Planning of Drilling Sequence Using the Swarm Intelligence Method

Conference Paper · April 2017

1 author:

Danijela Pezer
University Department of Professional Studies, University of Split, Croatia

PLANNING OF DRILLING SEQUENCE USING THE SWARM INTELLIGENCE METHOD

Danijela PEZER

Mechanical Engineering Faculty in Slavonski Brod, University of Osijek
Trg I.B. Mažuranić 2, Slavonski Brod, Croatia

Abstract

In the introduction of this paper the Traveling Salesman Problem was described and a brief review of the literature with regard to the applicability of the proposed method in practice, for various production problems, was given. Ant Colony Optimization (ACO), and the genetic algorithm (GA) methods used for optimization are explained giving the basic algorithm structure. Furthermore, the basic mathematical background was given, as well as the proposed ACO algorithm with the application for the given problem – tool path optimization in case of sequence of hole drilling. According to the steps, the implementation of the proposed ACO algorithm realized in the MATLAB program was also described. In the last chapter, the results achieved with the proposed ACO algorithm, in comparison with the results achieved by the genetic algorithm, and the selected CAM software, were given.

Keywords: Ant Colony Optimization, genetic algorithm, CAM software, drilling

1. INTRODUCTION

Ant Colony Optimization is metaheuristics [1, 2] for solving difficult combinatorial problems, inspired by the behaviour of different species of ants, in search for the shortest route. The first ACO algorithm, Ant System (AS), is proposed to solve the Traveling Salesman Problem (TSP) [3]. Except for the TSP problem, the AS algorithm has found application in many other types of combinatorial optimization problems such as Quadratic assignment problem [4, 5], Vehicle routing [6-8], job scheduling [9, 10], telecommunication routing [11, 12] and other. Many other algorithms inspired by the AS algorithm are proposed, such as ant-Q [13], the Ant Colony System [14], MAX-MIN ant system (MMAS) [15], rank-based ant system (ASrank) [16], hyper-cube ant system [17] and KCC-Ants [18].

With the combinatorial optimization problems we want to find discrete values for the variables that lead to the optimal solutions in relation to the specific objective function. The combinatorial problems, although easy to understand, are defined as the NP (non-deterministic polynomial-time) hard problems. The most studied combinatorial problem is the Traveling Salesman Problem. With the TSP problem, a salesman wants to find the shortest way to serve the customers in the neighbouring cities, starting from the initial city, in which he will return at the end of his tour, and at the same time to visit each city only once.

The TSP can be represented as a completely weighted undirected graph $G = (V, E)$ if it is symmetrical, completely weighted directed graph $G = (V, A)$ if it is asymmetrical. The set $V = \{1, 2, \ldots n\}$ is a set of vertices that represent the cities, $E = \{(i, j): i, j \in V, i < j\}$ represents a set of edges which completely connect the vertices, while $A = \{(i, j): i, j \in V, i \neq j\}$ is a set of arcs. For each edge, or arc (i, j), a value d_{ij} is assigned, which may indicate the distance, time,cost or other factors of interest associated with the edges or arcs.

With the standard TSP problem, the assumption is that the square cost matrix, $D = \{d_{ij}: (i, j) \in V, i \neq j\}$, is symmetric $d_{ij} = d_{ji}$, i.e. the distance is equal in both directions. Another standard assumption is that the distance matrix $D = (d_{ij})$ satisfies triangle inequality in the case when $d_{ij} \leq d_{ik} + d_{kj}$, for $\forall\, i, j, k \in V$. The objective of problem is to find the minimum Hamiltonian cycle, by which the tour closes after each of the $n = |V|$ vertices of G is visited only once.

Since the efficiency of production is of importance for each manufacturing process, the tool path optimization in the drilling process is essential, because it leads to increase of productivity and saving of the production costs. This particularly applies to the process of drilling a large number of holes, where the tool that performs the drilling operations must visit a significant number of places in order to perform drilling, where the production is dependent on the time required for drilling. Ultimately, the goal in mass production is to produce the high-quality products at reasonable prices with the reduced product costs and increased production efficiency. The basic assumption at the TSP problem is that the salesman should return to the starting point (city) from which the tour started, which is called a closed tour, otherwise we are dealing with an open tour, which will be considered in this paper.

