Supercomputer modeling
of open pit limits for ore deposits

Denis V. Petrov, Vladimir M. Mikhelev, Konstantin V. Kuznetsov

National Research University Belgorod State University, Pobeda 85 Belgorod 308015 Russia

petrov@bsu.edu.ru, mikhelev@bsu.edu.ru, chapaev28@ya.ru

Abstract.   *This article provides a theoretical basis of the genetic algorithm for search open pits' limits and covers the parallel implementation of the formulated algorithmic approach to the multiprocessor system*

Keywords

Supercomputer simulation, the open pit limits, grid systems, parallel programming.

1 Introduction

One of the most important issues of planning open pit is determination of open pit limits. On computing point of view this task is very difficult, because building open pits' model requires processing big data.

Traditionally modifications of "floating cone" algorithm and Lerch-Grossman algorithm are used to solve this problem. But these algorithms have some drawbacks, such as difficulty of realization for distributed computing systems [2]. That is why it is necessary to develop new approaches to solve this problem.

2 General scheme of genetic algorithm

Optimization genetic algorithms are based on modeling of process of natural evolution. In a genetic algorithm, a population of candidate solutions (called individuals or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, the more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

Fist of all it is necessary to develop chromosome representation to make genetic algorithm [2]. The following decision may be suggested in context of search open pit limits problem. Any available form of open pit may be represented as an array of integers. Each element of such array shows the depth of pit on current column of three-dimension open pit model.

Suppose there is tree-dimension block model of open pit $PIxJxK$, every element of which is characterized by number (or weight)

$$p_{ijk}, i \in [0,I], j \in [0,J], k \in [0,K] \tag{1}$$

This number shows clear profit, which can be given as a result of its excavation, in consideration of percentage of minerals, cost of its excavation and market value of minerals. Then it can be characterized by vector $X = \{x1,...,xn\}$, $n = I * J$. In this vector depth of column with coordinates (i, j) is put into position

$$xq, q = i * I + j \tag{2}$$

This array is a chromosome, because it is a full description of individual or one particular form of pit.

Open pit limits are searched by repeated applying genetic operators to set of these individuals (population).

It is necessary to compute the value of fitness function for every chromosome to estimate chromosomes' fitness in population. This function must be chosen so that the higher its values match the higher quality of chromosome and better decision. Also this function must be positive-definite [3].

As fitness function for estimation of open pit limits' quality functions (3), (4) are used.

$$f(X) = S + \sum_{i=0}^{I} \sum_{j=0}^{J} \sum_{k=0}^{K} p_{ijk}, k \le x_q, q = i * I + j \tag{3}$$

$$S = \sum_{i=0}^{I} \sum_{j=0}^{J} \sum_{k=0}^{K} p_{ijk}, p_{ijk} < 0 \tag{4}$$

So, value of fitness function is computed as a sum of all blocks pijk, that are parts of open pit specified by current height-map X. This function complies with all restrictions, which have been determined above. It is positive-definite and it is so that the higher its values match the higher clear profit, which can be given as a result of all model blocks' excavation.

The algorithm terminates when a satisfactory fitness level has been reached for the population. It is determinated by condition (5).

$$| \max_{i=1,N}(f(X_i^k)) - \max_{i=1,N}(f(X_i^{k-1})) | < \varepsilon, \tag{5}$$

Where N is the size of population and k is the number of algorithm iteration. The number ε is a constant and it determinates accuracy of calculation.

3 Hibrid parallel genetic algorithm

Parallel version of this algorithm (hierarchical parallel genetic algorithm with two levels of parallelism) was developed for extension of search area, minimization probability of premature convergence and reduction the computation time (fig. 1).
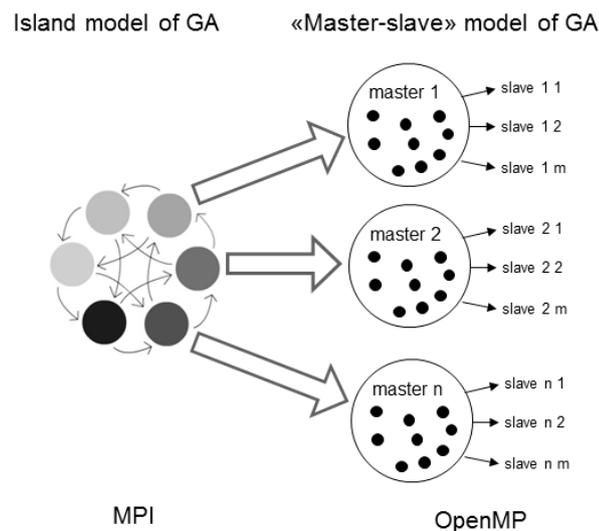
Fig. 1. Hierarchical parallel genetic algorithm with two levels of parallelism

The first level of parallelism is based on island model of multi-population parallel genetic algorithm. On this level acceleration is achieved through the usage of a few initial populations, which evolving independently and periodically exchanges the best genetic material. This exchange is carried out by the migration of individuals between populations.

