Optimization of the trajectory of movement of a parallel robot based on a modified ACO algorithm

L A Rybak¹, S Y Khalapyan² and D I Malyshev¹

¹ Belgorod State Technological University named after V G Shoukhov, 46, Kostyukov Street, Belgorod, 308012, Russia
² Stary Oskol Technological Institute (Branch) named after A A Ugarov, NUST MISiS, 42, Makarenko Microdistrict, Stary Oskol, 309516, Russia

E-mail: ribgtu@gmail.com

Abstract. The article describes the use of a modified ACO algorithm to optimize the trajectory of a planar 3RPR robot. The trajectories of the motion of agents of the first generation were obtained in accordance with the classical and modified algorithm. The efficiency of using the modified trajectory planning algorithm was demonstrated, which made it possible to reduce the length of the original trajectories and the time required to process them. At the same time, in order to save time, the trajectories of movement obtained during its operation can be subjected to discrete local optimization. The result of the work of the algorithms is a set of tasks reflecting the change in each of the input coordinates of the mechanism over time, which can be directly used during the operation of the robot control system. A detailed description of the algorithm and the results of mathematical modeling are presented.

1. Introduction
One of the most important tasks, the solution of which depends on the effectiveness of the use of robots of a parallel structure, is the planning of the trajectory of their movement. Currently, there is an increased attention of researchers to this problem, which indicates its relevance. In [1], an overview of various approaches to non-adaptive methods of controlling the movement of robots in free space is given. In this case, the problem of movement along a given trajectory is considered in cases when the trajectory is specified in a fixed coordinate system and relative to the target. In the second case, the position of the target cannot be known in advance and is determined using a sensor system. Two types of movements are considered: moving to a predetermined position and moving along an entirely specified trajectory. For non-dynamic control laws, satisfactory results are observed when solving problems of positional control and working off slow trajectories. For accurate testing of fast trajectories, dynamic control schemes are required. In [2], the problem of defining a trajectory of motion is also noted. It is indicated that the best way to set the trajectory is to control the robot in external coordinates, however, such systems are at the research stage. In [3], modern methods of intellectual planning of the trajectory of moving objects on flat environments with stationary obstacles, including grapho-analytical, neural network, fuzzy and bionic methods, as well as genetic search procedures are considered. Let us consider the possibility of using intelligent methods for planning the trajectory of parallel robots.

2. Synthesis of the modified ACO algorithm
Planning of the trajectory of parallel robots should be carried out taking into account the existing mechanism working space limitations, related to its geometric characteristics, sufficient remoteness of the trajectory of movement from the points of singularity and the workpiece size [4,5]. Optimization of the trajectory can, in particular, be aimed at minimizing the movement duration. In this case, it is advisable to carry it out in the space of the input coordinates of the mechanism associated with the operation of drive motors, choosing the modified Chebyshev metric as the objective function.

\[
\sum_{j=1}^{m} \left( \max_{i \in \{1,2,\ldots,K\}} \left| l_{i,j} - l_{i,1,j} \right| + \alpha \sum_{j=1}^{m} \left( l_{i,j} - l_{i,1,j} \right)^2 \right) \rightarrow \min,
\]

where \( l_{i,j} \) is the state of the input coordinate of the mechanism (length or drive link rotation angle) at the \( i \)-th step of the trajectory; \( \alpha \ll 1 \) is a small weighting factor; \( j \in \{1,2,\ldots,m\}, i \in \{1,2,\ldots,K\} \).

Since in the absolute majority of cases, the electric drive of all drive units of the mechanism is the same, optimization should be carried out on a uniform discrete grid. Recently bionic intellectual methods are widely used to solve the problem of discrete optimization. One of such methods is the ACO algorithm (ant colony optimization), which is successfully used, in particular, for solving the traveling salesman problem [6,7]. Let us show the possibility of using the ACO algorithm to find a solution to the problem of planning the path of movement of the output link of the mechanism of a parallel structure in order to minimize the duration of such move.

Let us consider the problem of optimizing the trajectory by the example of a planar 3-RPR robot. This robot (Figure 1) consists of three rods of variable length, pivotally attached to a fixed base at the vertices of an equilateral triangle. The other end of each rod is hinged at the vertices of an equilateral triangle on a moving platform, which allows it to move in the plane of the base. The input coordinates of the mechanism are the lengths of the rods \( (l_1, l_2, l_3) \). The output coordinates are the position of the geometric center of the moving platform in Cartesian coordinates \((x, y)\) associated with the center of the base of the mechanism and its angle of rotation \((\varphi)\) relative to the axis perpendicular to the base plane.

