Comparison of application of various crossovers in solving inhomogeneous minimax problem modified by Goldberg model

B V Kobak¹, A G Zhukovskiy² and A P Kuzin¹

¹ Don State Technical University, 1Gagarin Square 1, Rostov-on-Don, 344000, Russia
² North-Caucasian branch of Moscow technical university of communications and informatics, 62 b, Serafimovicha Street, Rostov-on-Don, 344002, Russia

Email: valera33305@mail.ru, zhykovskij@mail.ru, alexpavkuzin@gmail.com

Abstract. This paper considers one of the classical NP complete problems - an inhomogeneous minimax problem. When solving such large-scale problem, there appear difficulties in obtaining an exact solution. Therefore, let us propose getting an optimum solution in an acceptable time. Among a wide range of genetic algorithm models, let us choose the modified Goldberg model, which earlier was successfully used by authors in solving NP complete problems. The classical Goldberg model uses a single-point crossover and a single-point mutation, which somewhat decreases the accuracy of the obtained results. In the article, let us propose using a full two-point crossover with various mutations previously researched. In addition, the work studied the necessary probability to apply it to the crossover in order to obtain results that are more accurate. Results of the computation experiment showed that the higher the probability of a crossover, the higher the quality of both the average results and the best solutions. In addition, it was found out that the higher the values of the number of individuals and the number of repetitions, the closer both the average results and the best solutions to the optimum. The paper shows how the use of a full two-point crossover increases the accuracy of solving an inhomogeneous minimax problem, while the time for getting the solution increases, but remains polynomial.

1. Introduction

Allocation problems occupy one of the leading positions in the class of NP complete problems, and have not only important theoretical significance, but also are widely used in practice in many engineering and management tasks. The issue of effective planning becomes essential for any task where resource allocation between performers is required.

At the same time, it is very important for planning to have a certain effect connected with both the increase of economic indicators of processes (computing, technological, business, etc.), and operational indicators of the used equipment. Therefore, it is necessary to use optimization methods allowing getting such effect as, for example, the branch and bound method, an exhaustive search method, a dynamic programming method. However, these methods are too expensive in terms of the resources used, and the search for an optimal solution even on a multiprocessor computing (cluster) system can take months or even years.

Along with the exact methods of search optimization, approximations are also used that make it possible to get a quasi-optimal solution, very close to the optimal, but in a significantly shorter time. In
this connection, there arises a need to apply methods characterized by a polynomial dependence of the counting time on the problem dimension, and accuracy close to the optimal, or at least surpassing 10% the accuracy threshold. Such methods include evolutionary-genetic algorithms (EGA). Today the EGAs are very popular and well researched; they show good results in solving optimization problems.

2. Problem statement

In terms of the scheduling theory, the allocation problem can be formulated as follows. There is a maintenance system consisting of \( N \) independent devices \( P = \{ p_1, p_2, ..., p_n \} \). Final stream \( M \) – a set of independent parallel tasks (functional operators) comes to be served by \( T = \{ t_1, t_2, ..., t_m \} \).

\( \tau(t_i, p_j) \) – time for serving task \( t_i \) by device \( p_j \) determined by matrix \( T \). Generally, devices are not identical, task \( t_i \) can be served by any of the devices, and device \( p_j \) can process only one task at the same time. It is necessary to manage a continuous allocation of tasks through devices so that the time to execute the entire set of tasks should be minimal. The criterion for minimizing the time to complete the service of tasks is a minimax criterion, which can be defined as follows:

\[ f = \max_{1 \leq i \leq n} f_i \rightarrow \min, \text{ where } f_j = \sum_{\tau(t_i, p_j) \in T} \tau(t_i, p_j) \quad \text{--} \quad \text{the } p_j \text{ processor shutdown time} \quad [1, 2, 3, 4]. \]

In this work, as the basic algorithm for solving the inhomogeneous minimax problem, let us use the genetic algorithm - the modified Goldberg model \[5, 6, 7\], which differs from the classical Holland model \[6\] in using the tournament selection of individuals to a new generation that allows one to improve the performance of the algorithm with various mutation modifications in various crossovers.

3. The Modified Goldberg model

The modified Goldberg model can be described as a sequence of the following steps:

- **Step 1.** Formation of the initial generation, consisting of a given number of individuals.
- **Step 2.** Tournament selection of individuals and use of GA of crossover operators and mutations with a certain probability to create a new generation.
- **Step 3.** Checking the condition of the algorithm terminal work, which usually consists in the invariance of the best solution for a given number of generations. If the test fails, it returns to step 2.
- **Step 4.** The best individual is selected as the found solution \[8, 9\].