This paper proposes the use of ACO and GA algorithm for the analysis of the problem based on the traveling salesman problem, or troubleshooting of tool path optimization at drilling sequence.

2. ANT COLONY OPTIMIZATION ALGORITHM FOR DRILLING SEQUENCE PLANNING

Ants in the nature, in a simple manner with joint forces, perform complex tasks such as transportation of food and finding the shortest path to the food source. The ACO algorithm mimics this principle using a very simple communication mechanism by which an ant colony can find the shortest path between two points. Since ants do not have a well-developed sight, during the food search, they leave on the ground chemical traces - pheromones. Pheromones are actually fragrant and volatile substances, and their role is to lead other ants toward the target point. The greater the amount of pheromones on a particular path, the greater the chance that the ants will choose that path, the default ant chooses the route based on the amount of pheromone scent. Pheromones evaporate over time (evaporation process) and the amount that was left by one ant depends on the amount of food (reinforcement process). When ants are faced with an obstacle, the probability of selecting the left or right way is equal for each ant. If the left path is shorter than the right, and requires less travel time, an ant will finish the travel leaving a greater level of pheromones on that path. The more ants choose the left path, the higher is the pheromone trail. Algorithm 1 [19] shows the basic ACO algorithm, in which the pheromone traces are initialized as a first step. The algorithm is mainly made up of two iterative steps: solution construction and pheromone update.

Algorithm 1 – General algorithm for ACO

```
Initialize the pheromone trails;
Repeat
   For each ant Do
      Solution construction using the pheromone trail;
      Update the pheromone trails:
         Evaporation;
         Reinforcement;
 Until Stopping criteria
 Output: Best solution(s) found
```

The problem to be considered in this paper, the determination of the optimum sequence of drilling, based on the traveling salesman problem (symmetrical TSP), where each hole is drilled only once, and at the same time the coordinates of the defined holes are known. The goal is to find the minimum tool path length d_{ij}, which represents the Euclidean distance and for the plane problems, it is expressed by the equation [20]:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

The limitations of the model of problem:

- selection of the initial holes depends on the random algorithm selection
- after the drilling of last hole, the tool remains in its current position
- the tool wear was not considered
- drilling a hole of the same diameter (and depth).

The fitness function, or the total minimum tool path is defined as:

$$\min \sum_{(i,j)\in E} d_{ij} x_{ij}, \tag{2}$$

with degree constraints:

$$\sum_{j\in V|(i,j)\in E} x_{ij} + \sum_{j\in V|(j,i)\in E} x_{ji} = 2, \qquad i \in V \tag{3}$$

With the TSP problem, it is necessary to ensure a continuous tour of holes, i.e. that the edges/arcs are associated with the initial hole all the way of the tour, and each hole has to be visited. In order to ensure that, the limitation is set to eliminate all possible subtours (the solution containing degenerative visits/tours between the central vertex that are not associated with the initial hole).

Subtour elimination constraints:

$$\sum_{(i,j)\in E|(i,j)\in S} x_{ij} \leq |S| - 1, \quad S \subset V, 3 \leq |S| \leq n - 3 \tag{4}$$

where *S* is a subset of vertices of *G*.

Integrality constraints:

$$x_{ij} = 0 \text{ ili } 1, \ (i,j) \in E \tag{5}$$

The expression (3) ensures that each hole is drilled only once, and according to the second limitation, the expression (4), the subtours are not allowed, and the possibility that the solution has more than one path was eliminated.

Algorithm 2 [19] shows the Ant Colony algorithm for sequential hole drilling based on the Traveling Salesman Problem.

Algorithm 2 – Ant colony algorithm for the drilling sequence based on TSP problem

Initialize the pheromone information;
Repeat
  For each ant **Do**
    Solution construction using the pheromone trails:
     $N = \{1,2,\dots,n\}$ /*Set of potentially selected holes*/
     Random selection of the initial hole *i*;
     Repeat
       Select new hole *j* with probability
       $P_{ij} = \frac{\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}}{\sum_{k\in N} \tau_{ik}^{\alpha} \times \eta_{ik}^{\beta}}$;
       $N = N - \{j\}; i = j;$
     Until $N = \emptyset$
  End For
  Update the pheromone trail:
    For $i, j \in [1,n]$ **Do**
      $\tau_{ij} = (1-\rho)\tau_{ij}$ /*Evaporation*/;
    For $i \in [1,n]$ **Do**
      $\tau_{i\pi(i)} = \tau_{i\pi(i)} + \Delta$ /*π: best found solution*/;
Until Stopping criteria
Output: Best solution found

