ISSN 1990-4789, Journal of Applied and Industrial Mathematics, 2012, Vol. 6, No. 3, pp. 286–294. © Pleiades Publishing, Ltd., 2012.
Original Russian Text © A.V. Eremeev, 2012, published in Diskretnyi Analiz i Issledovanie Operatsii, 2012, Vol. 19, No. 2, pp. 41–54.

A Genetic Algorithm with Tournament Selection as a Local Search Method

A. V. Eremeev*

Omsk Department of Sobolev Institute of Mathematics, ul. Pevtsova 13, Omsk, 644099 Russia

Received June 18, 2011; in final form, August 2, 2011

Abstract—Some sufficient conditions are found under which the generational genetic algorithm with tournament selection first visits a local optimum in polynomially bounded time on average. These conditions are satisfied on a class of problems with guaranteed local optima if the appropriate parameters of the algorithm are chosen.

DOI: 10.1134/S1990478912030039

Keywords: *genetic algorithm, local search, approximate solution*

INTRODUCTION

A genetic algorithm (GA) was suggested by J. Holland [15] and represents a randomized heuristic search for an extremum which bases on an analogy with the genetic mechanisms in wildlife and using some evolving population of the sample solutions. Various modifications have been widely used in operations research, pattern recognition, artificial intelligence, and other areas [2, 6, 16]. Despite numerous experimental studies of these algorithms, the theoretical analysis of their efficiency is currently at an early stage [13].

In this paper, the genetic algorithms are studied in relation to finding the local optima of the combinatorial optimization problems, namely, the NP-optimization problems [4, 9]. Some particular attention is given to the situations when a GA is polynomially bounded on average time of searching a local optimum. Here and below, we call a value *polynomially bounded* if there exists a function polynomial in the length of input data which is an upper bound for this value.

The motivation of this study is the fact that a GA is often referred to as a class of local search methods (e.g., see [5, 8]). Therefore, it is of interest to study in detail the cases when the performance of a GA is explained by the similarity of its behavior to local search.

The article is structured as follows: Section 1 contains the definition of the class of NP-optimization problems and some related basic notions along with the description of the GA under study. In Section 2, an estimate of the average time of the first reach of a local optimum is obtained. In Section 3, using the estimate, we study the average time of finding an optimum for two special families of problems. In Section 4, the estimate from Section 2 is applied to the class of problems with guaranteed local optima. Section 5 contains some concluding remarks.

1. STATEMENT OF THE PROBLEM AND DESCRIPTION OF THE ALGORITHM

1.1. NP-Optimization Problems

The definition of NP-optimization problem [4, 9] formalizes the notion of the combinatorial or enumeration problem. Let $\{0,1\}^*$ denote the set of strings of arbitrary length consisting of 0s and 1s, let $\mathbb{N}$ be the set of natural numbers, and let $|S|$ denote the length of a string $S \in \{0,1\}^*$.

Definition 1. An *NP-optimization problem* is a triple $\Pi = (I, \mathrm{Sol}, f_x)$, where $I \subseteq \{0,1\}^*$ is a set of instances from Π and the following hold:

*E-mail: eremeev@ofim.oscsbras.ru

(i) there exists a deterministic Turing machine that recognizes in polynomially bounded time wether a string x of the initial data belongs to I;

(ii) $\mathrm{Sol}(x) \subseteq \{0,1\}^{n(x)}$ is the set of feasible solutions of an instance $x \in I$; and, for some polynomial poly, the dimension of the solution space is bounded as $n(x) \le \mathrm{poly}(|x|)$; moreover, for the input x and the string $\mathbf{x} \in \{0,1\}^*$, it is possible to determine the membership $\mathbf{x} \in \mathrm{Sol}(x)$ in polynomially bounded time;

(iii) for $x \in I$, the objective function $f_x : \mathrm{Sol}(x) \to \mathbb{N}$ which is subject to maximization (for NP-maximization problem Π) or minimization (for NP-maximization problem Π) is computable in polynomially bounded time.

Definition 2. An NP-optimization problem is called *polynomially bounded* if there exists a polynomial of $|x|$ that bounds the values $f_x(\mathbf{x})$ for $\mathbf{x} \in \mathrm{Sol}(x)$.

