Adv Syst Sci Appl 2018; **1**; 92-101
Published online at http://ijassa.ipu.ru.

A two-stage algorithm for generating a set of Pareto-optimal trajectories of an object

Elena L. Kulida [1]

[1)] *Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia*

E-mail: lenak@ipu.ru

Abstract: A two-stage algorithm is proposed for generating trajectories and parameters of the object's motion based on searching for a set of Pareto-optimal paths from the initial vertex to the terminal vertex. The features of graph construction in the first and second stages of the algorithm are described. At the first stage, the set of vertices of the graph uniformly cover the area of the object's motion, and the edges connect only the nearest neighboring vertices. At the second stage, the set of vertices of the graph includes only those vertices through which the paths constructed in the first stage pass. For this set of vertices, a complete graph is constructed: the edges connect each vertex to all the others. An algorithm for constructing a set of Pareto-optimal characteristics of graph paths is described. The two-stage approach allows to significantly reduce the calculation time in comparison with the application of the described algorithm for a complete graph in one stage. Two variants of the algorithm are considered. The first algorithm requires a longer calculation time, but allows to obtain more trajectories that are diverse. In particular, it is possible to obtain trajectories of different classes, for example, avoiding obstacles from different sides, etc. For the case when time for calculations is not enough, an abridged algorithm is proposed. The results of computational experiments are presented.

Keywords: approximation of trajectories by graph paths, a set of Pareto-optimal characteristics, search for a path in a graph, algorithm for generation of trajectories.

1. INTRODUCTION

Algorithms for generating trajectories that satisfy specified conditions are needed to effectively control the movement of an object moving in a conflict environment. It is necessary that flight management systems are able to comply with 4D trajectories, which needs to be done in computationally efficient ways due to the limited computational resources available [3, 5].

In [6], analytical and discrete approaches are considered to optimize the trajectory of an aircraft in a dangerous environment. The discrete approach allows solving the problem in case of the detection risk of an aircraft by several radars. However, the use of discrete optimization methods requires very large computational costs to generate optimal trajectories, especially if optimization by several criteria is required. In this paper, we propose a decomposition of the optimization process into two stages, which allows to reduce the time required for solving the problem.

In [4], an algorithm for controlling the trajectory and parameters of the motion of an object in a conflict environment is given. The risk of detecting an object and the possibility of collision with the earth's surface are taken into account. Optimization is carried out in accordance with specified quality criterion of optimizing the trajectory and traffic parameters with a restriction on the time of motion. For this purpose, the specified quality criterion and the criterion of the minimum time of completion of a route are considered. A graph is formed, and its paths approximate all possible trajectories of the object's motion in the given region. The graph must include vertices that cover the region of motion tightly enough to approximate the trajectories with sufficient accuracy. It is desirable that each vertex is connected by edges to all other vertices to move to an arbitrary direction if the motion along such an edge does not violate restrictions associated with the terrain [1]. Two-component characteristics are set on the edge of the graph. The characteristic consist the values of two optimization criteria. Sets of Pareto-optimal characteristics of paths from a given initial vertex are constructed for the vertices of a graph. However the calculation of the set of Pareto-optimal characteristics of the vertices of such a graph requires time-consuming unsatisfactory large for the practical use of the algorithm.

The paper proposes an algorithm to solve this problem more effectively in two stages.