4. SIMULATION RESULTS

The proposed ACO algorithm for the optimization of drilling sequence, has been tested on a prismatic workpiece (Figure 1), where the minimization of tool path length is achieved by using the MATLAB software. It is necessary to determine the minimum tool path, with the planned drilling of 158 holes. The starting point of drilling depends on the random algorithm selection. The results were compared with the results achieved with the CATIA V5 CAM software and the genetic algorithm (GA).

Figure 1 – Prismatic workpiece

The good performance of the ACO algorithm is significantly affected by the parameters α, β and ρ, where α affects the importance of pheromone trail, β is the importance of visibility (hole distance), while ρ is the evaporation factor. By testing the algorithm, it was concluded that the algorithm gives better solutions with less population size, i.e. the number of ants (nAnt), in relation to the population size which is equal to the number of planned holes for drilling, and the value nAnt=70 taken as the optimal one in this case. The change of evaporation rate ρ, certainly affected the solution. With ρ increasing, the effect of pheromone trail in relation to the visibility also increased. It is possible to increase the effect of pheromone trail in the algorithm performance if the parameter relating to the visibility increases. With the constant ρ, Q, nAnt, and the number of iterations, and gradual changing of the parameters α and β, with the multiple algorithm starting, the following results were obtained and shown in Table 1. The program was performed on a work station HP EliteBook 8560w, i72820 QM CPU, 2.3GHz, 16 GB of RAM.

Table 1 – Parameters of the ACO algorithm displayed with a minimum tool path and algorithm run-time

α	*β*	*ρ*	*nAnt*	*maxIter*	*Best sol.*, mm	*CPU time,* s
1	0	0,45	70	900	5171,9716	692,1709
1	1	0,45	70	900	875,4713	769,9698
1	2	0,45	70	900	871,7914	766,7427
1	3	0,45	70	900	867,7351	985,7253
1	**4**	**0,45**	**70**	**900**	**852,4165**	987,4733
1	5	0,45	70	900	866,9804	988,5877
1	6	0,45	70	900	875,7082	984,0857
1	7	0,45	70	900	881,8645	984,5588
2	1	0,45	70	900	1017,2737	478,3352
3	1	0,45	70	900	1121,0248	511,4258
4	1	0,45	70	900	1140,8844	628,9772
5	2	0,45	70	900	962,3287	599,2406

From Table 1, it is evident that if the parameter relating to the importance of the pheromone trail α is highlighted, the importance of the mutual distance between the holes, or the visibility of the same, is not so high. The same applies to the case when β=0, where it can be seen that the results are significantly lower than the optimum. Ants who passed for the first time the tour route of holes and set the shortest route have an impact on such a result, but also the strongest pheromone trail along the way, and the rest of the ants continue to follow the path set by the previous ants. In this way, the possibility of different solutions, or convergence to the global optimum is reduced. The algorithm was started 1500 times, using different combinations of parameters, and Table 1 shows the results in a certain range, because due to the extensive number of combinations it is not possible to show the complete results. The testing was conducted for the number of population which is equal to the number of holes, with the same combination of parameters α, β and ρ (Table 1), but the solutions are somewhat inferior than those represented in Table 1, while the best achieved solution is 864,9167 mm. Afterwards, the test was conducted in which the evaporation factor ρ is changed, while keeping the parameters α and β constant (the combinations according to Table 1) to determine with which values the satisfactory results are acheived. The range of value of parametar ρ is from 0-1, and during the testing, the values of ρ=0,05, ρ=0,1, ρ=0,2, ρ=0,4, ρ=0,45, ρ=0,5, ρ=0,6, ρ=0,7 and ρ=1 were taken.

According to the algorithm testing, with the different combinations of the above mentioned parameters, for a given problem the following parameters were determined as optimal: α=1, β=4, ρ=0,45, Q=1, *nAnt*=70, the maximum number of iterations *maxIter*=900, with which a minimum total tool path length in the amount of 852,4165 mm (in 253 iterations) was achieved.