Graphically, the functioning of the modified Goldberg model can be presented in Figure 1. The best individual is selected and placed in the next generation. The process is repeated until the best individual of all generations repeats itself the number of times predefined by the developer \[8, 10\].

![Figure 1. An operation scheme of the modified Goldberg model.](image)

In this paper, let us consider both a classic single-point crossover and a full two-point crossover. The operation scheme of the classical single-point crossover is shown in Figure 2.
The operation scheme of the full two-point crossover is shown in Figure 3.

4. Experimental comparison of algorithms

In this paper, let us consider the dependence of effect that the crossover probability (the most important parameter of the genetic algorithm) and the crossover selection have on the accuracy of the inhomogeneous minimax problem solution, as well as the solution time. The work used mutations, which were thoroughly investigated in [6]. It is extremely problematic to solve this problem analytically, if it is possible in the framework of the study of algorithms, one does the computational experiments allowing one to collect the statistics of solutions made by various algorithms. A software module enabling one to organize computer experiments was developed. A Windows 10 Pro x64 computer was used to complete the computational experiment. The hardware used was a computer with the following configuration: a four-core Intel Core i7-7700k processor, 16 gigabytes of DDR4 RAM, a SSD hard disk of M2 format. This hardware system has been chosen in connection with the fact that the processor supports simultaneously 8 data processing streams and therefore allows performing parallel computations for large-dimensional tasks. For the computational experiment, software was written in the modern C# programming language in the Microsoft Visual Studio 2017 development environment.

To make the computational experiment, the authors used a task with the following parameters: 3 devices, 101 tasks, 100% mutation probability, 1 elite individual, the number of experiments - 50. The results of the experiment with a 50% crossover probability are given in Table 1.
Table 1. The result of the experiment with a crossover probability of 50%.

| Individuals* Repeat | Crossover type | One-point crossover | Two-point crossover |
|---------------------|----------------|---------------------|---------------------|
|                     | Elite          | Elite               | Elite               |
|                     | Simple mutation| Simple mutation with a possible repeat | Simple mutation with a possible repeat |
|                     | Simple mutation| One-bit mutation     | Simple mutation with a possible repeat |
|                     | Simple mutation| Separated two-bit mutation | One-bit mutation |
|                     | Simple mutation|                         | Separated two-bit mutation |
| 500* 500            | -              | 923                 | 927                 |
|                     | avg.           | 932.16              | 935.98              |
|                     | time           | 157.6               | 149.1               |
|                     | avg.           | 931.72              | 936.7               |
|                     | time           | 170.9               | 169.3               |
|                     | +              | 924                 | 930                 |
|                     | avg.           | 931.72              | 928                 |
|                     | time           | 191.7               | 917                 |
| 1000* 1000          | -              | 924.3               | 929.7               |
|                     | avg.           | 924.3               | 927.36              |
|                     | time           | 498.3               | 478.8               |
|                     | avg.           | 924.14              | 926.9               |
|                     | time           | 578.8               | 553.4               |
|                     | +              | 920                 | 924                 |
|                     | avg.           | 922.22              | 928.7               |
|                     | time           | 1075.9              | 103.4               |
| 1500* 1500          | -              | 917                 | 923                 |
|                     | avg.           | 922.22              | 925.48              |
|                     | time           | 1194.0              | 1148.4              |

The results of the experiment with a 75% crossover probability are given in Table 2.

Table 2. The result of the experiment with a 75% crossover probability.

| Individuals* Repeat | Crossover type | One-point crossover | Two-point crossover |
|---------------------|----------------|---------------------|---------------------|
|                     | Elite          | Elite               | Elite               |
|                     | Simple mutation| Simple mutation with a possible repeat | Simple mutation with a possible repeat |
|                     | Simple mutation| One-bit mutation     | Simple mutation with a possible repeat |
|                     | Simple mutation| Separated two-bit mutation | One-bit mutation |
|                     | Simple mutation|                         | Separated two-bit mutation |
| 500* 500            | -              | 922                 | 927                 |
|                     | avg.           | 930.72              | 936.32              |
|                     | time           | 149.4               | 139.6               |
|                     | avg.           | 931.16              | 936.7               |
|                     | time           | 169.0               | 158.6               |
|                     | +              | 919                 | 924                 |
|                     | avg.           | 924.56              | 928.2               |
|                     | time           | 508.2               | 501.6               |
| 1000* 1000          | -              | 919                 | 923                 |
|                     | avg.           | 924.04              | 929.02              |
|                     | time           | 576.3               | 555.6               |
|                     | avg.           | 922.68              | 926.56              |
|                     | time           | 1144.1              | 1049.5              |
| 1500* 1500          | -              | 919                 | 920                 |
|                     | avg.           | 927.28              | 926.48              |
|                     | time           | 1243.8              | 1164.2              |

