Software Implementation of Parallel Genetic Optimization Algorithms for CAM

Galina V. Verkhova[1], Sergei V. Akimov[2]
Department of automation of communication enterprises
The Bonch-Bruevich Saint-Petersburg State University of Telecommunications
St. Petersburg, Russia
[1]galina500@inbox.ru, [2]akimov-sv@yandex.ru

***Abstract*— The results of software implementation of parallel genetic algorithms for computer-aided design of technological processes are presented. The features of parallelization of genetic algorithms and software structure in C# within the framework of the .NET platform are considered.**

***Keywords*— parallel genetic algorithms; CAD system; CNC machines; C#; .NET**

I. Introduction

Modern trends in the development of industry are to constantly increase the degree of automation of technological processes and production and integration of production and business processes at all stages of the product life cycle [1–2]. Within the complex automation of technological processes and productions, the creation of high-performance software for numerical control (CNC) machines, which play a major role in modern flexible production, plays an important role [3]. Machines with numerical control is constantly evolving, there are new types of such machines, changing the specific weight of such implemented machines technologies.

In modern production, additive technologies are gaining popularity [4], as well as laser beam processing technologies [5–6]. These technologies require the development of complex software and algorithmic support necessary for the full potential of these technologies.

This article presents the results of the development and software implementation of parallel genetic algorithms to optimize the trajectory of the Executive mechanism of electron beam processing of materials used in the machine with numerical control. This type of processing allows you to create flexible production systems that can be used in both large-scale and small-scale production. It does not require the creation of templates, and the processing process is completely software-defined.

The most widespread of this class of processing of materials is laser processing. The development of software and algorithmic software, considered in the article, was carried out primarily for it. However, the considered algorithms can be used in any systems of electron beam processing of materials, as well as in any other methods of processing of planar blanks.

Genetic algorithms were chosen to solve the problem of optimization of the motion path of the Executive mechanism, as they are well suited for solving this class of problems, allowing the possibility of effective parallelization [7]. The possibility of parallelization of algorithms is relevant in view of the wide spread of multi-core and multi-processor computing systems, as well as the large computational complexity of solving the problem of optimizing the trajectory of the Executive mechanism.

II. Formalization of the problem

The formalization scheme of the problem of optimization of the trajectory of the Executive mechanism of laser processing of the machine with numerical control is presented in Fig. 1. All of the tracks that define the total path of movement, are subjected to segmentation and numbered. Each segment $S_{(i)}$ can be represented as follows:

$$S_{(i)} = \langle x_a, y_a, x_b, y_b, l, v, t, d, i \rangle, \tag{1}$$

where x_a, x_b, y_a, y_b are the coordinates of the beginning and end of the segment (*a* and *b* of the segment in Fig. 1), respectively, *l* is the length of the segment, *v* is the speed of the segment, *t* is the time taken to pass the segment, segment, *i* is the segment number in the processing queue.

The list of segments completely defines the processing path:

$$\text{List} = \langle S_{(i,j)} \rangle,\ i, j \in [1, n], \tag{2}$$

In expression (2), *n* is the number of segments, *i* is the segment number (identifier), *j* is the segment number in the list that defines the processing queue. Taking into account (1) and (2) we get the time spent for processing:

$$T = T_t + T_{idle} \tag{3}$$

Here, T_t is the time spent processing, T_{idle} is the total time taken to pass the carriage from one segment to the next, plus the time required to approach the first segment, and from the first to the base.

