

Comparing Evolutionary Algorithms to the (1+1)-EA

P. A. Borisovsky¹, A. V. Eremeev^{*}

Omsk Branch of Sobolev Institute of Mathematics, Siberian Branch of Russian Academy of Sciences, st. Pevtsov, 13, 644099, Omsk, Russia

Abstract

In this paper, we study the conditions in which the (1+1)-EA compares favorably to other evolutionary algorithms (EAs) in terms of fitness function distribution at given iteration and with respect to the average optimization time. Our approach is applicable when the reproduction operator of an evolutionary algorithm is dominated by the mutation operator of the (1+1)-EA. In this case one can extend the lower bounds obtained for the expected optimization time of the (1+1)-EA to other EAs based on the dominated reproduction operator. This method is exemplified on the sorting problem with HAM landscape and the exchange mutation operator. We consider several simple examples where the (1+1)-EA is the best possible search strategy in the class of the EAs.

Key words: Evolutionary Algorithm, Comparison, Optimization, Monotonicity, Domination

1 Introduction

Let the optimization problem consist in finding a feasible solution $x \in Sol \subseteq X$, which optimizes the objective function $f : Sol \rightarrow \mathbf{R}$, where X is the space of solutions, $Sol \subseteq X$ is a set of feasible solutions, and \mathbf{R} is the set of real numbers. In this paper we assume that X is a subset of Euclidean finite-dimensional space. If necessary, the spaces of solutions of the discrete

^{*} Corresponding author.

Email addresses: borisovski@mail.ru (P. A. Borisovsky),
eremeev@iitam.omsk.net.ru (A. V. Eremeev).

¹ Present address: Omsk State Technical University, Prospect Mira 11, 644050 Omsk, Russia.

optimization problems (e.g. the Boolean cube $\{0, 1\}^n$ or the set of all permutations of n elements) will be considered as discrete subsets of the Euclidean space as well.

In general, an evolutionary algorithm searches for the optimal or near-optimal solutions using a population of individuals, which is driven by the principles observed in biological evolution. This paper is devoted to comparison of the evolutionary algorithms to the (1+1)-EA, one of the most simple evolutionary algorithms with the population consisting of a single individual (see e.g. [17]).

The search process is guided by evaluations of a fitness function $\Phi(x)$, which defines the fitness of an individual x . In case $x \in Sol$, it is supposed that Φ is a monotone function of $f(x)$: non-decreasing for maximization problems and non-increasing for minimization problems. In case $x \notin Sol$, the fitness function may incorporate a penalty for violation of constraints defining the set Sol . Through this paper we will assume that if $Sol \neq X$ then for any $x' \in X \setminus Sol$ holds $\Phi(x') < \max_{x \in Sol} \Phi(x)$.

2 Framework of Analysis and Comparison Theorem

In order to set up the standard probabilistic framework for analysis of the random operators of evolutionary algorithms, let us assume that a random operator $Op(u)$ maps some subset U of a Euclidean space E' into a Euclidean space E with the following three properties:

(i) Given a particular input $u \in U$, the output of $Op(u)$ is a random variable on E with a probability measure $p_{Op}(u, S) = P\{Op(u) \in S\}$, defined for all $S \subseteq E, S \in B_E$, where B_E denotes the Borel field of subsets on E . (ii) We will impose a standard assumption from the random processes theory, that the transition function $p_{Op}(u, S)$ is a B_U -measurable function in u for each $S \in B_E$, where B_U is the restriction of the Borel field of subsets on E' to the set U (see e.g. [9]). (iii) Besides that, if a random operator Op is used in some algorithm, then under condition of specific given input u , the outcome $Op(u)$ is independent of all other preceding events, that have occurred in computation of the algorithm.

The outcome of an operator may depend on the specific input data of the problem instance, which is assumed to be fixed (not random) in this paper. In what follows, the objective function $\Phi(x)$ is supposed to be a fixed function, defined by the problem instance. It is only required that Φ is measurable.

Most of the well-known models of random computation, e.g. the randomized Turing machine, in finite time allow to represent numbers only from a count-

able set of values. All random operators and functions computable in such models satisfy the measurability requirements (i) and (ii) above.