The results of the experiment with a 100% crossover probability are given in Table 3.
Table 3. The result of the experiment with a crossover probability of 100%.

| Individuals * Repeat | Crossover type | One-point crossover | Two-point crossover |
|----------------------|----------------|---------------------|---------------------|
|                      | Simple mutation | Simple mutation with a possible repeat | One-bit mutation | Separated two-bit mutation | Simple mutation | Simple mutation with a possible repeat | One-bit mutation | Separated two-bit mutation |
| 500* 500             | min. 923 927 928 926 | 917 921 919 918     + | avg. 931 935.64 936.96 933.36 | 920.5 925.46 923.38 922.42 |
|                      | time 156.6 150.7 151.1 157.6 | 325.0 308.5 320.9 340.8 |
|                      | min. 921 927 926 924 | 917 921 919 918     + | avg. 930.76 934.9 937.48 934.32 | 919.68 925.44 923.08 922.62 |
|                      | time 178.0 166.7 169.2 175.3 | 353.7 335.8 344.2 344.1 |
| 1000* 1000           | min. 920 922 923 920 | 916 918 918 916     + | avg. 924.56 929.52 929.22 927.4 | 918.42 921.52 920 919.36 |
|                      | time 543.7 526.4 533.7 533.5 | 1191.2 1139.2 1184.1 1189.6 |
|                      | min. 921 923 923 918 | 917 918 918 917     + | avg. 925.18 928.74 929.04 926.72 | 918.36 921.8 920.18 919.04 |
|                      | time 614.8 606.6 610.1 616.4 | 1255.6 1230.1 1285.6 1249.5 |
| 1500* 1500           | min. 919 921 920 920 | 916 918 916 916     + | avg. 922.54 925.52 924.86 924.44 | 918.08 920.2 918.94 918.36 |
|                      | time 1150.0 1119.2 1113.1 1135.6 | 2554.2 2420.9 2488.9 2525.8 |
|                      | min. 918 921 920 919 | 916 918 916 916     + | avg. 922.1 925.64 926.02 923.88 | 917.36 920.48 918.34 918.44 |
|                      | time 1302.2 1289.8 1258.5 1275.7 | 2690.5 2536.0 2609.0 2747.4 |

Thus, after analyzing the results given in Tables 1-3, several conclusions can be drawn:

- The higher the probability the crossover has, the better the quality of both the average results and the best solutions is.
- The basic parameters of the genetic algorithm (the number of individuals and the number of repetitions of the best solution) affect the quality of the solution when using the modified Goldberg model. The greater the values they have, the closer both the average results and the best solutions to the optimum are.
- The use of a full two-point crossover uniquely increases the accuracy of an inhomogeneous minimax problem solution; however, the time for getting the solution increases.

References
[1] Alexeev O T 1987 Complex application of discrete optimization methods (Moscow:Nauka)
[2] Kononov A V 2014 Actual scheduling problems: computational complexity and approximation algorithms (Novosibirsk)
[3] Sergiienko I V and Kasphitskaya M F, 1981, Models and methods of solving a computer combinatorial optimization problems (Kiev: Naukova Dumka)
[4] Golovkin B A 1980 Parallel computing systems (Moscow: Energiya)
[5] Kobak V G, Porsheyan V M and Kuzin A P 2017 The use of different variants of mutation in the solution of an inhomogeneous minimax problem modified by the Goldberg model Scientific and practical journal Aspirant 10 (Rostov-on-Don: Prioritet)
[6] Holland J 1975 Adaptation in Natural and Artificial Systems An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence (USA: University of Michigan)
[7] Goldberg D Genetic 1989 Algorithms In Search, Optimization, and Machine Learning (USA: Addison-Wesley Publishing Company, Inc.)
[8] Kobak V G and Titov D V 2008 Investigation of the tournament selection in the genetic
algorithm for solving a homogeneous minimax problem *Mathematical methods in engineering and technology* 5 (Saratov)

[9] Kureychik V M and Knysh D S 2009 Parallel genetic algorithm. Models and problems of construction *Integrated models and soft computations in artificial intelligence* 5 (Moscow: Fizmatlit)

[10] Affenzeller M, Wagner S, Winkler S and Beham A 2009 *Genetic Algorithms and Genetic Programming Modern Concepts and Practical Applications* (USA: CRC Press)