A Genetic Algorithm
with Expansion and Exploration Operators
for the Maximum Satisfiability Problem

Anna Gorbenko

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
gorbenko.ann@gmail.com

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
Vladimir.Popov@usu.ru

Abstract

There are many problems that standard genetic algorithms fail to solve. Refinements of standard genetic algorithms that can be used to solve hard problems has caused considerable interest. In this paper, we consider genetic algorithms with expansion and exploration operators for the maximum satisfiability problem.

The maximum satisfiability problem (MAX SAT) is the problem of determining the maximum number of clauses of a given Boolean formula in conjunctive normal form (CNF) that can be satisfied by some assignment. The problem MAX SAT remains **NP**-hard even if all expressions are written in conjunctive normal form with k variables per clause (k-CNF), for any $k \geq 2$ (see e.g. [1]). The problem MAX k-SAT is the problem of determining the maximum number of clauses of a given k-CNF that can be satisfied by some assignment. It should be noted that investigation of algorithms for MAX

SAT received a lot of attention (see e.g. [2]). In this paper, we consider genetic algorithms with expansion and exploration operators to try to solve the maximum satisfiability problem.

Genetic algorithms are used to solve many different hard problems. In particular, we can mention such problems as the set of parameterized k-covers problem [3], different problems of sensor placement (see e.g. [4] – [9]), the binary paint shop problem [10], various problems of bioinformatics (see e.g. [11] – [20]), technical vision (see e.g. [21] – [25]), robot self-awareness (see e.g. [26] – [30]), robot anticipation (see e.g. [31] – [34]), different planning problems (see e.g. [35] – [39]), graph problems (see e.g. [40] – [43]), visual landmarks problems (see e.g. [44] – [47]), etc. But, it is well known that there are many problems that standard genetic algorithms fail to solve.

One of a number of ideas of refinement of standard genetic algorithms was proposed in [48]. The idea is to preserve good building blocks found by the genetic algorithm. For this purpose, we can use some constraints on the choice of recombination. In particular, an expansion operator can be used [48]. Also, we can use exploration operator as a refinement of mutation [48].

Expansion and exploration operators were proposed in [48] for the shortest common superstring problem. In particular, an expansion operator is an addition of another block to increase the individual's genome length by one block. The added block is selected by expansion operator in such a way that the genome will still be a subsequence of at least one solution. Exploration operator acts as a mutative force within the building blocks population. This operator injects noise and thus promotes exploration of the huge building blocks search space. An individual may be removed and re-initialized as a new individual.

Let

$$f(z[1], z[2], \ldots, z[m])$$

be a Boolean function. Let

$$W = \{u[1], u[2], \ldots, u[n]\},$$

$$u[i] \in \{0, 1\}^+,$$

$$1 \le i \le n,$$

be a population of chromosomes. We assume that

$$u[i] = u[i, 1]u[i, 2] \ldots u[i, m],$$

$$u[i, j] \in \{0, 1\},$$

$$1 \le i \le n,$$

$$1 \le j \le m.$$

We can consider $u[i]$ as a solution for

$$f(z[1], z[2], \ldots, z[m]).$$

In particular, we assume that $z[j] = u[i, j]$.

We consider only CNFs. So,

$$f = \wedge_{j=1}^{k} C[j](z[1], \ldots, z[m])$$

where

$$C[j](z[1], z[2], \ldots, z[m])$$

is a clause. We say that $u[i, j]$ is a true assignment for

$$C[k](z[1], z[2], \ldots, z[m])$$

if $u[i, j] \in \{0, 1\}$ and $z[j] = u[i, j]$ evaluates

$$C[k](z[1], z[2], \ldots, z[m])$$

to true. Let $C(u[i, j])$ be the set of clauses such that

$$C[k](z[1], z[2], \ldots, z[m]) \in C(u[i, j])$$

if and only if $u[i, j]$ is a true assignment for

$$C[k](z[1], z[2], \ldots, z[m]).$$

Now, we consider a standard genetic algorithm (SGA) for the satisfiability problem. A proportion $\mathcal{P}$ of the existing population is selected to breed a new generation during each successive generation. We assume that

$$\mathcal{P} = \lceil \frac{n}{2} \rceil.$$

Individual chromosomes are selected by a fitness function $\mathcal{F}$. To generate a second generation population of chromosomes, we can use two genetic operators: crossover $\mathcal{C}$ and mutation $\mathcal{M}$.