![Figure 1. The planar 3RPR mechanism](image)

As shown in [8], the mechanism workspace is defined by:

1. its geometrical dimensions, the range of lengths of drive links:
\[
\begin{align*}
L_{\text{min}} \leq L_{1,i} &= \sqrt{\left( x_i + \frac{r}{2} (\sin \varphi_i - \sqrt{3} \cos \varphi_i) + \frac{\sqrt{3}}{2} R \right)^2 + \left( y_i - \frac{r}{2} (\sqrt{3} \sin \varphi_i + \cos \varphi_i) + \frac{R}{2} \right)^2} \leq L_{\text{max}}, \\
L_{\text{min}} \leq L_{2,i} &= \sqrt{\left( x_i + \frac{r}{2} (\sin \varphi_i + \sqrt{3} \cos \varphi_i) - \frac{\sqrt{3}}{2} R \right)^2 + \left( y_i + \frac{r}{2} (\sqrt{3} \sin \varphi_i - \cos \varphi_i) + \frac{R}{2} \right)^2} \leq L_{\text{max}}, \\
L_{\text{min}} \leq L_{3,i} &= \sqrt{(x_i - r \sin \varphi_i)^2 + (y_i + r \cos \varphi_i - R)^2} \leq L_{\text{max}},
\end{align*}
\]

where \((l_{1,i}, l_{2,i}, l_{3,i})\) and \((x_i, y_i, \varphi_i)\) are respectively the input and output coordinates of an arbitrary point of the trajectory, \(L_{\text{min}}\) and \(L_{\text{max}}\) – the minimum and maximum lengths of the drive units, respectively, taking into account the required distance from the boundaries of the working space, \(R\) and \(r\) are the radii of the circles described near the fixed base (mounting points of the rods) and the movable platform, respectively;

2. the area of the sign of the Jacobian:
\[
J(x_i, y_i, \varphi_i) = 12 \sqrt{3} R \ r \sin \varphi_i \left( R^2 - 2 R r \cos \varphi_i + r^2 - x_i^2 - y_i^2 \right) \geq J_{\text{min}},
\]
where \(J_{\text{min}}\) is the minimum allowable value of the Jacobian, determined by the required distance from the points of the singularity;

The above limitations should be considered when solving an optimization problem.

3. Description of the algorithm.

1. Enter the coordinates of the initial \((L1, L2, L3)\) and end \((L1k, L2k, L3k)\) points, enter array A, which contains data about the working space, allowed and forbidden to visit nodes of the discrete grid, and array B, which stores information about the probability choosing different directions of elementary movement (pitch) for each node of the discrete grid, the so-called pheromone trace — a weighted history of successful motion paths, i.e. paths leading to the end point.

2. The variable \(m1\), in which the minimum trajectory length will be stored within the current generation of agents, is entered into a deliberately large value \(M + 1\).

3. The \(B1\) array stores the current state of the \(B\) array. The \(B1\) array is used to form the trajectories of the current generation of agents.

4. The process of pheromone evaporation is modeled by proportionally decreasing all values contained in the \(B\) array:
\[
B_{1,i}^* = \frac{b}{B_{1,i}} \quad \text{for} \quad 0 < b < 1
\]
where \(0 < b < 1\) is the coefficient determining the evaporation rate.

5. Reset the number of agents in the current generation.

6. The variable \(m1\) is stored in the variable \(m2\), which will store the minimum path length for the entire time the algorithm is running.

7. The value of \(M + 1\) is again entered into the variable \(m1\).

8. An array \(A1\) is created - a copy of array \(A\), which is intended to account for visits to nodes.

9. The cycle is specified by the step number \(m\) from 1 to \(M\), where \(M\) is the maximum allowable path length. At the end of the cycle, go to step 35.

10. The variables \(k\), \(P\), and \(T1\) are initialized.

11. Incremental cycles are organized for each of the coordinates \(l1, l2\) and \(l3\) from -1 to 1. At the end of the cycles, go to step 20.

12. If the increments over all coordinates are zero, the iteration is skipped (go to 11).

13. The \(k\)-counter of the number of the direction of motion increases. If the considered point \((L1 + l1, L2 + l2, L3 + l3)\) coincides with the end point \((L1k, L2k, L3k)\), go to step 30.

14. If, according to array \(A1\), the point in question is not allowed to visit, the value 0 (zero amount of pheromone and zero probability of movement in this direction) is entered into the array \(p\) and the
transition to 11 is made.
15. Otherwise, the amount of pheromone from the B1 array, increased by one, is recorded in the array p, corresponding to the current point and the direction of movement being considered.
16. The calculation of the total amount of pheromone for this site is carried out.
17. The current value of the objective function (1), which determines the degree of closeness of the considered point to the final one, is compared with the lowest one achieved at previous iterations. If necessary, the minimum is updated, the value of k is saved.
18. The transition to the next iteration is underway (step 11).
19. If P does not increase after initialization (no valid steps), go to step 35.
20. The amount of pheromone (and the probability of choice) increases for the direction corresponding to the minimum of the objective function.
21. A pseudo-random number is generated from 0 to R and K is reset.
22. Incremental cycles are organized for each of the coordinates l1, l2, and l3 from -1 to 1. At the end of the cycles, go to step 9.
23. If the increments in all coordinates are zero, the iteration is skipped (go to step 24).
24. The number of the direction of movement increases by 1.
25. The value of R is reduced by p (k). If R ≥ 0, go to step 24.
26. Otherwise, the current point of the trajectory and the selected direction of movement are saved, a step is performed (current node change), a mark is made in the A1 array, the transition to 9 is made.
27. The number of agents is incremented by 1.
28. If the resulting path is not the shortest in the current generation, go to step 34.
29. Updated at least the length of the trajectory in the current generation.
30. If the resulting trajectory is the shortest of all found, it is saved.
31. In the cycle, a pheromone is added to all the edges of the trajectory in the B array in an inverse proportion to the length of the trajectory.
32. If the number of agents in the generation does not exceed the specified, the transition to step 8.
33. If the best trajectory of the current generation is shorter than the best trajectory, go to step 3.
34. The result of the algorithm is the best of the found trajectories and its length.