The current individual on iteration t of the (1+1)-EA will be denoted by $x^{(t)}$. The initial solution $x^{(0)}$ is generated with some a priori chosen probability distribution. Each new individual is built with the help of a random *mutation operator* $\text{Mut} : X \rightarrow X$, which adds some random changes to the solution. The mutation operator is applied to $x^{(t)}$ and if $x = \text{Mut}(x^{(t)})$ is such that $\Phi(x) > \Phi(x^{(t)})$, then we set $x^{(t+1)} := x$; otherwise $x^{(t+1)} := x^{(t)}$. We do not fix a particular stopping criterion, because it is not necessary in our analysis.

One of the frequently used mutation operators in the case $X = \{0, 1\}^n$ is the *standard mutation*, which consists in changing each bit with a fixed mutation probability p_{mut} . Other simple examples are the *1-bit-flip* mutation operator which chooses a random position i and changes the i -th bit (see e.g. [18]) and the isotropic normal mutation where the current solution in \mathbb{R}^n is summed to a random normally distributed variable with zero mean [5,17].

In order to define the general scheme of an evolutionary algorithm we will assume that the random reproduction operator Rep has a set of solutions a^1, a^2, \dots, a^r on its input and produces a random output of s offspring

$$(b^1, b^2, \dots, b^s) = \text{Rep}(a^1, a^2, \dots, a^r),$$

possibly using the specific data of the problem being solved. In general the reproduction operator may be a problem specific randomized heuristic including recombination, mutation and local improvement procedures. In the case of genetic algorithm the operator Rep consists of crossover and mutation.

Let us consider an evolutionary algorithm EA which corresponds to the following scheme: the initial set of solutions $a^{(0,1)}, \dots, a^{(0,N)}$ is given (deterministic or randomly generated), and on each iteration $t > 0$ a new group of individuals $a^{(t,1)}, \dots, a^{(t,s)}$ is produced by applying $\text{Rep}(c^1, \dots, c^r)$ where c^1, \dots, c^r are some individuals already generated before, i.e. $c^k \in A^{(t-1)}$ where

$$A^{(t-1)} = \{a^{(0,l)} : l = 1, \dots, N\} \cup \{a^{(\tau,j)} : \tau = 1, \dots, t-1, j = 1, \dots, s\}.$$

We will denote the sequence of solutions generated in the EA before iteration t by σ^{t-1} . Both $A^{(t-1)}$ and σ^{t-1} are the random values. Formally,

$$\sigma^{t-1} = a^{(0,1)}, \dots, a^{(0,N)}, a^{(1,1)}, \dots, a^{(1,s)}, \dots, a^{(t-1,1)}, \dots, a^{(t-1,s)}.$$

The parents c^1, \dots, c^r on iteration t are chosen by a random operator of *selection* $\text{Sel} : X^{N+s(t-1)} \rightarrow X^r$, such that $\text{Sel}(\sigma^{t-1}) \subseteq (A^{(t-1)})^r$.

The random operators Rep and Mut may vary, as a function of the current iteration number t , but we will not denote this explicitly for the sake of notation

simplicity.

It is easy to see that this scheme covers most of the evolutionary algorithms such as the genetic algorithms [12], many versions of the genetic programming algorithms [15] and of the $(\mu + \lambda)$ -EA – see e.g. [5,17], ”go with the winners” algorithms [1] and the simulated annealing [14].

In what follows, we will denote the maximum of fitness function on a sequence of solutions by $\check{\Phi}(\cdot)$, i.e. for a sequence $y = (y_1, \dots, y_k)$, $\check{\Phi}(y) = \max_{i=1, \dots, k} \Phi(y_k)$.

We will compare the reproduction operator of EA and a mutation operator of the (1+1)-EA using the following

Definition 1 *Reproduction operator Rep is dominated by mutation operator Mut if for arbitrary r -element sequence of individuals $a = (a^1, \dots, a^r)$ and arbitrary $x \in X$ such that $\Phi(x) \geq \check{\Phi}(a)$, the following condition holds for all $\phi \in \mathbf{R}$:*

$$P\{\Phi(\text{Mut}(x)) \geq \phi\} \geq P\{\check{\Phi}(\text{Rep}(a)) \geq \phi\}. \quad (1)$$

Theorem 1 (*Comparison Theorem*) *Suppose the reproduction operator Rep of EA is dominated by a mutation operator Mut of the (1+1)-EA on each iteration $t > 0$. Let the (1+1)-EA start from $x^{(0)}$ such that $\Phi(x^{(0)}) \geq \max_{i=1, \dots, N} \Phi(a^{(0,i)})$. Then for all $t \geq 0$ and all $\phi \in \mathbf{R}$ holds*

$$P\{\Phi(x^{(t)}) \geq \phi\} \geq P\{\check{\Phi}(\sigma^t) \geq \phi\}. \quad (2)$$

Proof. The scheme of the proof is analogous to that of the Fundamental Theorem for stochastically monotone Markov Chains [9].