2. FORMULATION OF THE PROBLEM

The area of motion of the object M is set on the map. When the object moves in this area conflicts with the relief may occur. A known height matrix of the relief $R(x, y)\ \forall(x, y) \in M$ for the area is assumed. We consider the set Z of all possible trajectories of the object motion from the initial point $A \in M$ with coordinates (x_A, y_A) and height h_A to the endpoint $B \in M$ with coordinates (x_B, y_B) and height h_B.

$$Z = \left\{ \begin{array}{c} Z(t) = \big(x(t), y(t), h(t), v(t)\big), \qquad t = \overline{0, T}, \qquad \big(x(t), y(t)\big) \in M, \\ x(0) = x_A, \qquad y(0) = y_A, \qquad h(0) = h_A \\ x(T) = x_B, \qquad y(T) = y_B, \qquad h(T) = h_B \\ h(t) \geq R\big(x(t), y(t)\big) + \Delta h, \qquad t = \overline{0, T}. \end{array} \right\},$$

where $x(t), y(t)$ are the coordinates of the object, $h(t)$ is the height of the object, $v(t)$ is the speed of the object's motion at time t, and T is the trajectory Z passing time. The trajectory should not have conflicts with the relief, at any point of the trajectory the height of the object above the relief must be greater than the specified value Δh.

The functional of the cost (the risk) of passing the point $(x(t), y(t), h(t))$ with the speed $v(t)$ is

$$F(t) = F\big(x(t), y(t), h(t), v(t)\big).$$

The cost of the trajectory Z is

$$C(Z) = \int_0^T F\big(x(t), y(t), h(t), v(t)\big)dt$$

The task is to generate trajectories of the object from point A to point B of the Pareto-optimal by two criteria: the minimum cost and the minimum time of passing of the route.

The problem cannot be solved analytically due to the complexity of the cost functional and the need to take into account the terrain. Discrete optimization methods are used to solve the problem. A graph is constructed to approximate the trajectories from the set Z, and the problem of finding the Pareto-optimal trajectories of the graph by two specified optimality criteria is solved.

3. DISCRETE OPTIMIZATION METHOD FOR SOLVING THE PROBLEM

3.1. The construction of the graph in the first stage of optimization

We proceed to a discrete solution to obtain an approximate solution from a continuous problem. The solution is sought in the form of a piecewise linear trajectory with piecewise constant speed and height of motion (speed and height is constant on segments of the trajectory and can be changed when moving from one segment to another). We consider a finite set of heights (depths) of the motion $H = \{h_0, h_2, \ldots, h_{l-1}\}$, l is the number of heights, and a finite set of velocities of the object $V = \{v_0, v_2, \ldots, v_{m-1}\}$, m – the number of considered speeds of movement of the object.

The trajectories of the object's motion are approximated by the paths of the graph $G = (N, E)$, N is the set of vertices of the graph, and E is the set of edges of the graph. The path is sought in the form of a sequence of edges $z_i = [(x_i, y_i, h_i), (x_{i+1}, y_{i+1}, h_{i+1}), v_i]$, $i = \overline{0, N-1}$, $(x_0, y_0, h_0) = (x_A, y_A, h_A)$, $(x_{N,} y_N, h_N) = (x_B, y_B, h_B)$where $v_i \in V$ is the speed of motion along the edge z_i. The edge z_i either lies in the horizontal surface at height h_i, then $h_i = h_{i+1}$, or is intended to go to another horizontal surface, then $x_i = x_{i+1}$ & $y_i = y_{i+1}$.

The given area of the object's motion is covered by a uniform grid of points at a distance d from each other to construct the set of vertices of a graph.

A separate grid layer is created for each of the l different heights of the object. The vertices are projected onto all layers. If there is an obstacle to the relief: $h(t) < R(x,y) + \Delta h$, the corresponding vertex is excluded from the set of vertices of the graph *N*. In addition the points *A* and *B* are included in the set *N*.

It is assumed that the constant velocity is ascribed to the edge of the graph. This speed determines the cost and time of movement along the edge. *m* edges connect each vertex from the set N with adjacent (nearest) vertices in the horizontal layer. One of the m speeds corresponds to one of these m edges. In addition, each vertex is connected by edges to vertices in adjacent horizontal layers; these vertices are assigned vertical velocities (ascending or descending) of the object.

The construction of the graph in the first stage of optimization is illustrated in fig. 3.1. In this figure in the horizontal layers the vertices form a square grid, each vertex is connected by edges with four neighboring vertices in the horizontal layer and with vertices under and above it in adjacent horizontal layers.