Usually, crossover defines a part of parent chromosome which used for construction of child chromosome. As $\mathcal{C}$ we use standard random operator. If $u[i]$ and $u[j]$ are two parent chromosomes, then we obtain two child chromosomes:

$$\mathcal{C}(u[i], u[j]) = u[i, 1] \ldots u[i, \mathcal{C}(u[i])]u[j, \mathcal{C}(u[i]) + 1] \ldots u[j, m],$$

$$\mathcal{C}(u[j], u[i]) = u[j, 1] \ldots u[j, \mathcal{C}(u[j])]u[i, \mathcal{C}(u[j]) + 1] \ldots u[i, m].$$

For

$$u[i[1]], u[i[2]], \ldots, u[i[p]],$$

we consider

$$\mathcal{C}(u[i[1]], u[i[2]]), \mathcal{C}(u[i[2]], u[i[1]]), \ldots,$$

$$\mathcal{C}(u[i[p-1]], u[i[p]]), \mathcal{C}(u[i[p]], u[i[p-1]])$$

if $p = 2q$,

$$u[i[1]], \mathcal{C}(u[i[2]], u[i[3]]), \mathcal{C}(u[i[3]], u[i[2]]), \ldots,$$

$$\mathcal{C}(u[i[p-1]], u[i[p]]), \mathcal{C}(u[i[p]], u[i[p-1]])$$

if $p = 2q + 1$. This generational process is repeated until a termination condition $\mathcal{T}$ has been reached.

As $\mathcal{T}$ we consider time function. We assume that

$$\mathcal{F}(u[i]) = |\cup_{j=1}^{m} C(u[i,j])|.$$

Let $\mathcal{M}$ be a random function which with small probability changes values of $u[i,j]$.

Now, we consider a genetic algorithm with expansion and exploration operators (GAEE). We use the following expansion operator instead $\mathcal{C}$. Let

$$\mathcal{E}_1(u[i], u[j]) = v[1] \ldots v[m]$$

where

$$v[k] = \begin{cases} u[i,k], |C(u[i,k])| > 0, \\ u[j,k], |C(u[i,k])| < 1. \end{cases}$$

Also, we use exploration operator $\mathcal{E}_2$ instead $\mathcal{M}$. If r is a number of generation

$$W_r = \{u_r[1], u_r[2], \ldots, u_r[n]\},$$

then

$$F_r = \sum_{i=1}^{n} \mathcal{F}(u_r[i]).$$

For any $r > 1$ and $i < n$, we assume that

$$\mathcal{F}(u_r[i]) \geq \mathcal{F}(u_r[i+1]).$$

Let $D_H(X, Y)$ be the Hamming distance between strings X and Y. Let

$$h_r = \sum_{i=1}^{n} D_H(u_r[i], u_{r-1}[i]),$$

$$H_r = \sum_{1 \leq i < j \leq n} D_H(u_r[i], u_r[j]).$$

Operator $\mathcal{E}_2$ used only if

$$\alpha(F_r, h_r, H_r) < 1$$

where $\alpha(F_r, h_r, H_r)$ is a polynomial function which is predicted by a genetic algorithm. Operator $\mathcal{E}_2$ removes chromosome $u_r[i]$ only if $u_r[i]$ has high fitness value and there is $u_{r-1}[j]$ such that

$$D_H(u_r[i], u_{r-1}[j]) < m^{\frac{1}{4}}.$$

Operator $\mathcal{E}_2$ replaces $u_r[i]$ by new random chromosome v such that

$$D_H(u_r[i], v) > \frac{m}{2}.$$

Operator $\mathcal{E}_2$ replaces no more than $\frac{n}{6}$ chromosomes.

Selected experimental results are given in Table 1.

average number of generations	10^3	10^4	10^5	10^6
average number of true clauses for SGA	55 %	59 %	64 %	68 %
average number of true clauses for GAEE	53 %	61 %	75 %	82 %

Table 1: Experimental results for SGA and GAEE.