An approximation algorithm for an NP-minimization (NP-maximization) problem has approximation ratio $\rho \ge 1$ if it finds a solution for which the value of the objective function is at most ρ times greater than the optimal value (at most ρ times less than the optimal value) for each solvable instance.

1.2. Neighborhoods and Local Optima

Assume that, for every element $\mathbf{y} \in \mathrm{Sol}(x)$, a neighborhood $\mathcal{N}_x(\mathbf{y}) \subseteq \mathrm{Sol}(x)$ is defined. The family $\{\mathcal{N}_x(\mathbf{y}) : \mathbf{y} \in \mathrm{Sol}(x)\}$ is called a *neighborhood system* [9]. Here and below we assume that the mapping $\mathcal{N}_x$ is computable in polynomially bounded time.

Definition 3. If, for some $\mathbf{x} \in \mathrm{Sol}(x)$ and every $\mathbf{y} \in \mathcal{N}_x(\mathbf{x})$, the inequality $f_x(\mathbf{y}) \le f_x(\mathbf{x})$ holds for the maximization problem or $f_x(\mathbf{y}) \ge f_x(\mathbf{x})$ holds for minimization problem then $\mathbf{x}$ is called a *local optimum in the neighborhood system* $\mathcal{N}_x$.

Assume that $D(\cdot,\cdot)$ is some metric defined on the elements from $\mathrm{Sol}(x)$. The set

$$\mathcal{N}_x(\mathbf{x}) = \{\mathbf{y} : D(\mathbf{x},\mathbf{y}) \le k\}, \qquad \mathbf{x} \in \mathrm{Sol}(x),$$

is called the *neighborhood system of radius k generated by $D(\cdot,\cdot)$*.

The local search algorithm starts with some feasible solution. Then, at each iteration, the algorithm moves from the current solution to a new feasible solution in its neighborhood that has a better value of the objective function as compared to the current. The process continues until a local optimum is reached. The specific of a particular local search algorithm consists in the method of choosing a new solution in the neighborhood of the current solution.

Since it is always clear from the context which instance x is considered, we will omit the symbol x for short.

1.3. A Genetic Algorithm

A GA was suggested in [15] to be as an algorithm that simulates adaptation of a population to the environment and uses a function of fitness of individuals. Subsequently, a GA has been actively used also as an optimization method, where the fitness function is defined by the objective function of the problem.

Following the generally accepted approach, we assume that $\mathbf{x} \in \mathrm{Sol}$ and the fitness function has the form $\Phi(\mathbf{x}) = \varphi(f(\mathbf{x}))$, where φ is some strictly increasing function for the maximization problem or strictly decreases for the minimization problem. If $\mathbf{x} \notin \mathrm{Sol}$ then $\Phi(\mathbf{x})$ takes a value which is less than that of any feasible solution; and this corresponds to a penalty for the violation of the constraints of the problem.

Execution of a GA represents a consequent change of populations (generations) that consist of N individuals. Here and below, by an *individual* we assume an element $\mathbf{x}$ of the solution space $B = \{0,1\}^n$. An individual with greater fitness has more chances to produce descendants in the next generation. The individuals of each next generation are constructed from the individuals of the current population which are randomly affected by the operators of selection $\mathrm{Sel} : B^N \to \{1,\dots,N\}$, mutation $\mathrm{Mut} : B \to B$, and crossing-over $\mathrm{Cross} : B \times B \to B \times B$. These operators can be generally considered as some randomized procedures computable in polynomially bounded time [3]. Given the

argument of such a procedure, the probability distribution of its output does not depend on the previous stages of the algorithm.

Denote the generation $t \geq 0$ of the population by $X^t = (\mathbf{x}^{1t}, \ldots, \mathbf{x}^{Nt})$. The enumeration method for the individuals in population does not matter. An *iteration* of a GA is a transition from X^t to X^{t+1}. For ease of description of the algorithm we assume that N is even. Let us describe a scheme of the generational genetic algorithm with tournament selection (e.g., see [16]). The genetic algorithm which corresponds to this scheme will be referred to simply as GA.

Genetic algorithm GA

Step 1. Put $t := 0$.

Step 2. For k from 1 to N, do:

Step 2.1. Randomly construct an individual $\mathbf{x}^{k,0}$.

Iteration t:

Step 3. For k from 1 to $N/2$, do Steps 3.1–3.3.