The number of vertices of the obtained graph can be approximately estimated by the number $w \approx \frac{S}{d} l$, where S is the area of the region of motion, d is the distance between neighboring vertices in the layer, and l is the number of layers. The number of edges $r \approx (4m + 2)w$. These are estimates from above, because some vertices and edges are excluded from consideration due to the obstacles of the relief

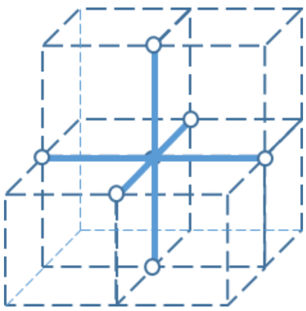

Fig. 3.1. Graph construction in the first stage of optimization

3.2. The algorithm for constructing the set of Pareto-optimal characteristics

Each edge is assigned a characteristic in the resulting graph.

Definition 3.1:

The characteristic of the edge z_i at the speed of motion v_i is the two-component vector (c_i, t_i), where c_i is the cost of passing the edge z_i with the speed v_i, t_i is the time of the z_i passing with the speed v_i,

$$t_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (h_i - h_{i+1})^2} / v_i.$$

Definition 3.2:

The characteristic of the trajectory Z is the two-component vector $d_Z = (C_Z, T_Z)$, *where* C_Z *is the cost of the trajectory Z,* T_Z *is the trajectory Z passing time, defined by the formulas:*

$$C_Z = \sum_{i=0}^{N-1} z_i, \quad T_Z = \sum_{i=0}^{N-1} t_i.$$

The problem is to construct in the graph the set of paths from vertex A to vertex B, Pareto-optimal by two criteria: $\min_Z C_Z$ and $\min_Z T_Z$.

An algorithm is used to construct the sets of Pareto-optimal characteristics of paths from the initial vertex to other vertices of the graph to solve this problem. Then the paths corresponding to the Pareto-optimal characteristics obtained are constructed for the final vertex.

Since there are two criteria, each vertex is associated with a set of incomparable two-component characteristics.

Consider two arbitrary characteristics $\mu_1 = (c_1, t_1)$ and $\mu_2 = (c_2, t_2)$.

Definition 3.3:

The characteristic μ_1 *and* μ_2 *are equal:* $\mu_1 = \mu_2$, *if* $(c_1 = c_2 \& t_1 = t_2)$.

Definition 3.4:

The characteristic μ_1 *is smaller than the characteristic* μ_2*:* $\mu_1 < \mu_2$, *if* $(c_1 < c_2)||(c_1 = c_2 \& t_1 < t_2)$.

Definition 3.5:

The characteristics $\mu_1 = (c_1, t_1)$ *and* $\mu_2 = (c_2, t_2)$ *are incomparable if* $(c_1 < c_2 \; \& \, t_1 > t_2)||(c_1 > c_2 \; \& \; t_1 < t_2)$.

Definition 3.6:

We call the set Q of path characteristics from the vertex A to the vertex B Pareto-optimal if all the characteristics of this set are pairwise incomparable and for the characteristic μ_Z *of an arbitrary path Z from the vertex A to the vertex B* $\exists \mu \in \Omega$: $\mu \leq \mu_Z$.

The algorithm for constructing the set of Pareto-optimal characteristics

Notation:

D_n *is an intermediate set of Pareto-optimal characteristics of paths from vertex A to vertex n;*

P_n *is the final set of Pareto-optimal characteristics of paths from vertex A to vertex n;*

Q is the set of vertices n for which the set $D_n \neq \emptyset$.

Step 1:

The initial vertex A is considered. The characteristic (0,0) is stored in the set D_{A} , the vertex A is stored in the set Q.

Cycle:

A cyclic process is performed until the set Q is not empty.