Assume by induction that (2) holds for all $\tau < t$ (we have the basis of induction for $t = 0$ by assumption of the Theorem). Let us fix an arbitrary $\phi \in \mathbf{R}$ and define $p(x) = P\{\Phi(\text{Mut}(x)) \geq \phi\}$ for every $x \in X : \Phi(x) < \phi$, and $p(x) = 1$ if $\Phi(x) \geq \phi$. Using the conditional distribution of $\Phi(x^{(t)})$ given $x^{(t-1)}$ (see e.g. [8], p.84) we obtain

$$P\{\Phi(x^{(t)}) \geq \phi\} = P\{\Phi(x^{(t-1)}) \geq \phi\} + \int_{\{x: \Phi(x) < \phi\}} P\{\Phi(\text{Mut}(x)) \geq \phi \mid x^{(t-1)} = x\} dP(x) = E[p(x^{(t-1)})]$$

Let us introduce a function $\delta(\psi) = \inf_{x: \Phi(x) \geq \psi} p(x)$. Note that $\delta(\psi)$ is non-

decreasing. For any positive integer K define a sequence of subsets

$$A_n = \left\{ \psi : \frac{n}{K} \leq \delta(\psi) < \frac{n+1}{K} \right\}, n = 0, \dots, K-1,$$

$$A_K = \{ \psi : \delta(\psi) = 1 \}.$$

Let $X_n = \Phi^{-1}(A_n)$. For every $x \in X_n$ we have $\Phi(x) \in A_n$ so $\delta(\Phi(x)) \geq n/K$ and $p(x) \geq n/K$. Thus,

$$E[p(x^{(t-1)})] = \sum_{n=0}^K \int_{X_n} p(x^{(t-1)}) dP(x^{(t-1)}) \geq \frac{1}{K} \sum_{n=0}^K n \int_{X_n} dP(x^{(t-1)}) =$$

$$\frac{1}{K} \sum_{n=0}^K n P\{x^{(t-1)} \in X_n\} = \frac{1}{K} \sum_{n=0}^K n P\{\Phi(x^{(t-1)}) \in A_n\} = \frac{1}{K} \sum_{n=1}^K P\{\Phi(x^{(t-1)}) \in B_n\},$$

where $B_n = A_n \cup A_{n+1} \cup \dots \cup A_K$. Due to the monotonicity of $\delta(\psi)$ each set B can be represented in a form $[b_n, +\infty)$ or $(b_n, +\infty)$, where b_0, b_1, \dots, b_K is some nondecreasing sequence (it may be that $b_i = +\infty$ for some $i = 1, \dots, K$, in this case all B_i, B_{i+1}, \dots, B_K are empty).

Analogously, for a sequence $\sigma \in X^{N+s(t-1)}$ we define

$$q(\sigma) = P\{\check{\Phi}(\text{Rep}(\text{Sel}(\sigma))) \geq \phi\}$$

if $\check{\Phi}(\sigma) < \phi$ and $q(\sigma) = 1$ if $\check{\Phi}(\sigma) \geq \phi$. Then

$$P\{\check{\Phi}(\sigma^t) \geq \phi\} = E[q(\sigma^{(t-1)})]$$

Consider $Y_n = \check{\Phi}^{-1}(A_n)$ in the space $X^{N+s(t-1)}$. If $\sigma^{t-1} \in Y_n$ then $\check{\Phi}(\sigma^{t-1}) \in A_n$, $\delta(\check{\Phi}(\sigma^{t-1})) \leq (n+1)/K$. Due to the domination assumption, $q(\sigma^{(t-1)}) \leq p(x)$ for all $x : \check{\Phi}(x) \geq \check{\Phi}(\sigma^{t-1})$, so $q(\sigma^{(t-1)}) \leq \delta(\check{\Phi}(\sigma^{t-1})) \leq (n+1)/K$ and

$$P\{\check{\Phi}(\sigma^t) \geq \phi\} \leq \sum_{n=0}^K \int_{A_n} \frac{n+1}{K} dF_{\check{\Phi}(\sigma^{t-1})}(\psi) \leq$$

$$\frac{1}{K} + \frac{1}{K} \sum_{n=0}^K n P\{\check{\Phi}(\sigma^{t-1}) \in A_n\} = \frac{1}{K} + \frac{1}{K} \sum_{n=1}^K P\{\check{\Phi}(\sigma^{t-1}) \in B_n\}.$$