Step 3.1. Selection: choose the individuals $\mathbf{x} := \mathbf{x}^{\mathrm{Sel}(X^t),t}$ and $\mathbf{y} := \mathbf{x}^{\mathrm{Sel}(X^t),t}$.

Step 3.2. Crossing-over: construct $(\mathbf{x}', \mathbf{y}') := \mathrm{Cross}(\mathbf{x}, \mathbf{y})$.

Step 3.3. Mutation: put $\mathbf{x}^{2k-1,t+1} := \mathrm{Mut}(\mathbf{x}')$ and $\mathbf{x}^{2k,t+1} := \mathrm{Mut}(\mathbf{y}')$.

Step 4. Put $t := t + 1$.

Step 5. If the stopping condition is not satisfied then go to Step 3 else go to Step 6.

Step 6. The result of the GA is the best found individual at all iterations.

At Step 2, the initial population X^0 is formed whose elements are generated by some deterministic or randomized procedure, e.g., according to the uniform distribution on B.

The selection operator has the same sense as the natural selection in the wildlife. The action of this is selection of the index of the parent individual for constructing the next descendant. In this paper, we study a widely used tournament selection operator [14]. This operator extracts s individuals from the population with the uniform distribution and chooses the one with the best fitness as a parent. The parameter s is called the *tournament size*.

The sizes of the population N and the tournament s, generally speaking, depend on the instance x; and the choice of these parameters can significantly influence the speed of the population convergence to the solutions of acceptable quality (e.g., see [10, 12]).

The stopping condition (Step 5) can be formulated in different ways: e.g., on reaching some given fitness, on executing a given number of iterations, or on passing some given number of iterations without any improvement of the best solution. For the ease of analysis we assume that the stopping condition is never satisfied.

We assume that, with the probability at least ε, $0 < \varepsilon \leq 1$, the individuals $(\mathbf{x}', \mathbf{y}') = \mathrm{Cross}(\mathbf{x}, \mathbf{y})$ appear in result of crossover such that at least one of them has a fitness not worse than that of the parent individuals $\mathbf{x}, \mathbf{y} \in B$; i.e.,

$$\mathbf{P}\{\max\{\Phi(\mathbf{x}'), \Phi(\mathbf{y}')\} \geq \max\{\Phi(\mathbf{x}), \Phi(\mathbf{y})\}\} \geq \varepsilon \tag{1}$$

for all $\mathbf{x}, \mathbf{y} \in B$ and, moreover, ε does not depend on x.

1.4. Examples of Mutation and Crossover Operators

Of the greatest interest are several versions of the mutation and crossover that are widely used in genetic algorithms and model the recombination and mutation processes in nature. We consider the most famous operators of mutation Mut* and crossing-over Cross* of the classic genetic algorithm [15].

The result of crossing-over $(\mathbf{x}', \mathbf{y}') = \text{Cross}^*(\mathbf{x}, \mathbf{y})$ acting on the parent solutions $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ is formed with probability P_c as

$$\mathbf{x}' = (x_1, \ldots, x_\chi, y_{\chi+1}, \ldots, y_n), \qquad \mathbf{y}' = (y_1, \ldots, y_\chi, x_{\chi+1}, \ldots, x_n),$$

where the random coordinate χ is chosen with uniform distribution from 1 to $n-1$. With the probability $1 - P_\text{c}$, the parent individuals are copied without changes; i.e., $\mathbf{x}' = \mathbf{x}$ and $\mathbf{y}' = \mathbf{y}$. This operator Cross* is usually called a *single-point crossover*. The condition (1) is satisfied for it with $\varepsilon = 1 - P_\text{c}$ if $P_\text{c} < 1$ is a constant that does not depend on x. Condition (1) is satisfied with $\varepsilon = 1$ if one of the two descendants is a solution of the problem of optimal recombination of parent solutions (e.g., see [10]).

The action of Mut* is a computing the individual $\mathbf{x}' = \text{Mut}^*(\mathbf{x})$, where each bit x'_i, $i = 1, \ldots, n$, independently with equal probability P_m is assigned the value $1 - x_i$, and with probability $1 - P_\text{m}$, the value x_i.

The level of crossing-over and mutation effects depends on the parameters P_c and P_m which, generally speaking, can depend on x. Increasing the mutation probability to 0.5 transforms the GA into a simple random enumeration. Decreasing P_m to zero leads to small diversity in the population and can lead to the "infinite loop" of GA when, at each iteration, only the previously generated individuals are produced.