References

[1] C.H. Papadimitriou, Computational complexity, Addison Wesley, Reading, 1994.

[2] I. Bliznets and A. Golovnev, A New Algorithm for Parameterized MAX-SAT, *Lecture Notes in Computer Science*, 7535 (2012) 37-48.

[3] A. Gorbenko and V. Popov, The set of parameterized k-covers problem, *Theoretical Computer Science*, 423 (2012) 19-24.

[4] A. Gorbenko, M. Mornev, V. Popov, and A. Sheka, The Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012) 965-967.

[5] A. Gorbenko, M. Mornev, V. Popov, and A. Sheka, The problem of sensor placement for triangulation-based localisation, *International Journal of Automation and Control*, 5 (2011) 245-253.

[6] A. Gorbenko and V. Popov, SAT Solvers for the Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012) 1235-1238.

[7] A. Gorbenko and V. Popov, Clustering Algorithm in Mobile Ad Hoc Networks, *Advanced Studies in Theoretical Physics*, 6 (2012) 1239-1242.

[8] V. Popov, Partially Distinguishable Guards, *Applied Mathematical Sciences*, 6 (2012) 6587-6591.

[9] A. Gorbenko and V. Popov, On the Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012) 1117-1120.

[10] A. Gorbenko and V. Popov, The Binary Paint Shop Problem, *Applied Mathematical Sciences*, 6 (2012) 4733-4735.

[11] A. Gorbenko and V. Popov, The Longest Common Parameterized Subsequence Problem, *Applied Mathematical Sciences*, 6 (2012) 2851-2855.

[12] A. Gorbenko and V. Popov, On the Longest Common Subsequence Problem, *Applied Mathematical Sciences*, 6 (2012) 5781-5787.

[13] A. Gorbenko and V. Popov, Multiple Occurrences Shortest Common Superstring Problem, *Applied Mathematical Sciences*, 6 (2012) 6573-6576.

[14] V. Popov, Approximate Periods of Strings for Absolute Distances, *Applied Mathematical Sciences*, 6 (2012) 6713-6717.

[15] V. Popov, The Approximate Period Problem, *IAENG International Journal of Computer Science*, 36 (2009) 268-274.

[16] V. Popov, Multiple genome rearrangement by swaps and by element duplications, *Theoretical Computer Science* 385 (2007) 115-126.

[17] V. Yu. Popov, Computational complexity of problems related to DNA sequencing by hybridization, *Doklady Mathematics*, 72 (2005) 642-644.

[18] V. Popov, The approximate period problem for DNA alphabet, *Theoretical Computer Science*, 304 (2003) 443-447.

[19] A. Gorbenko and V. Popov, The Far From Most String Problem, *Applied Mathematical Sciences*, 6 (2012) 6719-6724.

[20] A. Gorbenko and V. Popov, The c-Fragment Longest Arc-Preserving Common Subsequence Problem, *IAENG International Journal of Computer Science*, 39 (2012) 231-238.

[21] A. Gorbenko and V. Popov, Self-Learning Algorithm for Visual Recognition and Object Categorization for Autonomous Mobile Robots, *Lecture Notes in Electrical Engineering*, 107 (2012), 1289-1295.

[22] A. Gorbenko and V. Popov, Face Detection and Visual Landmarks Approach to Monitoring of the Environment, *International Journal of Mathematical Analysis*, 7 (2013) 213-217.

[23] A. Gorbenko and V. Popov, On Face Detection from Compressed Video Streams, *Applied Mathematical Sciences*, 6 (2012) 4763-4766.

[24] A. Gorbenko and V. Popov, Usage of the Laplace Transform as a Basic Algorithm of Railroad Tracks Recognition, *International Journal of Mathematical Analysis*, 6 (2012) 2413-2417.

[25] A. Gorbenko, A. Lutov, M. Mornev, and V. Popov, Algebras of Stepping Motor Programs, *Applied Mathematical Sciences*, 5 (2011) 1679-1692.