The results of ACO algorithm were compared with the results of testing performed using the genetic algorithm (GA). The initial population of chromosomes is generated randomly, each chromosome representing a set of holes that need to be drilled, and each gene represents the assigned number of holes. The starting point of drilling was selected randomly, as well as with the ACO algorithm.

For the defined problem (the tool path optimization), the number of individuals in the population and the number of generations (iterations) are determined experimentally. Genetic algorithm was running several times with different combinations of the population size and stopping conditions, in order to determine the parameters of the algorithm providing the best solution. With this algorithm, the minimum total tool path length in the amount of 860,8151 mm (in 5300 iterations) with a population size of 5,000 and the max. number of iterations 10,000 was achieved. The results are different from the results of previous research shown in [21] because it was used an another method of selection. In this case, for both methods (ACO and GA) roulette wheel selection was used because it gives better results.

The total tool path length achieved by the CATIA V5 software is 981,078 mm. The drawing of prismatic workpiece is imported from the CAD program, and the holes have been selected for drilling to enable generating of the tool path. The CATIA software offers a choice to select all holes at the same time, which significantly speeds up the process. The order of execution of drilling starts from a randomly selected starting point defined by the user, while the sequence of drilling of the remaining holes is performed based on the embedded module for the tool path optimization.

Figure 2 shows the optimal tool path achieved by the proposed ACO algorithm, genetic algorithm and the CATIA V5 software.

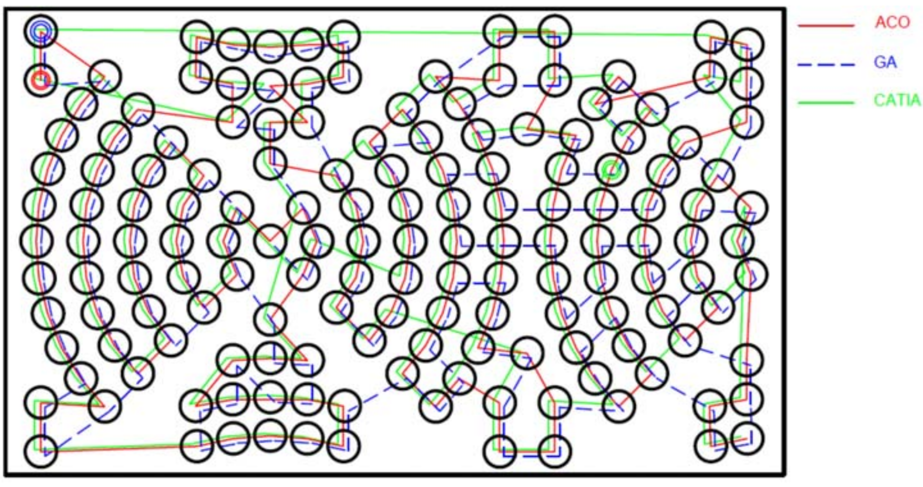

Figure 2 – The optimal tool path achieved by the proposed ACO algorithm, GA and CATIA V5 software

The proposed ACO algorithm finds a satisfactory solution with the smaller number of iterations compared to the GA, but the execution time of one iteration is much longer in relation to the GA, which is unfavorable in the case with significantly large number of holes (up to several thousand) that need to be drilled. Increase of the number of iterations directly affects the algorithm runtime, as well as the quality of obtained solutions. A larger number of iterations gives a better solution (but it also increases the algorithm runtime), and it is necessary to determine the optimal number of iterations with which the algorithm will find a satisfactory solution in a relatively short period, or in real-time, and which would be satisfying (acceptable). In that way, we directly affect the quality of obtained solutions, and by increasing the size of population, we reduce the probability of premature convergence to a local optimum.

In addition to defining the number of iterations, the selection of the population size or the number of ants is essential, with whose increase we directly affect the quality of obtained solutions and reduce the probability of premature convergence to a local optimum. The algorithm has been tested with different numbers of iterations, the population size, and other parameters mentioned above.

Figure 3 and Figure 4 provide a comparison of the impact of the number of iterations in the relation to the objective function (best obtained solution). The figures show that the value of minimum tool path length improves approximatively from the initial to the final iteration, which is an indicator of optimization occurring. It is also evident that the proposed ACO algorithm converges faster compared to the GA, and it also finds a better solution with a quite smaller number of iterations compared to the proposed genetic algorithm.