An iteration of the cycle consists of two stages.

The first stage is to select the vertex for processing:

Among all vertices $n \in Q$, we look for a vertex whose set D_n contains the minimal characteristic $\mu_n = (c_n, t_n)$. The characteristic μ_n is excluded from the set D_n and is included in the set P_n. If after this the set D_n becomes empty, then the vertex *n* is excluded from the set Q.

The second stage is processing the selected vertex:

All vertices connected by edges with vertex n are considered. For each vertex *m*, the characteristic $\mu_m = (c_n + c_{nm}, t_n + t_{nm})$, where $\mu_n = (c_n, t_n)$ is the minimal characteristic in the set D_n, $\mu_{nm} = (c_{nm}, t_{nm})$ is the characteristic of the edge that connects the vertices *n* and *m*.

If in the sets D_m and P_m there is no characteristic smaller than μ_m, then μ_m is added to the set D_m. All the characteristics $\mu \in D_m$: $\mu_m < \mu$, are removed from D_m – this ensures the Pareto optimality of the set D_m. If before the insertion of μ_m in D_m this set was empty, then the vertex *m* is stored in Q.

After the completion of the algorithm, the set P_B is the set Pareto-optimal characteristics of paths from vertex *A* to vertex *B*.

3.3. The algorithm for generation of Pareto-optimal trajectories

The goal of the first stage of optimization is to construct a set *N* of vertices of the graph for the second stage of optimization.

The first stage of optimization

Step 1:

The construction of a graph in the form of a uniform grid. Calculation of the characteristics of edges of a graph.

Step 2:

Construction of sets of Pareto-optimal path characteristics from the vertex A to vertices of the graph.

Step 3:

Construction of the set of Pareto-optimal paths from vertex *A* to vertex *B* based on the constructed set P_B. We denote by N^* all the vertices through which the paths corresponding to the set P_B pass.

Examples of the Pareto-optimal trajectories generated in the first stage are shown in figures 3.2 – 3.4.

Dangerous areas for the movement of the object are represented in the figures with filled rectangles. The color intensity characterizes the degree of danger of the zone. Optimization according to two criteria allows obtaining different trajectories, including bypassing dangerous zones from different directions or passing through dangerous zones, as this degrades the quality of the trajectory by one criterion, but improves by the other – it allows reducing the trajectory transit time.