978-1-5386-5824-6/18/$31.00 ©2018 IEEE

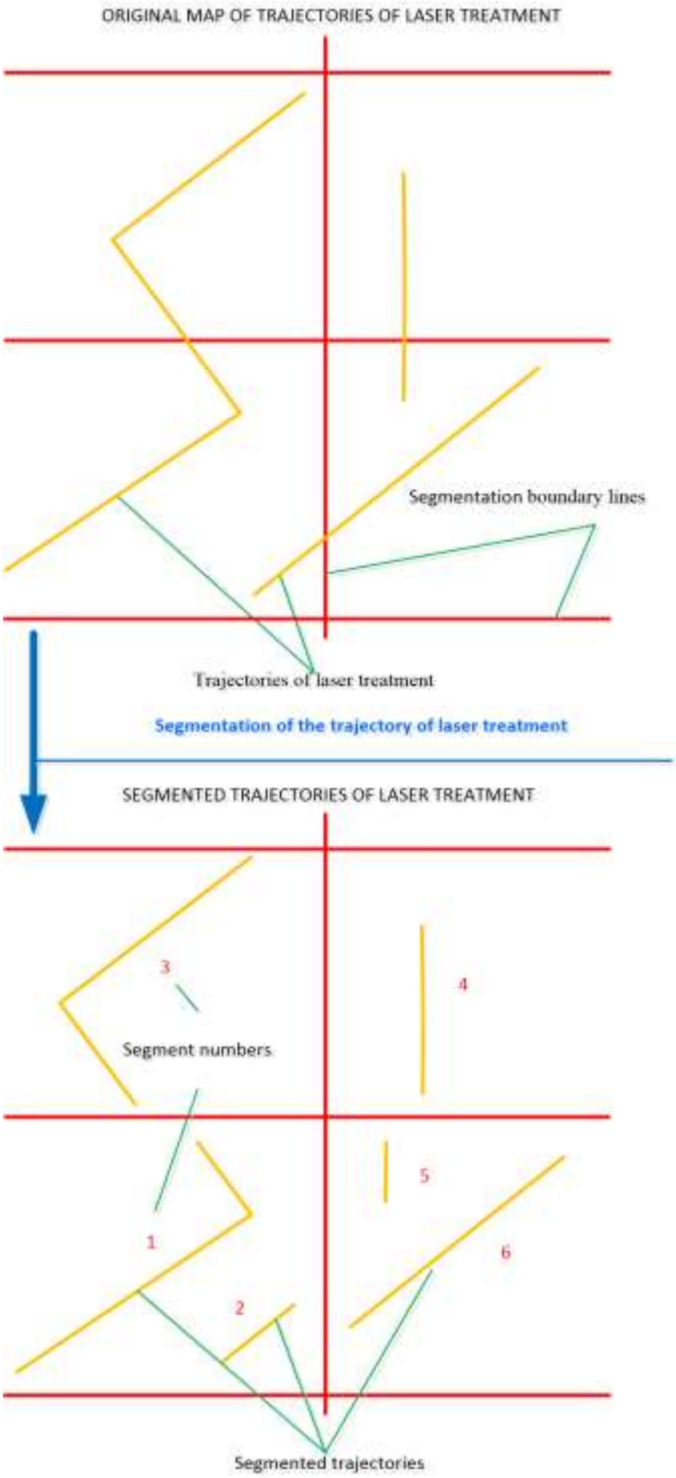

Fig. 1. Formalization of the optimization procedure of the trajectory of laser processing

Hence the optimization problem can be formulated as finding the sequence of processing segments and their orientation (the beginning and the end of processing), at which time T will be minimal. Since the time spent on laser processing of segments does not depend on the sequence of processing segments, the interest is the minimization of only the time T_{idle}.

III. Basic Genetical Algorithm of Optimization of the Trajectory of Movement of the Executive Mechanism of the CNC Machine

The task of determining the trajectory of the laser head of a machine with numerical program control can be solved using genetic algorithms. The scheme for the formation of the chromosome is shown in Fig. 2.