This approach minimizes probability of premature degeneration of populations, increases their variety and accelerates the convergence of the search algorithm.

The second level is organized by the usage of single-population model of parallel genetic algorithm ("master-slave") for each subpopulation. It means that fitness function for each individual is computed in different threads. It helps to accelerate algorithm. One of the streams is "master", it is responsible for storage populations' data. Other threads are "slaves" and they compute fitness function.

Besides, as a result of attentive analyses of serial version of the algorithm several computationally independent parts were identified. These parts can be computed parallel during evolution of population. So, crossover and mutation can be computed independently.

This model of parallel genetic algorithm is very well suited to be implemented on large heterogenic multiprocessor systems such as grid-systems.

Technically these levels of parallelism are implemented as follow:

- on the first level MPI technology is used, which helps to distribute program on compute nodes of grid system;

- the second level is a result of usage multi-core processors on every node of grid system. The creation few threads helps use all these cores.

This distribution allows spread the load on the computer system more evenly and use multi-core nodes more effectively.

4 Computational experiment

Segment of grid system under control of Globus Toolkit is used as a platform for computational experiment. This segment consists of management server and computing nodes, which are based on Intel Xeon processors. Structure scheme of all these components is shown on fig. 2.
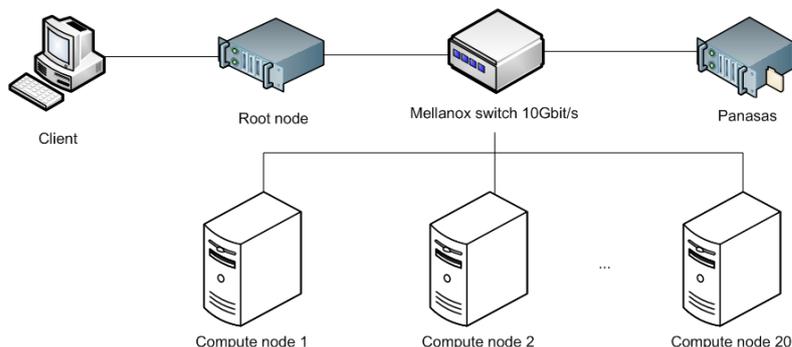
Fig. 2. Grid-system scheme

A summary of technical characteristics of cluster system is shown on table 1.

Tab. 1. Technical characteristics of cluster system

Parameter	Value
Processors' type	Intel Xeon
CPU frequency	2.4GHz
CPU count	40
Core count	320
RAM	1280Gb
HDD	8 Tb

Benchmark data for testing is generated by quasi-random method due to lack of access to real geological models of real open pits. Algorithm is tested on different minerals' spatial distribution models:

- sloping layering stratification;

- vertical stratification;

- uniform random distribution.

Example of tree-diminutions visualization of open pits' limits is shown on fig. 3.
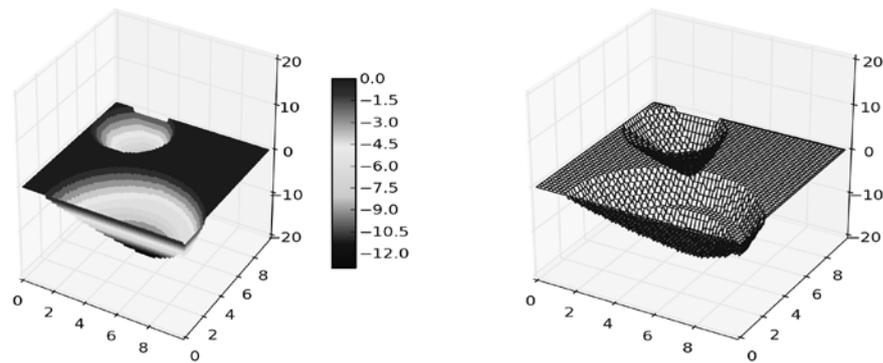
Fig. 3. Tree-dimential view of open pit model, 1:10000

5 Conclusion

Results of computational experiments had showed high promise of the proposed method to perform calculations on regular block model of open pits. The main advantage of this method is a providing a new principle of optimization problem pits, which let to work directly with the three-dimensional model of the pit. It greatly improves the adequacy of the model obtained. In addition, the flexibility to scale computing process can shorten the calculation model almost linearly with the number of nodes.

References

[1] Denby B., Schofield D. The Use of Genetic Algorithms in Underground Mine Scheduling., 1995
[2] Ramazan S., Dagdelen K., Johnson T.B. Fundamental tree algorithm in optimizing production scheduling for open pit mine design. Trans IMM (Section A: Mining Industry) vol. 114, 2005
[3] D. Rutkovskaya, M. Pilinskiy, L. Rutkovskiy, Neural networks, genetic algorithms and fuzzy systems, Goryachaya Liniya Telekom, 2007
[4] Shpakovskiy G.IThe implementation of parallel computing: MPI, OpenMP, clusters, grid, multicore processors, graphics processors, quantum computers. Minsk BGU, 2011