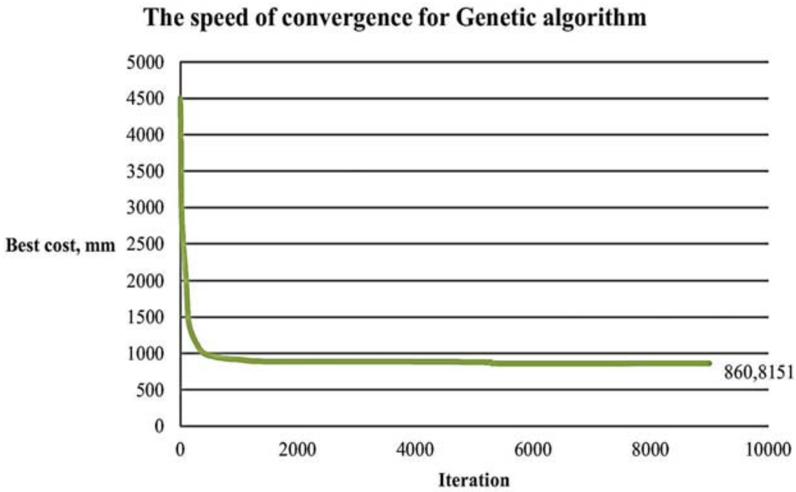

Figure 3 – The speed of convergence of GA algorithm for drilling sequence problem

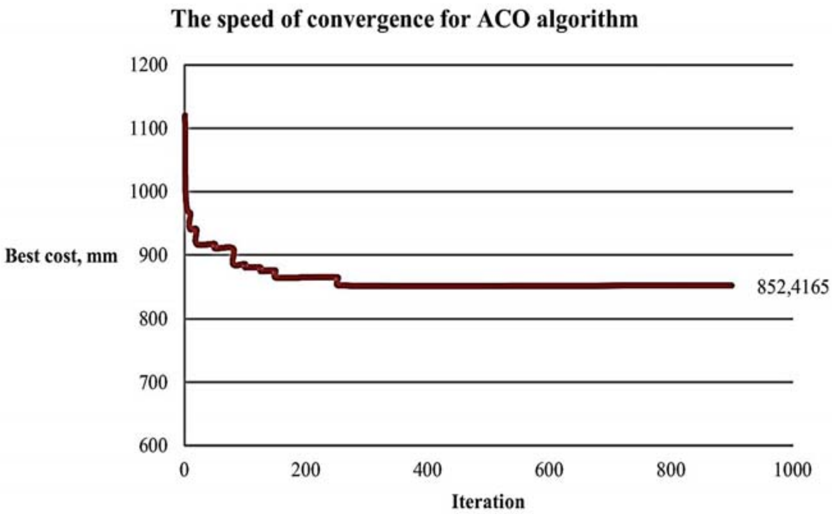

Figure 4 – The speed of convergence of GA algorithm for drilling sequence problem

5. CONLUSION

In this paper, with the assistance of Ant Colony Optimization method, was tried to find a path of drilling operation sequence that provides the shortest route and reduce the total work time and increase efficiency, compared to the path obtained by the genetic algorithm, and the CATIA V5 CAM software. The proposed ACO algorithm finds a satisfactory solution with the smaller number of iterations compared to the GA, but the runtime of one iteration is much longer in relation to the GA. Since the both algorithms (ACO and GA) were first tested with a small number of holes (7 holes), by increasing the number of holes it was evident that runtime of one iteration is longer for a larger number of holes. Based on the above we can conclude that the ACO algorithm is unfavorable in the case with significantly large number of holes (up to several thousand) which need to be drilled. The efficiency of genetic algorithm is reflected in the selection of genetic operators (selection, crossover and mutation), as well as in the selection of parameters such as population size, the number of generations or iterations, the probability of crossover and mutation probability, which significantly influence the behaviour of these operators. The proposed ACO algorithm is reliable to use for a given problem, because in the end, in a relatively short time, it finds the (sub)optimal solution. The obtained results are much better than the results obtained from the CAM software, especially if we take into account that the defined holes are very close to each other, and in the case of different schedule, the results would be also more significantly expressed.

