1} \cdot \boldsymbol{q}^{S-2} \\ &\quad + (\boldsymbol{G}^{S-1})^{\mathrm{tr}} \cdot \hat{\boldsymbol{\Gamma}}^{S-1} \cdot \boldsymbol{q}^{S-2} + c' \\ &= [(\boldsymbol{G}^S)^{\mathrm{tr}} \cdot \hat{\boldsymbol{\Gamma}}^S(\mathcal{V}^S) + (\boldsymbol{G}^{S-1})^{\mathrm{tr}}] \cdot \hat{\boldsymbol{\Gamma}}^{S-1} \cdot \boldsymbol{q}^{S-2} + c' \\ &= (\tilde{\boldsymbol{G}}^{S-1})^{\mathrm{tr}} \cdot \hat{\boldsymbol{\Gamma}}^{S-1} \cdot \boldsymbol{q}^{S-2} + c' \to \min .\end{aligned}$$

Here $\mathcal{V}^S$ denotes the set $\boldsymbol{v}^{1,S}$, $\boldsymbol{v}^{2,S},\ldots,\boldsymbol{v}^{m,S}$ of vertices and $(\tilde{\boldsymbol{G}}^{S-1})^{\mathrm{tr}}$ is the implicitly defined objective function for step $S-1$. Also in this step, the dependence on each single chosen distribution $\boldsymbol{d}^{i,S-1}$, $i=1,2,\ldots,m$ is only linear and with the same argument as above there is an optimal combination $\boldsymbol{d}^{1,S-1}$, $\boldsymbol{d}^{2,S-1},\ldots,\boldsymbol{d}^{m,S-1}$ of distributions, equivalent to vertices $\boldsymbol{v}^{1,S-1}$, $\boldsymbol{v}^{2,S-1},\ldots,\boldsymbol{v}^{m,S-1} \in V$. These chosen vertices depend only on the input vector $\boldsymbol{q}^{S-2}$.

The remaining steps $S-2$, $S-3,\ldots,1$ can be processed in a similar way and in each step, distributions equivalent to vertices from V are obtained as optimal solutions. The equivalent dynamics is visualized in Fig. 3. This completes the proof. Hence, threshold selecting is also for the objectives (O3) to (O5) an optimal strategy.

VII. CONCLUSIONS

The problem of selecting individuals from the population of a GA for crossover based on their energy values has been considered by applying the master equation to describe the corresponding dynamics as a random walk in the state space and some straightforward assumptions on the probability distributions for selecting the individuals in a certain generation have been formulated. Our goal was to find transition probabilities assuring the optimal control of the evolutionary development in the GA. A rectangular distribution of selection probabilities is provably optimal, provided that the performance is measured by a linear function in modified state probabilities, which includes many reasonable choices for the objective function.

The proof above is based on the fundamental theorem of linear programming, which states that a linear function defined on a simplex reaches its minimum at a vertex. However, the proof does not state that all optimal selection strategies in GAs are rectangular. Other strategies may do equally well but not better.

If there exists an optimal strategy other than *threshold selecting*, it follows that an edge or a face of the described simplex gives equivalent results. Thus, it seems unlikely that a strictly monotonic distribution can be optimal [15], which would imply that all the vertices in V perform equally well.

As presented, the proof can be applied for any crossover procedure in GAs with independent probability distributions for the selection of the crossover individuals and for both, the *generation replacement model*, where the mating pool has size n for populations of size n, and also for the *steady-state replacement model*, where only some individuals are replaced [34]. Integrating a mutation operator, the same as for crossover holds as described in Sec. III.

Currently, the knowledge that the best performance can be achieved using *threshold selecting* is only of limited use, since the cutoff ranks ν to be used are not known *a priori*. Therefore, it would be interesting to carry out numerical experiments, comparing distributions empirically. Furthermore, it is reasonable to introduce a schedule on the cutoff rank ν, narrowing the rectangular distribution during the optimization process and thus increasing the evolutionary pressure gradually. Moreover, it would be interesting to obtain also theoretical progress concerning the choice of one of the possible rectangular distributions or to reduce the choice to a certain assortment.

Another future research task is the investigation of the influence caused by transformations of the energy function as introduced in [35].

[1] C. Meinel and T. Theobald, *Algorithms and Data Structures in VLSI Design*, 2nd ed. (Springer-Verlag, Berlin, 2005).

[2] Q. Yang, *Intelligent Planning: A Decomposition and Abstraction Based Approach* (Springer-Verlag, Berlin, Heidelberg, 1997).