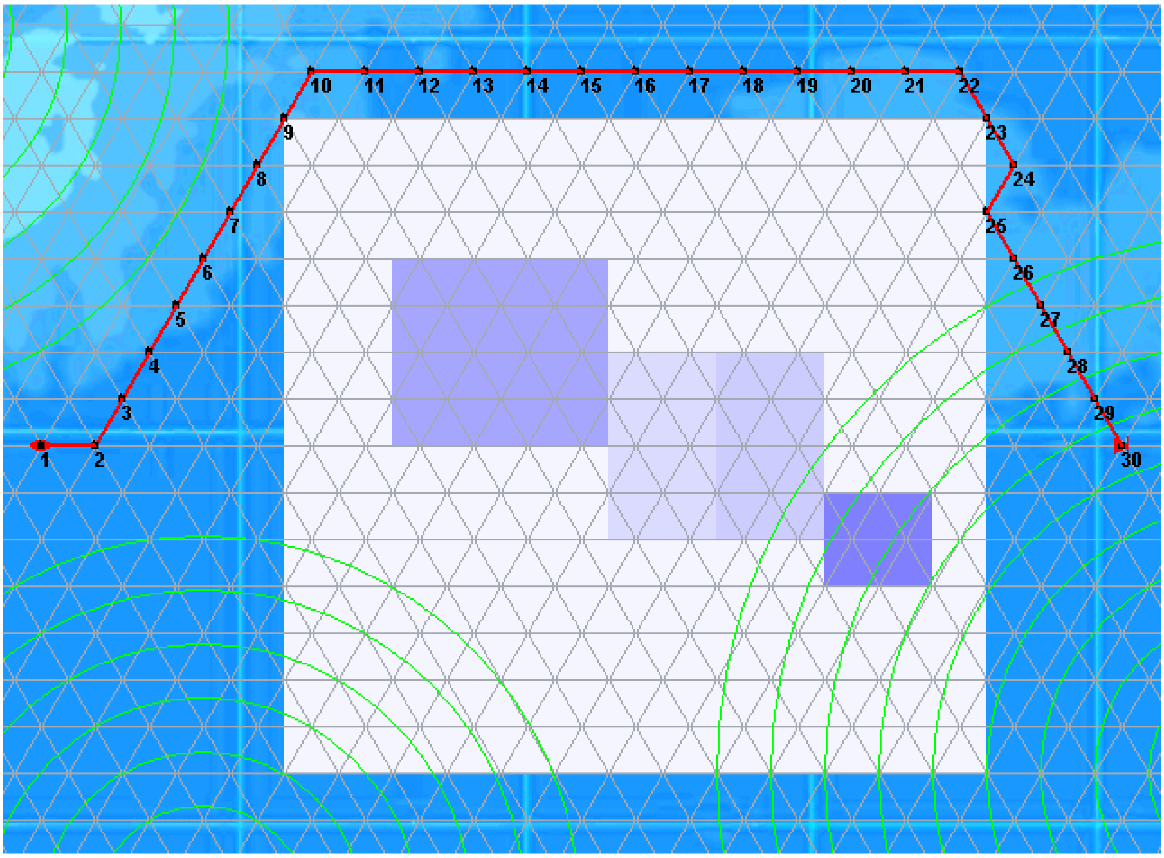

Fig. 3.2. Trajectory bypasses dangerous areas

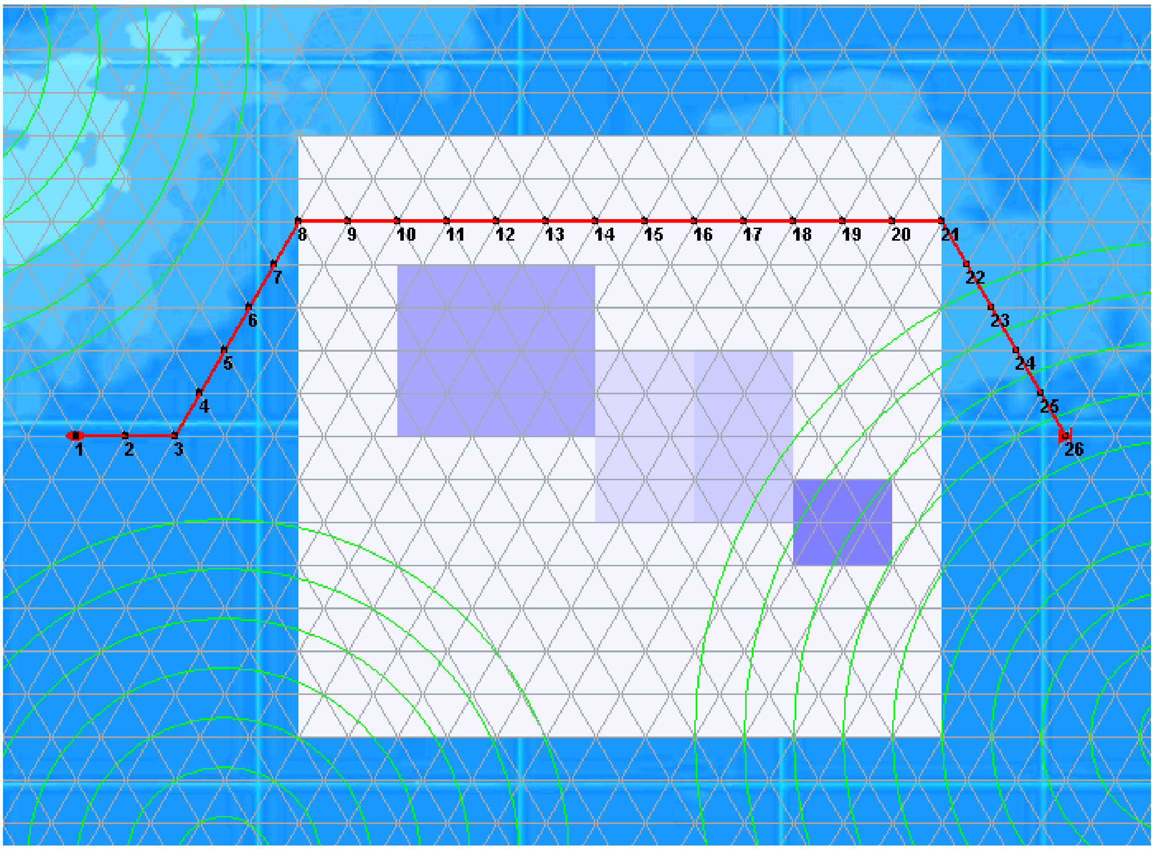

Fig. 3.3. This trajectory is more dangerous but the travel time is less

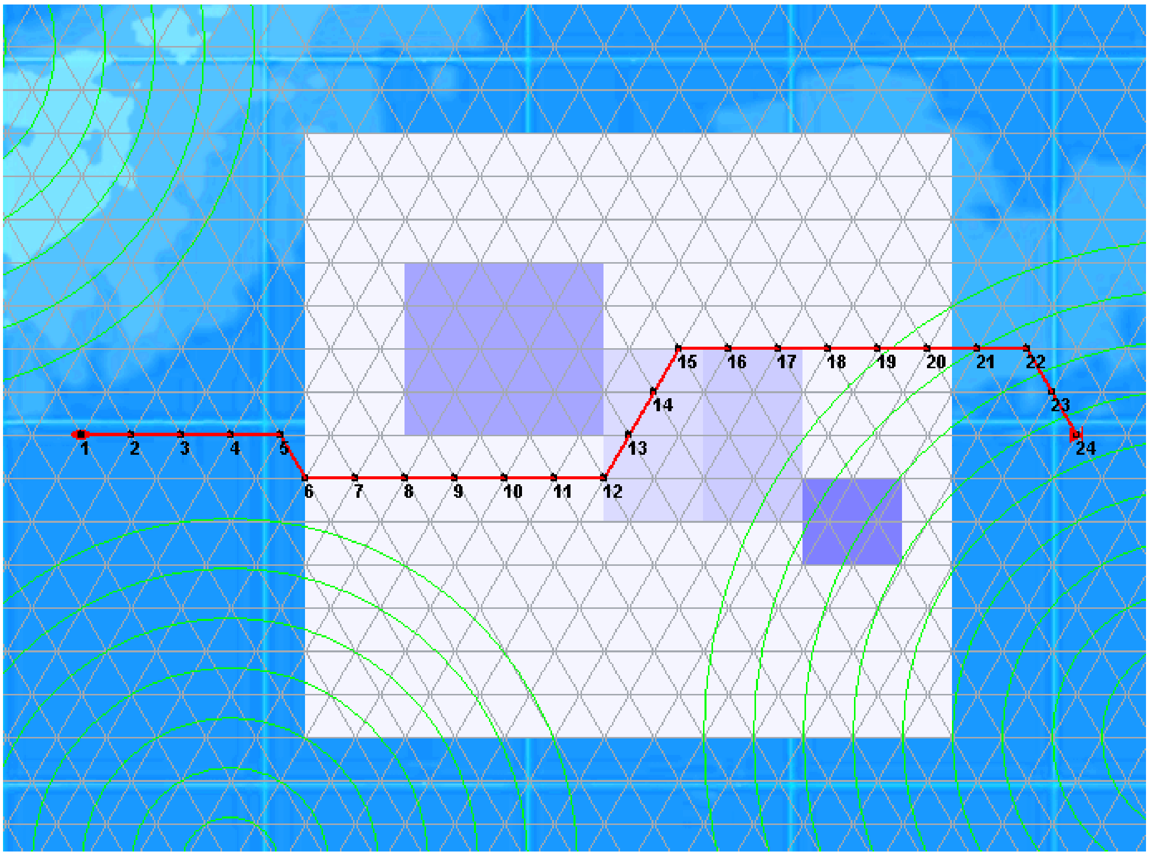

Fig. 3.4. The trajectory is passing through dangerous zones but the travel time is less

The second stage of optimization

Step 4:

Construction of a graph for the second stage of optimization.

N^* is the set of vertices through which the Pareto-optimal paths constructed in the first stage pass. Each of the vertices of the set N^* is projected onto all l layers of the grid to construct the set N of vertices of the graph for the second stage of optimization:

$$N = \begin{Bmatrix} N_1 \\ \dots \\ N_l \end{Bmatrix} = \begin{Bmatrix} (x_0, y_0, h_0), (x_1, y_1, h_0), \dots, (x_k, y_k, h_0) \\ \dots \\ (x_0, y_0, h_{l-1}), (x_1, y_1, h_{l-1}), \dots, (x_k, y_k, h_{l-1}) \end{Bmatrix}.$$

A complete graph is constructed in each horizontal layer in the second stage of optimization; each of the vertices $N_i, i = 1, \dots, l$ of the layer is joined by edges with all the other vertices of this layer to construct real trajectories of the object's motion. The vertices are connected by edges between the horizontal layers one under the other. A characteristic is calculated for each edge in the resulting graph.