In the further research, the optimal tool path of drilling sequence will be solved using the swarm intelligence method - Artificial Bee Colony (ABC). There is also the possibility of modifying the proposed ACO algorithm, and depending on the possible results achieved by using the ABC algorithm, perhaps even the ability to combine these two algorithms. Further research might go in the direction of parameter optimization which significantly influences the final solution, which would ultimately lead to even more favourable results of the proposed algorithm.

8. REFERENCES

[1] Dorigo, M., Caro, G. D., Gambardella, L. M.: *Ant algorithms for discrete optimization*, Artificial Life, 5(1999)2, 137-172.

[2] Dorigo, M., Stützle, T.: *Ant colony optimization*, MIT Press, 2004.

[3] Dorigo, M., Maniezzo, V., Colorni, A.: *Ant system: Optimization by a colony of cooperating agents*, IEE Transactions on Systems, Man and Cybernetics, Part B – Cybernetics, 26(1996)1, 29-41.

[4] Gambardella, L.M., Taillard, E., Dorigo, M.: *Ant colonies for the quadratic assignment problem*, Journal of Operational research Society, 50(1999)2, 167-176.

[5] Maniezzo, V., Colorni, A.: *The ant system applied to the quadratic assignment problem*, IEEE Transactions on Knowledge and Data Engineering, 11(1999), 769-784.

[6] Bullnheimer, B., Hartl, R.F., Strauss, C.: *An improved ant system algorithm for the vehicle routing problem*, Annals of Operations Research, 89(1999), 319-334.

[7] Bell, J. E., McMullen, P. R.: *Ant colony optimization techniques for the vehicle routing problem*, Advanced Engineering Informatics, 18(2004)1, 41-48.

[8] Montemanni, R., Gambardella, L. M., Rizzoli, A. E., Donati, A.: *Ant colony system for a dynamic vehicle routing problem*, Journal of Combinatorial Optimization, 10(2005)4, 327-343.

[9] Heinonen, J., Pettersson, F.: Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem, *Applied Mathematics and Computation*, 187(2007)2, 989-998.

[10] Udomsakdigool, A., Kachitvichyanukul, V.: *Multiple colony ant algorithm for job-shop scheduling problem*, International Journal of Production Research, 46(2008)15, 4155-4175.

[11] Di Caro, G., Dorigo, M.: AntNet: Distributed stigmergetic control for communications networks, Journal of Artificial Intelligence Research, 9(1998)1, 317-365.

[12] Schoonderwoerd, R., Holland, O., Bruten, J., Rothkrantz, L.: *Ant-based load balancing in telecommunications networks*, Adaptive Behavior, (1997), 169-207.

[13] Gambardella, L., Dorigo, M.: *Ant-Q: A reinforcement learning approach to the traveling salesman problem*, Proceedings of the twelfth international conference on machine learning (ML-95), Palo Alto, CA, 1995, 252-260

[14] Dorigo, M., Gambardella, L.: *Ant colony system: a cooperative learning approach to the traveling salesman problem*, IEEE Transactions on Evolutionary Computation, 1(1997)1, 53-66.

[15] Stützle, T., Hoos, H. H.: *MAX–MIN ant system*, Future Generation Computer Systems, 16(2000)8, 889-914.
[16] Bullnheimer, B., Hartl, R., Strauss, C.: *A new rank-based version of the ant system: A computational study*, Central European Journal for Operations Research and Economics, 7(1999)1, 25-38.
[17] Blum, C., Roli, A., Dorigo, M.: *HC-ACO: The hyper-cube framework for ant colony optimization*, Proceedings of MIC'2001 - metaheuristics international conference, Porto, 2001, 399-403.
[18] Naimi, H. M., Taherinejad, N.: *New robust and efficient ant colony algorithms: Using new interpretation of local updating process*, Expert Systems with Applications, 36(2009)1, 481-488.
[19] Talbi, E-G.: *Metaheuristics: from design to implementation*, University of Lille –CNRS-INRIA, John Wiley & Sons Inc., Hoboken, New Jersey, 2009.
[20] Matai, R., Singh, S.; Mittal, M.: *Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches*, www.intechopen.com/books/traveling salesman-problem-theory-and-applications/traveling-salesmanproblem-an-overview-of-applications-formulations-and-solution-approaches, 2016-05-10.
[21] Pezer, D.: *Efficiency of Tool Path Optimization Using Genetic Algorithm in Relation to the Optimization Achieved with the CAM Software*, Procedia Engineering, 149(2016), 374-379.