By the inductive assumption

$$P\{\Phi(x^{(t-1)}) \in B_n\} \geq P\{\check{\Phi}(\sigma^{t-1}) \in B_n\},$$

so

$$P\{\Phi(x^{(t)}) \geq \phi\} - P\{\check{\Phi}(\sigma^t) \geq \phi\} \geq \frac{1}{K} \sum_{n=1}^K \left(P\{\Phi(x^{(t-1)}) \in B_n\} - P\{\check{\Phi}(\sigma^{t-1}) \in B_n\} \right) - \frac{1}{K} \geq -\frac{1}{K}$$

Finally, setting $K \rightarrow \infty$, we obtain (2). Q.E.D.

2.1 Monotone Reproduction

Now we will see that in some cases there is a natural way to construct a mutation operator allowing the (1+1)-EA to work at least as good as any EA using a particular reproduction operator Rep.

Let us define a mutation operator corresponding to Rep as

$$\text{Mut}_{\text{Rep}}(x) = \operatorname{argmax} (\Phi(x), \Phi(c_1), \dots, \Phi(c_s)),$$

where $(c_1, \dots, c_s) = \text{Rep}(x, \dots, x)$, i.e. in $\text{Mut}_{\text{Rep}}(x)$ firstly the reproduction Rep is applied to a set of identical parent individuals and then the output is chosen as the fittest among the parent and the offspring. In case there are several offspring with maximal fitness we assume that $\text{Mut}_{\text{Rep}}(x)$ is uniformly distributed among them.

In order to identify the situation where Mut_{Rep} is helpful, we will use the following definition.

Definition 2 *Reproduction operator Rep is called monotone if for arbitrary r -element sequences a^1, \dots, a^r and g^1, \dots, g^r such that*

$$\Phi(a^1) \leq \Phi(g^1), \dots, \Phi(a^r) \leq \Phi(g^r), \quad (3)$$

the following conditions hold for all ϕ :

$$P \left\{ \max_{i=1, \dots, s} \Phi(h^i) \geq \phi \right\} \geq P \left\{ \max_{i=1, \dots, s} \Phi(b^i) \geq \phi \right\}, \quad (4)$$

where $(b^1, \dots, b^s) = \text{Rep}(a^1, \dots, a^r)$ and $(h^1, \dots, h^s) = \text{Rep}(g^1, \dots, g^r)$.

Informally, this definition requires that substitution of parent genotypes by genotypes with greater or equal fitness should never decrease the chances of obtaining sufficiently good offspring. Note that if all conditions (3) are equalities for the sets of parents a^1, \dots, a^r and g^1, \dots, g^r , then the probability distributions of the best offspring fitness for $\text{Rep}(a^1, \dots, a^r)$ and $\text{Rep}(g^1, \dots, g^r)$ must coincide.

If operator Rep is monotone, then Mut_{Rep} dominates Rep by construction, so in this case we can compare an EA with reproduction Rep to the (1+1)-EA based on Mut_{Rep} by the means of Theorem 1:

Corollary 2 *Suppose a monotone reproduction operator Rep is used in the EA and the operator Mut_{Rep} is used in the (1+1)-EA. Let the (1+1)-EA start*

from $x^{(0)}$ such that $\Phi(x^{(0)}) \geq \max_{i=1, \dots, N} \Phi(a^{(0,i)})$. Then for all $t \geq 0$ and all $\phi \in \mathbb{R}$ holds

$$P\{\Phi(x^{(t)}) \geq \phi\} \geq P\{\check{\Phi}(\sigma^t) \geq \phi\}. \quad (5)$$

A simple proof of this fact in the case of discrete optimization problems can be found in [6].

A special case of monotone reproduction operator with $r = s = 1$ is the monotone mutation operator (note that if Rep is monotone then Mut_{Rep} is monotone too). It is important that a mutation operator dominates itself iff it is monotone, so in this case the (1+1)-EA is an optimal EA, as shown by inequality (5).