2. AVERAGE TIME OF REACHING A LOCAL OPTIMUM

Assume that we have NP-minimization problem $\Pi = (I, \text{Sol}, f_x)$ and a family of neighborhoods $\mathcal{N}$. Put $h = |\{f(\mathbf{x}) : \mathbf{x} \in \text{Sol}\}| - 1$; i.e., h is a number of all nonoptimal values of the objective function f. Then, starting with an arbitrary solution, the local search reaches a local optimum in at most h iterations improving the value of the objective function. Assume that L is the minimal probability of reaching a solution inside a neighborhood:

$$L = \min_{\mathbf{x} \in \text{Sol},\, \mathbf{x}' \in \mathcal{N}(\mathbf{x})} \mathbf{P}\{\text{Mut}(\mathbf{x}) = \mathbf{x}'\}.$$

The greater the value of L, the bigger is the consistency of the mutation operator Mut with the neighborhood system $\mathcal{N}$. We consider the size of population N, the size of tournament s, and the value L as functions of the problem input data x. Assume that e is the Euler number.

Lemma 1. *If X^0 contains some feasible solution such that $s \geq rN$, $r > 0$, $h > 1$, $L > 0$, and*

$$N \geq \frac{2(1 + \ln h)}{L\varepsilon(1 - 1/e^{2r})} \tag{2}$$

then (i) *GA visits a local optimum by iteration h with the probability at least $1/e$ and* (ii) *local optimum is reached in at most eh iterations of GA on average.*

Proof. Note that, in the initial population, an individual with the highest fitness is a feasible solution since the fitness of the individuals that correspond to the infeasible solutions is always less than the fitness of the solutions from the feasible area. Assume that an event E_k^{t+1}, $k = 1, \ldots, N/2$, is a satisfaction of the following three conditions:

(a) a solution $\mathbf{x}_*^t$ with the biggest fitness in the population X^t is chosen from X^t while constructing the kth pair of descendants of the next generation;

(b) while constructing the kth pair of descendants by crossing-over, one of them has a fitness at least $\Phi(\mathbf{x}_*^t)$ (let it be $\mathbf{x}'$ for definiteness);

(c) the mutation operator applied to $\mathbf{x}'$ constructs a solution with the best fitness in the neighborhood $\mathcal{N}(\mathbf{x}')$; i.e.,

$$\Phi(\mathrm{Mut}(\mathbf{x}')) = \max_{\mathbf{y}\in\mathcal{N}(\mathbf{x}')} \Phi(\mathbf{y}).$$

Given the population X^t, let p denote the probability of occurrence of at least one event E_k^{t+1} for $k = 1, \ldots, N/2$. Let us find a bound $\lambda \le p$ independent of the choice of X^t. By the GA scheme

$$\mathbf{P}\{E_1^{t+1}\} = \cdots = \mathbf{P}\{E_{N/2}^{t+1}\},$$

we denote this probability by q. Since the events E_k^{t+1}, $k = 1, \ldots, N/2$, are independent for fixed X^t,

$$p \ge 1 - (1-q)^{N/2} \ge 1 - e^{-qN/2}.$$

Find the lower bound for q:

$$q \ge L\varepsilon(1 - (1 - 1/N)^{2s}).$$

However, $(1-1/N)^{2s} \le (1-1/N)^{2rN} \le 1/e^{2r}$; therefore,

$$q \ge L\varepsilon(1 - 1/e^{2r}) = Lc,$$

where $c = \varepsilon(1 - 1/e^{2r})$. Next, from the last inequality and from (2) it follows that

$$N \ge \frac{2}{L\varepsilon\,(1 - 1/e^{2r})} \ge 2/q. \tag{3}$$

Find the lower bound for p. At first, note that $1 - z/e \ge e^{-z}$ for any $z \in [0,1]$ and put $z = e^{-qN/2+1}$. Then, by (3) and $z \le 1$, we obtain

$$p \ge \exp\{-e^{1-qN/2}\} \ge \exp\{-e^{1-LcN/2}\}.$$

Next, we move from analysis of the descendants of the fixed population X^t to the random sequence of the populations $X^0, X^1, \ldots$. Note that λ^h is a lower bound for the probability of finding a local optimum in a series of at most h iterations that improve the record value of the objective function. Indeed, assume that $A_t = E_1^t + \cdots + E_{N/2}^t$, $t = 1, 2, \ldots$. Then

$$\mathbf{P}\{A_1 \& \ldots \& A_h\} = \mathbf{P}\{A_1\} \prod_{t=1}^{h-1} \mathbf{P}\{A_{t+1} \mid A_1 \& \ldots \& A_t\} \ge \lambda^h. \tag{4}$$