4. Simulation results
For a visual illustration of the operation of the algorithm, we consider a situation where a change in the length of one of the rods is not required. In this case, the optimization of the trajectory can be carried out on a plane. Let us set the parameters of the mechanism and fix the length of one of the rods: R = 100, r = 25, l1 = 100, l2 = l3 = 10 ÷ 190. In fig. 2 shows the working space of such a mechanism (red area in the middle) in the coordinates (l2, l3) taking into account the size of the workpiece (the purple area in the middle) and the required margin (brown). In addition, the area close to zero is also shown in purple, and the negative values of the Jacobian are shown in yellow.

Figure 2. The working space of the 3RPR-mechanism in coordinates (l2, l3) taking into account the size of the workpiece provided that l1 = 100.
A distinctive feature of the proposed algorithm is an additional increase in the probability of movement in the direction of the end point of movement (minimizing the number of remaining steps). The formula for determining the probability of choosing the direction of movement of an agent provides for its dependence on the length of each step:

\[ p_k = \frac{\tau_k^\alpha \eta_k^\beta}{\sum_{k=1}^{26} \tau_k^\alpha \eta_k^\beta}, \]

where \( \tau_k \) is the number of pheromone on the edge \( k \), \( \eta_k \) is the attractiveness of the edge, the reciprocal of its length, \( \alpha \) and \( \beta \) are the parameters controlling the influence of \( \tau_k \) and \( \eta_k \).

In the absence of the described modification (and at \( P_0 = 0 \)), unnecessarily long trajectories (yellow areas in Fig. 3) appear at the initial stage, reflecting the “aimless wandering” of the agent. The proposed modification thus makes it possible to significantly increase the speed of the algorithm (Fig. 4). The possibility of flexible adjustment of the algorithm by changing the coefficient \( P_0 \) allows us to find a balance between the stochastic and deterministic components of the motion of agents.

**Figure 3.** Examples of the trajectories of motion of agents of the first generation in accordance with the classical algorithm of ACO.

**Figure 4.** Examples of the trajectories of motion of agents of the first generation in accordance with the modified ACO algorithm.

Agents of subsequent generations significantly improve (straighten and shorten) the trajectories found, but process can be accelerated by using a modified discrete local optimization algorithm [9].

5. Conclusion

In the course of the experiments, it was possible to obtain an empirical dependence of the duration of the discrete local optimization of the trajectory on its initial length (Figure 5). As expected, the dependence is quadratic. Thus, for planning the trajectory of the robot of a parallel structure, the modified ACO algorithm proposed in this work can be successfully applied. At the same time, in order
to save time, the trajectories of movement obtained during its operation can be subjected to discrete local optimization based on the algorithm proposed earlier.

![Figure 5](image)

**Figure 5.** Dependence of duration (in seconds) of discrete local optimization of a trajectory on its initial length (number of steps).

The analysis showed that the use of the modified ACO algorithm allows one to shorten the length of the original trajectories 15 times on average, and the time required to process them using the modified discrete local optimization algorithm - more than 20 times. The result of the work of the algorithms is a set of tasks reflecting the change in each of the input coordinates of the mechanism over time. These tasks can be directly used in the operation of the robot control system.

6. **Acknowledgments**

This work was supported by the Russian Science Foundation, agreement number 16-19-00148.

**References**

[1] Pshihopov V H 2014 Intellectual planning of the trajectories of moving objects in environments with obstacles p 300

[2] Vukobratovich M, Karan B 1995 *Theory and Control Systems* 1 205-220.

[3] Voynov V V, Yushchenko A S 2009 *Extreme Robotics. Nano, Micro and Marobot: Proceedings of the International Conference* 2009 115-116

[4] Su T, Cheng L, Wang Y, Liang X, Zheng J, Zhang H 2018 *IEEE Access* 6 28530-28539

[5] Farajtabar, M., Daniali, H., & Varedi, S. 2017. *Robotica*, 35 (2) 241-253

[6] Dorigo M, Stützle T 2003 *Handbook of Metaheuristics*. International Series in Operations Research & Management Science 272 250-285

[7] Mavrovouniotis M, Müller F M, Yang S 2017 *IEEE Transactions on Cybernetics* 47(7) 1743-1756

[8] Rybak L, Khalapyan S, Kholoshevskaya L, Malyshev D 2018 *AIP Conference Proceedings* 2029 020066

[9] Virabyan L G, Khalapyan S Y, Kuzmina V S 2018 *Bulletin of BSTU named after V.G. Shukhov* 9 106–113