[26] A. Gorbenko and V. Popov, Robot Self-Awareness: Usage of Co-training for Distance Functions for Sequences of Images, *Advanced Studies in Theoretical Physics*, 6 (2012) 1243-1246.

[27] A. Gorbenko and V. Popov, Robot Self-Awareness: Formulation of Hypotheses Based on the Discovered Regularities, *Applied Mathematical Sciences*, 6 (2012) 6583-6585.

[28] A. Gorbenko and V. Popov, Robot Self-Awareness: Occam's Razor for Fluents, *International Journal of Mathematical Analysis*, 6 (2012) 1453-1455.

[29] A. Gorbenko, V. Popov, and A. Sheka, Robot Self-Awareness: Exploration of Internal States, *Applied Mathematical Sciences*, 6 (2012) 675-688.

[30] A. Gorbenko, V. Popov, and A. Sheka, Robot Self-Awareness: Temporal Relation Based Data Mining, *Engineering Letters*, 19 (2011) 169-178.

[31] A. Gorbenko and V. Popov, Robot's Actions and Automatic Generation of Distance Functions for Sequences of Images, *Advanced Studies in Theoretical Physics*, 6 (2012) 1247-1251.

[32] A. Gorbenko and V. Popov, Anticipation in Simple Robot Navigation and Finding Regularities, *Applied Mathematical Sciences*, 6 (2012) 6577-6581.

[33] A. Gorbenko and V. Popov, Anticipation in Simple Robot Navigation and Learning of Effects of Robot's Actions and Changes of the Environment, *International Journal of Mathematical Analysis*, 6 (2012) 2747-2751.

[34] A. Gorbenko and V. Popov, The Force Law Design of Artificial Physics Optimization for Robot Anticipation of Motion, *Advanced Studies in Theoretical Physics*, 6 (2012) 625-628.

[35] A. Gorbenko and V. Popov, Footstep Planning for Humanoid Robots, *Applied Mathematical Sciences*, 6 (2012) 6567-6571.

[36] A. Gorbenko and V. Popov, Multi-agent Path Planning, *Applied Mathematical Sciences*, 6 (2012) 6733-6737.

[37] A. Gorbenko and V. Popov, Task-resource Scheduling Problem, *International Journal of Automation and Computing*, 9 (2012) 429-441.

[38] A. Gorbenko and V. Popov, Programming for Modular Reconfigurable Robots, *Programming and Computer Software*, 38 (2012) 13-23.

[39] A. Gorbenko, M. Mornev, and V. Popov, Planning a Typical Working Day for Indoor Service Robots, *IAENG International Journal of Computer Science*, 38 (2011) 176-182.

[40] A. Gorbenko, V. Popov, and A. Sheka, Localization on Discrete Grid Graphs, *Lecture Notes in Electrical Engineering*, 107 (2012), 971-978.

[41] A. Gorbenko and V. Popov, The Hamiltonian Alternating Path Problem, *IAENG International Journal of Applied Mathematics*, 42 (2012) 204-213.

[42] A. Gorbenko and V. Popov, The Problem of Finding Two Edge-Disjoint Hamiltonian Cycles, *Applied Mathematical Sciences*, 6 (2012) 6563-6566.

[43] A. Gorbenko and V. Popov, Hamiltonian Alternating Cycles with Fixed Number of Color Appearances, *Applied Mathematical Sciences*, 6 (2012) 6729-6731.

[44] A. Gorbenko and V. Popov, Computational Experiments for the Problem of Selection of a Minimal Set of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012) 5775-5780.

[45] A. Gorbenko and V. Popov, The Problem of Selection of a Minimal Set of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012) 4729-4732.

[46] A. Gorbenko and V. Popov, A Real-World Experiments Setup for Investigations of the Problem of Visual Landmarks Selection for Mobile Robots, *Applied Mathematical Sciences*, 6 (2012) 4767-4771.

[47] A. Gorbenko and V. Popov, On the Problem of Placement of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012) 689-696.

[48] A. Zaritsky and M. Sipper, The preservation of favoured building blocks in the struggle for fitness: The puzzle algorithm, *IEEE Transactions on Evolutionary Computation*, 8 (2004), 443-455.