Thus, put $\lambda = \exp\{-e^{1-LcN/2}\}$. Again, using (2), we obtain the lower bound for the probability of finding a local optimum in a series of at most h iterations improving the best-found solution:

$$\lambda^h = \exp\{-he^{1-LcN/2}\} \ge \exp\{-he^{-\ln h}\} = 1/e.$$

The proof of (i) of the lemma is complete.

To estimate the average time of obtaining a local optimum we consider a sequence of series of h iterations each. Assume that the event D_i, $i = 1, 2, \ldots$, is the absence of a local optimum in the population in the ith series. Under the conditions of the lemma the probability of each D_i is at most $\mu = 1 - 1/e$ for every preceding execution of algorithm. Similarly to (4), we conclude

$$\mathbf{P}\{D_1 \& \ldots \& D_k\} \le \mu^k.$$

Thus, if we denote by η a random value equal to the number of the first series when the local optimum was obtained then, using the properties of the mathematical expectation (e.g., see [1]), we have

$$E[\eta] = \sum_{i=0}^{\infty} \mathbf{P}\{\eta > i\} = 1 + \sum_{i=1}^{\infty} \mathbf{P}\{D_1 \& \ldots \& D_i\} \le 1 + \sum_{i=1}^{\infty} \mu^i = e.$$

Hence, the local optimum is found in at most eh GA iterations on average.

The proof of Lemma 1 is complete. □

Denote the upper rounding by $\lceil\cdot\rceil$. Then under the conditions of Lemma 1 the choice

$$N = 2\left\lceil \frac{1+\ln h}{L\varepsilon(1-1/e^{2r})} \right\rceil, \qquad s = \lceil rN \rceil \tag{5}$$

assures obtaining a local optimum in GA in $O(h)$ iterations on average.

The execution times of the operators Mut and Cross are polynomially bounded, and the procedure of the tournament selection requires $O(s) = O(N)$ time. Hence, we obtain

Theorem 1. *If some Problem $\Pi = (I, \mathrm{Sol}, f_x)$ and a function $L^{-1}(x)$ are polynomially bounded and a population X^0 of GA always contains feasible solution then, for the sizes of population and tournament chosen according to* (5)*, GA first visits a local optimum in polynomially bounded time on average.*

Note that the assumption of polynomial computability of $\mathcal{N}$ in Section 1 does not guarantee that the power of the neighborhood is polynomially bounded; i.e., $|\mathcal{N}(\mathbf{x})|$ for all $\mathbf{x} \in \mathrm{Sol}$ is bounded with a polynomial in $|x|$. If $\mathcal{N}$ satisfies the last condition then it is called *polynomially bounded* [17]. For this $\mathcal{N}$, there exists a mutation operator $\mathrm{Mut}(\mathbf{x})$ computable in polynomially bounded time which sets the uniform distribution of the descendant individuals on the set $\mathcal{N}(\mathbf{x})$ for $\mathbf{x}$ given. Then $1/L$ is also upper bounded by some polynomial in $|x|$. Thus, Theorem 1 is applicable to many available families of neighborhoods for the NP-optimization problems.

Denote a Hamming distance between two binary strings $\mathbf{x}$ and $\mathbf{y}$ by $\delta(\mathbf{x}, \mathbf{y})$.

Definition 4. Assume that Π is an NP-optimization problem. The family of neighborhoods $\mathcal{N}$ is called *k-bounded* [9] if, for every $\mathbf{x} \in \mathrm{Sol}$ and $\mathbf{y} \in \mathcal{N}(\mathbf{x})$, the inequality $\delta(\mathbf{x}, \mathbf{y}) \le k$ holds with a constant k.