2.2 Examples of Monotone Reproduction Operators

A simple example of monotone mutation can be demonstrated on maximization of fitness function $\Phi(x) \equiv \text{ONEMAX}(x) \equiv x_1 + \dots + x_n$ with $\text{Sol} = X = \{0, 1\}^n$, when the standard mutation operator is used and $p_{\text{mut}} \leq 1/2$ (see the proof of monotonicity in [11]). In fact it was shown in [6] that if there is a unique maximum x^* of $\Phi(x)$ on $\{0, 1\}^n$ and $|\text{Range}(\Phi)| = n + 1$, then Mut is monotone iff Φ belongs to a class of functions ONEMAX^{**} introduced in [10]. To describe ONEMAX^{**} we first need to define ONEMAX^* as the class of functions

$$\text{ONEMAX}_a(x) = \sum_{i=1}^n ((x_i + a_i) \bmod 2),$$

where $x, a \in \{0, 1\}^n$. Then, ONEMAX^{**} by definition consists of all functions $\mu \circ f$ where $f \in \text{ONEMAX}^*$ and $\mu : \mathbb{R} \rightarrow \mathbb{R}$ is a strictly increasing function.

Another case of monotonicity in discrete optimization is established for a family of instances of the vertex cover problem. In general, given a graph $G = (V, E)$, the vertex cover problem (VCP) asks for a subset $C \subset V$ (called a *vertex cover*), such that every edge $e \in E$ has at least one endpoint in C . The size of C should be minimized. Let us consider the *non-binary representation* (see e.g. [4]) of the problem solutions, where $X = \{0, 1\}^{|E|}$ and each coordinate $x^i \in \{0, 1\}, i = 1, \dots, |E|$ of x corresponds to an edge $e_i \in E$, assigning one of its endpoints to be included into the cover $C(x)$ (one endpoint of e_i is assigned if $x^i = 0$ and the other one is if $x^i = 1$). Thus, $C(x)$ contains all vertices, assigned by at least one of the coordinates of x , and the feasibility of $C(x)$ is guaranteed. The fitness function is by definition $\Phi(x) \equiv |V| - |C(x)|$. It is shown in [11] that in this case the standard mutation operator is monotone on

the family of VCP instances, where G consists of $k = |V|/3$ disjoint cliques of size 3.

As an example of mutation monotonicity in continuous optimization, consider minimization of the Euclidean vector norm $f(x) \equiv \|x\|$ in $Sol = X = \mathbf{R}^n$ (also called *the quadratic sphere function*), using the isotropic normal mutation. Formally this case may be characterized by $\Phi(x) \equiv -\|x\|$ and operator $\text{Mut}(x)$ with normal distribution (x, σ) , where σ is a diagonal matrix with identical diagonal elements.

Let us now consider a simple example illustrating the monotone reproduction with $r = 2, s = 1$. Suppose we have an arbitrary function $\Phi \in \text{ONEMAX}^{**}$ and Rep is the standard 1-point or uniform crossover operator, but the genes in one of the parent individuals are randomly permuted before the crossover. It is not difficult to see that in this case Rep is monotone, and it may not be monotone without the random permutation phase.

For the continuous optimization in \mathbf{R}^n the monotone reproduction with “scalable” values $r = \mu, s = \lambda, \lambda \geq \mu$ can be exemplified by the reproduction operator of $(\mu/\mu_I, \lambda)$ -ES evolution strategy (see e.g. [5]) in the case of a linear fitness function. The reproduction of $(\mu/\mu_I, \lambda)$ -ES consists in computing the centroid $\bar{a} = \frac{1}{\mu} \sum_{k=1}^{\mu} a^k$ of the parent solutions a^1, \dots, a^{μ} , and building independently λ descendants by means of isotropic normal mutation of the vector \bar{a} .

The monotonicity condition in Corollary 2 may be relaxed in the following way. We will call Rep *weakly monotone* if the inequality (4) holds at least for all

$$\phi > \max\{\Phi(h^k) : k = 1, \dots, r\}.$$

It is easy to see that weak monotonicity of Rep implies monotonicity of Mut_{Rep} .

In all examples given above we deal with the problems of regular structure. In practice, however, the optimization problems tend to have highly irregular structure and monotonicity might be a rare case. For instance, if $X = \{0, 1\}^n$, then existence of a local optimum of Φ , which is not global, in terms of Hamming distance, implies that the 1-bit-flip mutation is non-monotone.

In situations where the monotonicity condition is violated, the (1+1)-EA may be less efficient than other EAs. For example, the $(\mu/\mu_I, \lambda)$ -evolution strategy is known to outperform the (1+1)-EA on the quadratic sphere function in presence of normal additive noise (see e.g. [2]). An illustration of the similar behavior of genetic algorithm in case of discrete optimization problems can be found in [13].