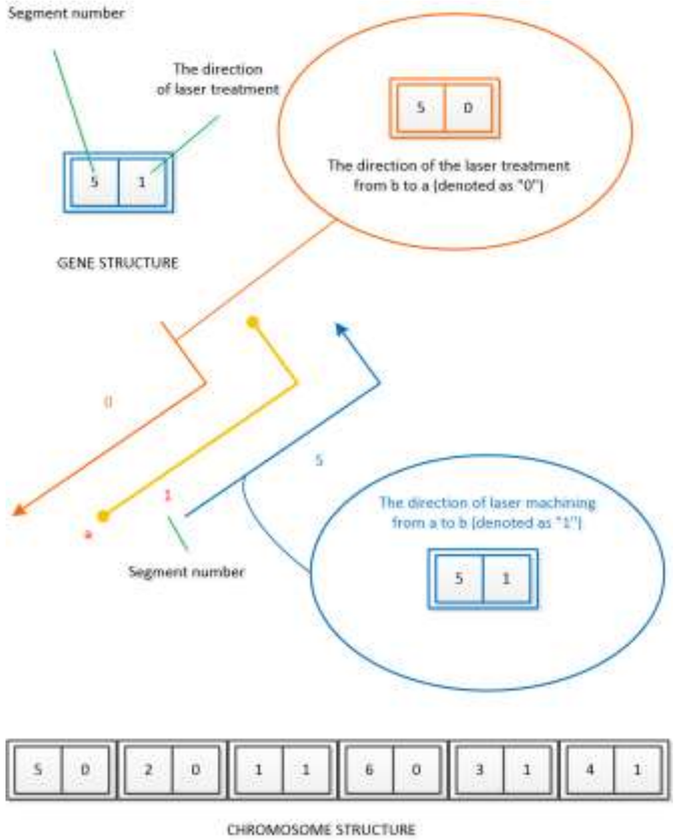

Fig. 2. Structure of the chromosome

Each individual gene contains information about the segment number and processing direction, which determines the movement of the laser head along the segment. The case where the laser head moves in the direction from *a* to *b*, we denote as "0", the case of movement in the opposite direction is "1". The sequence of genes in the chromosome determines the sequence of laser processing of individual segments, and thus, the chromosome uniquely encodes the trajectory of the laser head movement during the processing of the article.

Taking into account the peculiarity of the problem, it is necessary to use crossing-over that preserves all the genes that are part of the parent chromosomes. If this condition is not met, some of the segments to be processed will not be processed, and some segments will be processed twice. This kind of crossing-over is well studied, and is used to solve traveling salesman problems with the help of genetic algorithms.

The simplest form of the crossover operator is shown in Fig. 3. Another effective type of crossing-over for this task is a cyclic crossing-over, which has proved itself well in solving the traveling salesman's problems.

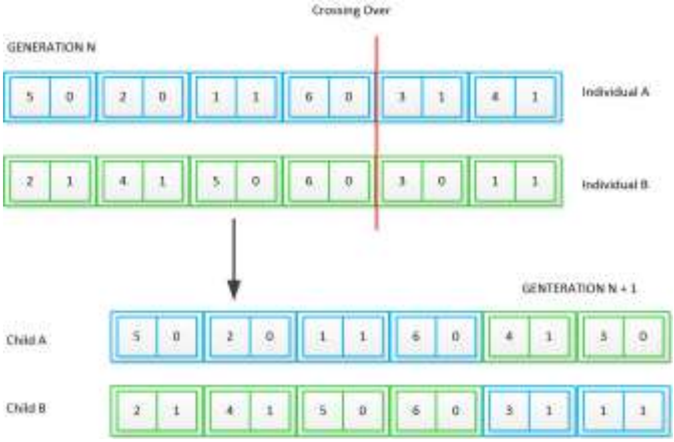

Fig. 3. Crossover operator

The mutation operator can consist in changing the direction of the segment processing and in rearranging the individual segments (Fig. 4). There are other types of mutation operators that significantly change the original chromosome.

Proceeding from the condition of the problem, it is clear that the coding and the crossing-over operator are in many respects similar to those used in the traveling salesman problem [8-9], solved with the help of genetic algorithms. The difference lies in the need to take into account the direction of the segment *d*. Since genetic algorithms have shown their effectiveness in solving the traveling salesman problem, there is every reason to believe that they will be effective for solving the problem of optimizing the trajectory of motion by radiation treatment.