The mutation operator Mut^* of the classic genetic algorithm [15] constructs a string $\mathbf{y}$ when mutation is $\mathbf{x}$ with probability $P_{\mathrm{m}}^{\delta(\mathbf{x},\mathbf{y})}(1-P_{\mathrm{m}})^{n-\delta(\mathbf{x},\mathbf{y})}$. Note that the probability $P_{\mathrm{m}}^k(1-P_{\mathrm{m}})^{n-k}$ as a function of $P_{\mathrm{m}} \in [0,1]$ reaches its maximum at $P_{\mathrm{m}} = k/n$. The following gives a lower bound for the probability

$$\mathbf{P}\{\mathrm{Mut}^*(\mathbf{x}) = \mathbf{y}\} \quad \text{for all} \quad \mathbf{y} \in \mathcal{N}(\mathbf{x})$$

at optimal in the described sense choice of $P_{\mathrm{m}} = k/n$:

Proposition 1. *Assume that the family of neighborhoods $\mathcal{N}$ is k-bounded and $k \le n/2$. Then, for $P_{\mathrm{m}} = k/n$, $\mathbf{x} \in \mathrm{Sol}$, and $\mathbf{y} \in \mathcal{N}(\mathbf{x})$, we have*

$$\mathbf{P}\{\mathrm{Mut}^*(\mathbf{x}) = \mathbf{y}\} \ge \left(\frac{k}{en}\right)^k.$$

Proof. For $\mathbf{x} \in \mathrm{Sol}$ and $\mathbf{y} \in \mathcal{N}(\mathbf{x})$, we have

$$\mathbf{P}\{\mathrm{Mut}^*(\mathbf{x}) = \mathbf{y}\} = \left(\frac{k}{n}\right)^{\delta(\mathbf{x},\mathbf{y})}\left(1-\frac{k}{n}\right)^{n-\delta(\mathbf{x},\mathbf{y})} \ge \left(\frac{k}{n}\right)^k\left(1-\frac{k}{n}\right)^{n-k}$$

since $P_{\mathrm{m}} = k/n \le 1/2$. Further,

$$\frac{\partial(1-k/n)^{n-k}}{\partial n} < 0 \quad \text{for} \quad n > k;$$

and, moreover,

$$(1-k/n)^{n-k} \to 1/e^k \quad \text{as} \quad n \to \infty.$$

Hence, $(1-k/n)^{n-k} \ge 1/e^k$, which implies the claim. □

3. AN AVERAGE TIME OF OBTAINING AN OPTIMUM FOR TWO SPECIAL FAMILIES OF PROBLEMS

3.1. The Family of Problems ONEMAX**

As an example of application of Lemma 1 and Proposition 1 we consider the family of the single-extremal problems that is often used in the analysis of evolutionary algorithms [11].

Let us define the family ONEMAX* of the objective functions of the form $\delta(\mathbf{x}, a)$, where $a \in \{0,1\}^n$ is an optimal solution. Then ONEMAX** is defined as the family of functions $\varphi \circ g$, where $g \in$ ONEMAX*, $\varphi : \mathbb{Z}^+ \to \mathbb{R}$ is a strictly increasing function, $\mathbb{Z}^+$ is the set of nonnegative integers, and $\mathbb{R}$ is the set of real numbers. It is required to maximize a function $f \in$ ONEMAX** on $\mathrm{Sol} = B$. Note that, without the loss of generality, it suffices to consider φ with natural numbers. As a fitness function for $f \in$ ONEMAX** it is natural to choose $\Phi(\mathbf{x}) \equiv f(\mathbf{x})$.

In the neighborhood system of radius 1 generated by the Hamming metric, the point a is a unique local optimum; i.e., it is the global optimum. According to Proposition 1 for $P_\mathrm{m} = 1/n$, $n > 1$, for every $\mathbf{x} \in \mathrm{Sol}$ and $\mathbf{y} \in \mathcal{N}(\mathbf{x})$, we have

$$\mathbf{P}\{\mathrm{Mut}^*(\mathbf{x}) = \mathbf{y}\} \geq 1/(en).$$

Lemma 1 implies that if

$$s = rN, \qquad r > 0, \qquad N = \left\lceil \frac{2en(1+\ln n)}{\varepsilon(1-1/e^{2r})} \right\rceil$$

then the genetic algorithm first visits an optimum on average in at most en iterations. The time complexity of the tournament selection in this genetic algorithm is $O(N)$. If we assume that crossing-over, mutation and computing of the objective function can be done in certain time T then the GA first visits optimum on average in time $O(Tn^3 \ln^2 n)$. For example, in the classic genetic algorithm [15], $T = O(n)$; i.e., the optimum is first reached in time $O(n^4 \ln^2 n)$ on average.