[3] M. L. Pinedo, *Planning and Scheduling in Manufacturing and Services* (Springer-Verlag, New York, 2005).

[4] K. H. Hoffmann and M. Schreiber, *Computational Statistical Physics* (Springer, Berlin, 2001).

[5] R. Rojas, *Neural Networks: A Systematic Introduction* (Springer-Verlag, Berlin, 1996).

[6] Y. Zhang and J. Skolnick, *Proceedings of the National Academy of Sciences USA*, 2005, Vol. 102, pp. 1029–1034.

[7] U. Hansmann, Eur. Phys. J. B **12**, 607 (1999).

[8] M. Sipser, *Introduction to the Theory of Computation* (PWS, Boston, 1997).

[9] S. Kirkpatrick, C. Galatt, and M. Vecchi, Science **4598**, 671 (1983).

[10] G. Dueck and T. Scheuer, J. Comput. Phys. **90**, 161 (1990).

[11] J. Kennedy and R. Eberhart, in *Proceedings of the IEEE International Conference on Neural Networks, IV* (IEEE, Perth, 1995), pp. 1942-1948

[12] J.-P. Rennard, Introduction to Genetic Algorithms, 2000, http://www.rennard.org/alife/english/gavgb.pdf

[13] A. Franz and K. H. Hoffmann, Appl. Math. Lett. **16**, 27 (2003).

[14] A. Franz and K. H. Hoffmann, J. Comput. Phys. **176**, 196 (2002).

[15] A. Franz, K. H. Hoffmann, and P. Salamon, Phys. Rev. Lett. **86**, 5219 (2001).

[16] F. Heilmann, K. H. Hoffmann, and P. Salamon, Europhys. Lett. **66**, 305 (2004).

[17] K. H. Hoffmann, F. Heilmann, and P. Salamon, Phys. Rev. E **70**, 046704 (2004).

[18] P. G. Busacca, M. Marseguerra, and E. Zio, Reliab. Eng. Syst. Saf. **72**, 59 (2001).

[19] L. D. Chambers, *The Practical Handbook of Genetic Algorithms: Applications* (CRC, Boca Raton, FL, 2000).

[20] P. Godefroid and S. Khurshid, in *LNCS 2280*, edited by J.-P. Katoen and P. Stevens (Springer-Verlag, Berlin, Heidelberg, 2002), pp. 266–280.

[21] B. Hemmateenejad, M. Akhond, R. Miri, and M. Shamsipur, J. Chem. Inf. Model. **43**, 1328 (2003).

[22] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, Bioinformatics **19**, 643 (2003).

[23] A. Kolen, Stat. Neerl. **61**, 4 (2007).

[24] M. Chakraborty and U. K. Chakraborty, *Proceedings of the International Conference on Information, Communications and Signal Processing* (IEEE, Singapore, 1997), pp. 407–411.

[25] U. K. Chakraborty, Inf. Process. Lett. **56**, 281 (1995).

[26] U. K. Chakraborty, K. Deb, and M. Chakraborty, Evol. Comput. **4**, 133 (1996).

[27] U. K. Chakraborty, Inf. Process. Lett. **56**, 77 (1995).

[28] H. Mühlenbein and D. Schlierkamp-Voosen, Evol. Comput. **1**, 335 (1993).

[29] *Foundations of Genetic Algorithms: 9th International Workshop, FOGA 2007*, edited by C. R. Stephens, M. Toussaint, D. Whitley, and P. F. Stadler (Springer, Berlin, Mexico City, 2007).

[30] J. Lässig, K. H. Hoffmann, and M. Enăchescu, *International Genetic and Evolutionary Computation Conference 2008* (ACM, Atlanta, Georgia, 2008), pp. 2181–2185.

[31] C. Tsallis and D. A. Stariolo, Physica A **233**, 395 (1996).

[32] S. Boettcher and A. G. Percus, in Proceedings of the Genetic and Evolutionary Computation Conference (ACM, Orlando, FL, 1999), pp. 825–832.

[33] R. E. Bellman, *Dynamic Programming* (Princeton University Press, Dover, Mineola, NY, 2003).

[34] D. Srinivasan and L. Rachmawati, *Proceedings of the Genetic and Evolutionary Computation Conference* (ACM, Seattle, WA, 2006), pp. 715–722.

[35] W. Wenzel and K. Hamacher, Phys. Rev. Lett. **82**, 3003 (1999).