Theorem 1 yields only a sufficient condition of optimality of the (1+1)-EA in the class of evolutionary algorithms. However, the (1+1)-EA remains a

method of choice in many situations where the mutation is non-monotone, and the computational experiments indicate, that it is useful to consider some relaxed versions of monotonicity, e.g. monotonicity on average [7].

2.3 Expected Hitting Times and Expected Fitness

Let us now consider the average number of calls to the reproduction operator made in search for the solution of required quality (e.g. the optimal one) and the expected fitness after a given number of the calls. We will denote by $t_\phi^{(1+1)}$ the expected number of iterations until level $\Phi(x) = \phi$ or greater is reached by the (1+1)-EA. Analogous hitting time for the EA is t_ϕ^{EA} .

Corollary 3 *In conditions of Theorem 1:*

- (i) *If Φ is bounded on X then for all t holds $E[\Phi(x^{(t)})] \geq E[\check{\Phi}(\sigma^t)]$;*
- (ii) *if t_ϕ^{EA} is finite then $t_\phi^{EA} \geq t_\phi^{(1+1)}$.*

Proof. (i) easily follows from the properties of expectations of stochastically comparable random variables (see e.g. Chapt. IV[16]).

(ii) Let t be a non-negative integer random variable. Then

$$E[t] = \sum_{n=1}^{\infty} nP\{t = n\} = \sum_{n=1}^{\infty} \sum_{k=1}^n P\{t = n\} = \sum_{k=1}^{\infty} \sum_{n=k}^{\infty} P\{t = n\} = \sum_{k=0}^{\infty} P\{t > k\}.$$

The change of summation order is valid here if $E[t]$ is finite. For $t = t_\phi^{EA}$ this implies

$$t_\phi^{EA} = \sum_{k=0}^{\infty} P\{\check{\Phi}(\sigma^k) < \phi\} = \sum_{k=0}^{\infty} (1 - P\{\check{\Phi}(\sigma^k) \geq \phi\}),$$

and analogously for $t = t_\phi^{(1+1)}$,

$$t_\phi^{(1+1)} = \sum_{k=0}^{\infty} (1 - P\{\Phi(x^{(k)}) \geq \phi\}).$$

Application of Theorem 1 completes the proof. Q.E.D.

Theorem 1 and Corollary 3 show that if the choice is to be made between an EA with a reproduction operator Rep and the (1+1)-EA with mutation operator Mut, in situation where Mut dominates Rep, and these operators have the same computational cost, then the (1+1)-EA should be preferred.

3 Lower Bounds on Expected Running Times

Let ϕ^* be the maximal fitness value for a given instance of optimization problem. Suppose there exists some mutation operator Mut , dominating reproduction operator Rep . In case there exists a lower bound L on $t_{\phi^*}^{(1+1)}$, this will imply by Corollary 3 that for any EA based on Rep , the hitting time for the optimum is at least L . To illustrate this approach we will consider the sorting problem as a discrete optimization problem following the framework suggested by Scharnow, Tinnefeld and Wegener in [19].

Given an order \preceq on the set of distinct elements $\{1, \dots, n\}$, the standard sorting problem asks for such permutation π^* of elements $\{1, \dots, n\}$ that $\pi^*(1) \preceq \pi^*(2) \preceq \dots \preceq \pi^*(n)$. In what follows, sorting is considered as an optimization problem of maximizing the sortedness function $HAM(\pi) = |\{i : \pi(i) = \pi^*(i)\}|$ on the set of all permutations. Here $HAM(\pi)$ gives the number of correctly set positions in $(\pi(1), \pi(2), \dots, \pi(n))$. From the practical point of view this approach is unlikely to be useful, since the evaluation of $HAM(\pi)$ is usually impossible until one knows π^* ; however, it is of interest for the theoretical analysis.

In the framework of this paper, we will consider all permutations as vectors of the form $(\pi(1), \dots, \pi(n)) \in \mathbf{R}^n$, thus X is the set of all vectors of permutations, $\Phi(x) \equiv HAM(x)$ for any $x \in X$ and $\phi^* = n$.

Without loss of generality we will assume that $\pi^*(i) = i$ for all $i = 1, \dots, n$. This assumption is legitimate due to the symmetry of the set of permutations, as long as the EAs obtain the problem-specific data only through the evaluation of $\Phi(x)$ (i.e. this is a black-box optimization problem – see e.g. [10]).