3.2. The Family of Graph $G(\ell)$ Vertex Cover Problems

A Minimum Vertex Cover Problem (VCP) can be stated as follows:

Let $G = (V, E)$ be a graph with the sets of vertices V and the set of edges E. A subset $C \subseteq V$ is called a vertex cover if each edge from E is incident to at least one vertex from C. It is required to find a vertex cover of minimal power.

Consider the family of VCPs of a particular form, where the graph $G(\ell)$ consists of ℓ three-vertex complete subgraphs that are pairwise disconnected, $\ell = 1, 2, \ldots$. Obviously, each of the triangular subgraph is optimally covered with two vertices and nonoptimally, with three vertices. Despite the simplicity of this problem, it is shown in [7] that some known algorithms, such as branch-and-bound, have exponentially increasing time complexity with respect to the number of graph vertices.

Assume that the vertices and edges of the graph are enumerated, and, to each edge e_i from E, one bit $\mathbf{x}_i$ is assigned in the encoding of the solutions from Sol, $i = 1, \ldots, n$, and $n = |E|$. Assume that, for each e_i, the zero value of the bit $\mathbf{x}_i$ means that the vertex incident to e_i with a less index number is included in $C(\mathbf{x})$ and the value $\mathbf{x}_i = 1$ means that the vertex with a bigger number is included. Obviously, with this approach of representing solutions, each string $\mathbf{x} \in B$ encodes some feasible solution; i.e., $C(\mathbf{x})$ is a cover.

As a fitness function for VCP, we can choose a strictly decreasing function of $|C(\mathbf{x})|$. Assume that $\Phi(\mathbf{x}) = |V| - |C(\mathbf{x})|$. In this case, for the graph $G(\ell)$, the value $\Phi(\mathbf{x})$ coincides with the number of connected components that are optimally covered with vertices $C(\mathbf{x})$.

In the neighborhood system with radius 1 generated by Hamming metric, every local optimum is global. Similarly to the previous example, we obtain $\mathbf{P}\{\mathrm{Mut}^*(\mathbf{x}) = \mathbf{y}\} \geq 1/en$, where $n = 3\ell$. By Lemma 1, for $s = rN$, where

$$r > 0, \qquad N = \left\lceil \frac{6e\ell(1+\ln \ell)}{\varepsilon(1-1/e^{2r})} \right\rceil,$$

the genetic algorithm first visits an optimum in at most $e\ell$ iterations on average; i.e., the average time of its execution till finding an optimum is polynomially bounded.

4. ANALYSIS OF THE PROBLEMS WITH GUARANTEED LOCAL OPTIMA

Let us apply Theorem 1 to estimate the ability of a GA to find an approximate solution with the guaranteed approximation ratio.

Definition 5. Assume that Π is a polynomially bounded NP-optimization problem. Problem Π belongs to a class of the problems with *guaranteed local optima* (GLO) [9] if the following two conditions hold:

(i) for every input $x \in I$, at least one feasible solution $\mathbf{y}_x \in \mathrm{Sol}$ can be computed in polynomially bounded time;

(ii) there exists $k \in \mathbb{N}$ such that all local optima of Problem Π with respect to some k-bounded family of neighborhoods have a constant approximation ratio.

The examples of problems from the GLO class are the problems of the maximum satisfiability of a Boolean formula, or a maximum independent set, or a minimum dominating set, or a minimum vertex cover of a graph with degrees of vertices bounded with a constant, as well as the problem of a maximum graph cut [9].

If the NP-optimization problem Π lies in the class GLO then, by Proposition 1 for $k \le n/2$, for every $\mathbf{x} \in \mathrm{Sol}$ and $\mathbf{y} \in \mathcal{N}(\mathbf{x})$, the mutation operator Mut^* with $P_\mathrm{m} = k/n$ satisfies

$$\mathbf{P}\{\mathrm{Mut}^*(\mathbf{x}) = \mathbf{y}\} \ge 1/\mathrm{poly}(|x|),$$

where poly is some polynomial. The probability $\mathbf{P}\{\mathrm{Mut}^*(\mathbf{x}) = \mathbf{y}\}$ has a positive constant lower bound for $n/2 < k < n$. Thus, Theorem 1 implies

Corollary. *Assume that Π is a problem from the class* GLO, *$n > k$, and the population X^0 contains $\mathbf{y}_x$. Then the genetic algorithm with the mutation operator* Mut^* *for $P_\mathrm{m} = k/n$ and parameters N and s chosen according to* (5) *obtains a solution with a constant approximation ratio on average in polynomially bounded time.*

5. CONCLUSION

The main results establish some new properties of genetic algorithms from the perspective of local optimization. The obtained estimations of average time complexity and proposed values for the adjustable parameters provide the ideas on the trends of the genetic algorithms depending on the increase of the size of tournament and the size of the population.