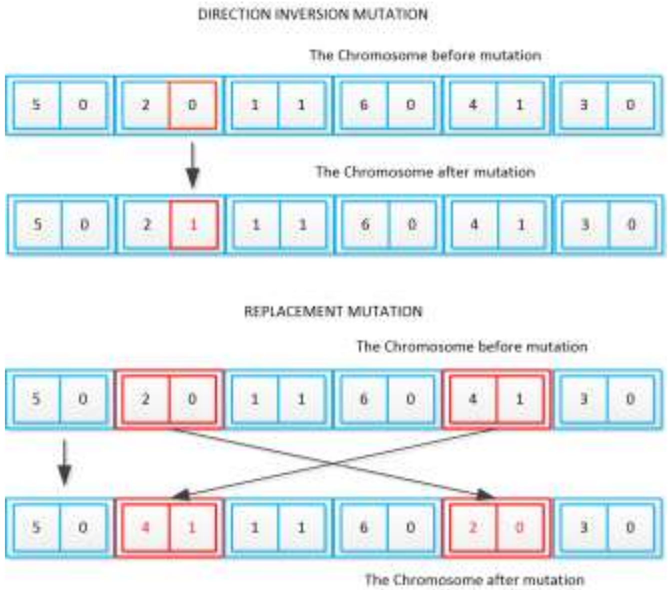

Fig. 4. Operators of mutation

IV. SOFTWARE IMPLEMENTATION OF THE ISLAND ALGORITHM

There are various models of parallelizing genetic algorithms [10]. The simplest model of parallelization is the "master-slabe" model, when the calculation of the fitness function is carried out in parallel, and the algorithm itself is consistent. This kind of algorithms is suitable for those tasks where the calculation of the fitness function requires considerable computational costs. This situation arises when using complex models of optimized (synthesized) objects, for example, electronic devices. However, in our case this model is not suitable, because the calculation of the fitness function does not require significant computational costs. Also, the cellular model does not fit.

Taking into account the specifics of the problem of optimization of the trajectory of the executive mechanism of the machine with numerical control, the best candidate is the island model of parallelization of genetic algorithms. The scheme of the software implementation of the island model is shown in Fig. 5. The exchange of individuals between populations occurs according to a cyclic scheme.

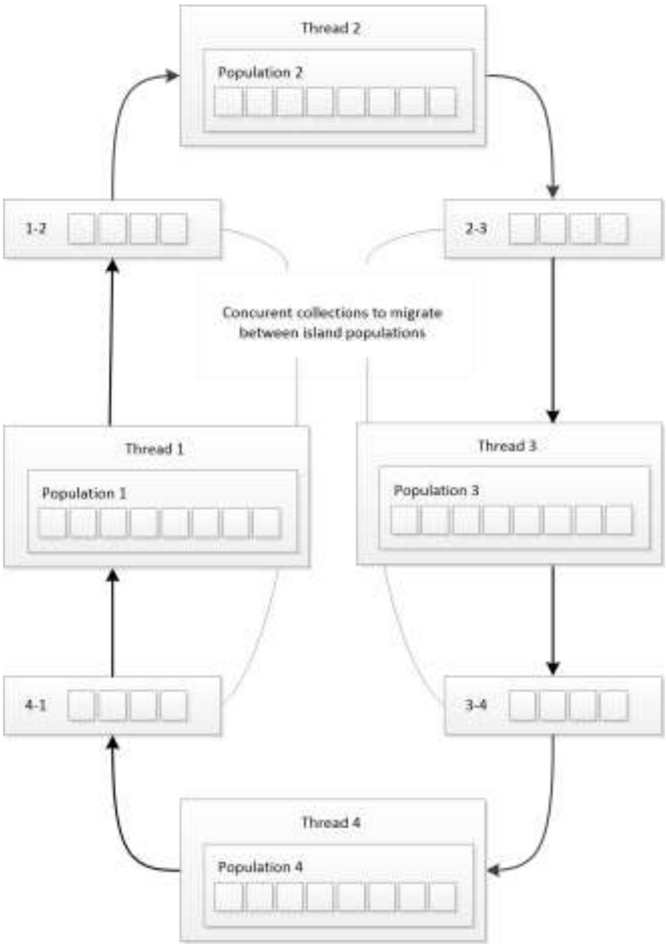

Fig. 5. Scheme of software implementation of the island model

This model is implemented as follows. From the main thread other threads (streams) are created, implementing the evolution of the island populations. The number of island populations should be equal to the number of cores in the computer system, which will provide maximum performance. The maximum part of the time in the threads is completely independent, which positively affects the efficiency of the parallel algorithm.. Only the thread that created it has access to the populations. This ensures that there are no conflicts and performance degradation due to locks, because no other thread can attempt to access this population. In each flow, conventional non-thread-safe computational constructs can be used to model evolution.