In order to describe the mutation, let us first define the *exchange operation*: choose $i \neq j \in \{1, \dots, n\}$ uniformly and exchange $\pi(i)$ and $\pi(j)$. Let Mut_x be the mutation operator that consists of $k+1$ independent exchange operations, where k is a Poisson distributed random variable with $\lambda = 1$. The positions of exchange operations are assumed to be independent of all other preceding events in the algorithm, besides that, all realizations of the random variable k are mutually independent. The following tight bound is known.

Theorem 4 [19] *For the (1+1)-EA using mutation operator Mut_x and $\Phi(x) \equiv HAM(x)$, $t_n^{(1+1)} = \Theta(n^2 \ln n)$.*

It is easy to see from the proof in [19] that the same result applies to a simplified mutation operator that makes only one exchange operation each time it is called. Let us denote this operator by Mut_{1x} . An analogous lower bound holds for all mutation-based evolutionary algorithms using Mut_{1x} :

Proposition 5 For any EA with reproduction operator Mut_{1x} and $\Phi(x) \equiv \text{HAM}(x)$, $t_n^{EA} = \Omega(n^2 \ln n)$.

Proof. Note that for any $i = 0, \dots, n-1$ whenever individual x is such that $\Phi(x) = \text{HAM}(x) = i$, the probability of increasing its fitness is at least $(n-i)/n^2$. Indeed, there are $n-i$ elements at incorrect positions and if element k is at position $j \neq k$, then element j also occupies an incorrect position.

However this situation corresponds to an "underestimated" scenario where each improvement of fitness equals 1 and always there is only one improving position to exchange with position j . In fact the fitness increments by 1 only when there are two fitness-increasing ways to substitute the element at position j ; otherwise the k -th element equals j and the fitness increases by 2.

In case n is even, the "overestimated" scenario corresponds to an assumption that the probability of a fitness-increasing mutation equals to its upper bound $2(n-i)/(n(n-1))$ but the fitness always increments by 2. This scenario may be implemented with an "artificial" mutation operator $\text{Mut}'(x)$ which returns a solution of fitness $\Phi(x) + 2$, unless x is optimal, with probability $2(n-i)/(n(n-1))$; otherwise $\text{Mut}'(x)$ returns x . Operator $\text{Mut}'(x)$ dominates $\text{Mut}(x)$ and the expected optimization time is

$$n(n-1) \sum_{i=0}^{n/2-1} \frac{1}{2(n-2i)} = n(n-1)H(n/2)/4, \quad (6)$$

where $H(n) = \sum_{i=1}^n 1/i$ is the n -th harmonic number, $\ln(n) \leq H(n) \leq \ln(n) + 1$.

In case $n > 2$ is odd, some fitness improvement of $\text{Mut}'(x)$ should be odd as well, so we can assume that the improvement equals 3 (rather than 2) when $\Phi(x) = 0$. This yields the expected optimization time:

$$\frac{n(n-1)}{2} \left(\frac{1}{n} + \frac{1}{n-3} + \frac{1}{n-5} + \dots + \frac{1}{2} \right) = \frac{n(n-1)}{2} \left(\frac{1}{n} + \sum_{i=1}^{\lfloor n/2 \rfloor - 1} \frac{1}{2i} \right).$$

In general, for Mut' we have the expected (1+1)-EA optimization time $t_n^{(1+1)'} \geq n(n-1)H(\lfloor n/2 \rfloor - 1)/4$. Thus, $t_n^{(1+1)'} \geq \Omega(n^2 \ln n)$ but by Corollary 3 any EA based on Mut_{1x} has the hitting time for the optimum at least $t_n^{(1+1)'}$. Q.E.D.

Now let us turn to the mutation operator Mut_x .

Theorem 6 For any EA with mutation operator Mut_x and $\Phi(x) \equiv \text{HAM}(x)$, $t_n^{EA} = \Omega(n^2 \ln n)$.

Proof. Let us construct an artificial mutation operator Mut'' that works as $k+1$ iterations of the (1+1)-EA with the operator Mut' defined in the proof

of Proposition 5, and k is the same Poisson distributed random variable with $\lambda = 1$, as used in the original operator Mut_x (the source of randomness is the same). Let t be the random number of calls to Mut'' , made in the (1+1)-EA with mutation Mut'' until it finds the optimum. This (1+1)-EA acts like the (1+1)-EA with the operator Mut' and the number of calls to Mut' is equal to $t_1 = \sum_{i=1}^t (k_i + 1)$, where k_i is the i -th realization of the random variable k . According to the Wald's identity (see e.g. [8]), $E[t_1] = 2E[t]$ since $E[k_i + 1] = 2$. By Theorem 4 we have $E[t_1] = \Omega(n^2 \ln n)$, so the same holds for $E[t]$.