ACKNOWLEDGMENTS

The author was supported by the Russian Foundation for Basic Research (project no. 10–01–00598) and the Presidium of the Russian Academy of Sciences (Program no. 2 of Basic Research, project no. 227).

REFERENCES

1. B. V. Gnedenko, *Theory of Probability* (Nauka, Moscow, 1988; Gordon and Breach,1997).
2. N. G. Zagoruiko, "Self-Learning Genetic Prediction Algorithm (GAP)," in *Artificial Intelligence and Expert Systems*, Vol. 160 (Vychisl. Sistemy, Novosibirsk, 1997), pp. 3–17.
3. A. Kitaev, A. Shen', and M. Vyalyi, *Classical and Quantum Computing* (MTsNMO, Moscow, 1999) [in Russian].
4. Yu. A. Kochetov, "Computational Bounds for Local Search in Combinatorial Optimization," Zh. Vychisl. Mat. i Mat. Fiz. **48** (5), 788–807 (2008) [Comput. Math. Math. Phys. **48** (5), 747–763 (2008)].
5. Yu. A. Kochetov, "Probabilistic Approaches of Local Search for Problems of Discrete Optimization," in *Discrete Mathematics and Its Applications*: *A Collection of Lectures of Youth Schools of Thought on Discrete Mathematics and Its Applications* (Izd. Tsentra Prikl. Issled. Moskov. Gos. Univ., Moscow, 2001), pp. 84–117.

6. D. Rutkovskaya, M. Pilin'skii, and L. Rutkovskii, *Neural Networks, Genetic Algorithms, and Fuzzy Systems* (Goryachaya Liniya, Moscow, 2006) [in Russian].
7. L. A. Saiko, "Studying the Power of L-Foldings of Some Problems of Covering," in *Discrete Optimization and Analysis of Complex Systems*: *Proceedings* (Vychisl. Tsentr, Novosibirsk, 1989), pp. 76–97.
8. E. H. L. Aarts and J. K. Lenstra, *Introduction. Local Search in Combinatorial Optimization* (Wiley, New York, 1997), pp. 1–19.
9. G. Ausiello and M. Protasi, "Local Search, Reducibility, and Approximability of NP-Optimization Problems," Inform. Process. Lett. **54**,73–79 (1995).
10. E. Balas and W. Niehaus, "Optimized Crossover-Based Genetic Algorithms for the Maximum Cardinality and Maximum Weight Clique Problems," J. Heurist. **4** (2), 107–122 (1998).
11. S. Droste, T. Jansen, and I. Wegener, "Upper and Lower Bounds for Randomized Search Heuristics in Black-Box Optimization," Theory Comput. Syst. **39** (4), 525–544 (2006).
12. A. V. Eremeev, "Modeling and Analysis of Genetic Algorithm with Tournament Selection," in *Proceedings of the Artificial Evolution Conference (Dunkerque, France, November 3–5, 1999)* (Springer, Berlin, 2000), pp. 84–95.
13. A. V. Eremeev and C. R. Reeves, "Evolutionary Algorithms in Discrete Optimization," in *Discrete Optimization and Operations Research Conference (Novosibirsk, June 24–28, 2002). Abstracts* (Inst. Mat., Novosibirsk, 2000), pp. 40–45.
14. D. E. Goldberg, "A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing," Complex Syst. **4**, 445–460 (1990).
15. J. Holland, *Adaptation in Natural and Artificial Systems* (Univ. Michigan, Ann Arbor, 1975).
16. C. R. Reeves, "Genetic Algorithms for the Operations Researcher," INFORMS J. Comput. **9** (3), 231–250 (1997).
17. V. Rödl and C. Tovey, "Multiple Optima in Local Search," J. Algorithms **8**, 250–259 (1987).