The exchange of individuals between populations is carried out with the help of concurrent collections. Each thread has access to the input and output concurrent collections. So, in the case of a quad-core processor for stream 1, these will be collections 1-2 (output) and 4-1 (output).

Before processing a new generation, each thread checks the presence of individuals in the input collection. If these are absent and the mechanism for initiating the exchange of individuals has not worked, then the processing of a new generation in the evolutionary process is launched. If the stream has detected the presence of migrant specimens in the input collection, it transmits as many of its individuals to the output collection, and replaces the ones received from the input collection.

If a mechanism triggers the migration, the thread transfers some of its individuals to the output collection, and immediately initiates the process of processing the new generation on a reduced population. Thus, the threads that implement evolution on island populations never go into standby mode. Population replenishment will occur after completion of the exchange cycle of individuals of all populations.

V. CONCLUSION

The implementation of this algorithm demonstrated its high efficiency of parallelization. Three variants of algorithms were sequentially programmed: 1) sequential, 2) parallelized with the help of standard means of .NET platform and 3) realizing the island model. Table 1 shows the results of loading a quad-core processor.

In the future it is supposed to continue research in this field, examining various strategies of selection, crossing-over and mutation, and adaptation of the genetic algorithm.

TABLE I. СРАВНЕНИЕ ЗАГРУЗКИ ПРОЦЕССОРА

Type of parallelization	***CPU utilization factor, %***	
	Min	***Max***
Sequential genetic algorithm	24	25
Parallelizing with the help of standard platform tools .NET (Parallel.For, Parallel.Foreach)	53	59
Island model	94	98

REFERENCES

[1] Samaranayake P., Ramanathan K., Laosirihongthong T. Implementing industry 4.0 – A technological readiness perspective. 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). 2017, pp. 529-533.

[2] Verkhova G.V., Akimov S.V. Multi-aspect modeling system objects in CALS. Proceedings of 2017 XX IEEE International conference on soft computing and measurements (SCM) 2017. pp. 449-451.

[3] Vasil'ev V.U. Trendy razvitiâ tehnologij i proizvodstva komponentov i uzlov mikrosistemnoj tehniki. Nano- i mikrosistemnaâ tehnika. 2016. Vol. 18, No. 7. Pp. 403–415. (in Russian)

[4] Nguyen D.S., Vignat F. Topology optimization as an innovative design method for additive manufacturing. 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). 2017, pp. 304-308.

[5] Afanasyev M.Y., Fedosov Y.V., Nemkova A.A. Designing features of power optical units for technological equipment. Scientific and Technical Journal of Information Technologies, Mechanics and Optics. 2015. Vol. 16, no. 2 (102), pp. 244–250. DOI:10.17586/2226-1494-2016-16-2-244-250.

[6] Afanas'ev M.A., Gribovskij A A. Koncepcia adaptivnoj platformy tehnologiceskogo oborudovania. Izv. vuzov. Priborostroenie. 2015. Vol. 58, No. 4. Pp. 268-272. DOI 10.17586/0021-3454-2015-58-4-268-272. (in Russian)

[7] Baesler F., Palma C. Multiobjective parallel machine scheduling in the sawmill industry using genetic algorithms. The International Journal of Advanced Manufacturing Technology. 2014. 74(5), pp. 757-768.

[8] Oliver I.M., Smith D.J., Holland J.R.C. A study of permutation crossover operators on the traveling salesman problem. Proceedings of the Second International Conference on Genetic Algorithms and Their Applications. MIT, Cambridge, MA, Lawrence Erlbaum Associates: Hillsdale, New Jersey. 1987. Pp. 224–230.

[9] Davoian K., Gorlatch S. A modified genetic algorithm for the traveling salesman problem and its parallelization. Artificial Intelligence and Applications AIA 2005. Innsbruck (Austria). 2005. Pp. 453-116.

Emel'anov V.V., Kurejcik V.V., Kurejcik V.M. Teoria i praktika evolucionnogo modelirovania. Moscow. Publ. Fizmatlit. 2003. 432 p. (in Russian)