To complete the proof we only need to show that Mut'' dominates Mut_x . Indeed, it was shown that Mut' dominates Mut_{1x} . The operator Mut_x can be interpreted as a simple evolutionary algorithm, and operator Mut'' can be considered as the (1+1)-EA, both working with the same operator Mut_{1x} for the same (Poisson distributed) number of steps k , so the domination follows by Theorem 1. Q.E.D.

4 Conclusion

We have compared the (1+1)-EA to other EAs and proved that under the domination condition it is an optimal search technique with respect to probability of finding the solutions of sufficient quality after a given number of iterations. In the case of domination, the (1+1)-EA is also preferable with respect to the expected fitness at any iteration and the expected optimization time. As shown in Section 3, in some cases these results allow to extend the lower bounds on expected optimization time of the (1+1)-EA to a wider class of EAs.

Acknowledgements

The authors thank Sergey Klokov for very helpful discussions.

References

- [1] D. Aldous, U.U. Vazirani, "Go with the winners" algorithms, Proceedings of FOCS 1994, 1994, 492–501.
- [2] D.V. Arnold, H.-G. Beyer, Performance analysis of evolution strategies with multi-recombination in high-dimensional \mathbf{R}^n - search spaces disturbed by noise, Theor. Comp. Sci. **289** (2001) 629–647.
- [3] E. Balas, A sharp bound on the ratio between optimal integer and fractional covers, Math. of Operat. Res. **9** (1984) 1–5.

- [4] J. E. Beasley, P. C. Chu, A genetic algorithm for the set covering problem, *Europ. Journ. of Operat. Res.* **94** (2) (1996) 394–404.
- [5] H.-G. Beyer, *The Theory of Evolution Strategies*, Natural computing series, Springer, Heidelberg, 2001.
- [6] P.A. Borisovsky, A.V. Eremeev, A study on performance of the (1+1)-evolutionary algorithm, in: K. De Jong, R. Poli, and J. Rowe (Eds.), *Foundations of Genetic Algorithms*, 7, Morgan Kaufmann, San Francisco, (2003), pp. 271–287.
- [7] P.A. Borisovsky, M.S. Zavolovskaya, Experimental comparison of two evolutionary algorithms for the independent set problem, in: S. Cagnoni et al. (Eds.) *Applications of evolutionary computing: Proc. of EvoWorkshops 2003*, Springer Verlag, LNCS **2611**, (2003), pp. 154–164.
- [8] A.A. Borovkov, *Probability theory*, Gordon and Breach, 1998.
- [9] D.J. Daley, Stochastically monotone Markov chains. *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete.* **10** (1968) 307–317.
- [10] S. Droste, T. Jansen, K. Tinnefeld, I. Wegener, A new framework for the valuation of algorithms for black-box optimization, in K. De Jong, R. Poli, and J. Rowe, (Eds.) *Foundations of Genetic Algorithms*, 7, Morgan Kaufmann, San Francisco, (2003), pp. 253–270.
- [11] A.V. Eremeev, Modeling and analysis of genetic algorithm with tournament selection, in C. Fonlupt et al (Eds.), *Proc. of Artificial Evolution Conference (AE'99)*, LNCS **1829**, Springer Verlag, (2000), pp. 84–95.
- [12] J. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, 1975.
- [13] T. Jansen, I. Wegener, On the utility of populations in evolutionary algorithms, in *Proc. of Genetic and Evolutionary Computation Conference (GECCO)*, (2001), pp. 1034–1041.
- [14] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, *Science.* **220** (1983) 671–680.
- [15] J.R. Koza, *Genetic programming: On the programming of computers by means of natural selection*, MIT Press, 1992.
- [16] T. Lindvall, *Lectures on the coupling method*, Wiley, New York, 1992.
- [17] I. Rechenberg, *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Formann-Holzboog Verlag, Stuttgart, 1973.
- [18] G. Rudolph, Finite Markov chain results in evolutionary computation: A tour d'horizon, *Fundamenta Informaticae.* **35** (1-4) (1998) 67–89.
- [19] J. Scharnow, K. Tinnefeld, I. Wegener, The analysis of evolutionary algorithms on sorting and shortest paths problems, *Journ. Math. Mod. and Alg.* **3** (2004) 349–366.