In the first step, the set of vertices of the graph includes vertices evenly covering the area of motion, and the edges of the graph connect only to the nearest neighboring vertices.

At the second stage, the set of vertices of the graph contains only those vertices through which the paths built at the first stage pass. Each of the vertices is connected by edges to all other vertices of the graph. A set of Pareto-optimal paths is constructed in the complete graph. This two-step approach allows to significantly reduce the computation time in comparison with the use of the algorithm for constructing a set of Pareto-optimal paths in the initial graph.

Step 5:

Construction of sets of Pareto-optimal characteristics of paths from vertex A for the vertices of the graph.

Step 6:

An analysis of the constructed set of Pareto-optimal characteristics; the choice of characteristics satisfying the requirements and constraints [2]; the construction of appropriate paths.

4. SIMULATION RESULTS

A series of experiments was performed to evaluate the efficiency and accuracy of the proposed algorithm using the example of generating object trajectories moving in a conflict environment, where it is detected by means of detection of different nature.

Below are the results characterizing the use of the proposed optimization algorithm in one of such experiments.

As can be seen from the Table 4.1, a decrease in the distance *d* between vertices in the construction of the grid at the first stage of optimization predictably allows to build more diverse trajectories and substantially improve the accuracy of the solution, but the calculation time increases dramatically. The evaluation of the trajectory is the probability of not detecting an object during the trajectory, so the evaluation lies in the range [0,1]. The best trajectory has greater evaluation.

Table 4.1. Results for different length of edge of graph

Length of edge of graph, km	Evaluation of the trajectory	Travel time, hours	Calculation time hour:min:sec	Number of trajectories
32	0,4078	28,29	00:01:50	254
16	0,5612	21,05	00:26:23	1140
8	0,6148	18,4	06:20:21	2639

When analyzing the set of generated trajectories, it turns out that the parameters of incomparable trajectory characteristics can differ by negligibly small values. In order to reduce the number of generated trajectories and to shorten the calculation time, the following method was used: during the calculation, the values of the parameters of the segments characteristics were rounded up to four decimal places. The results of the computational experiment are shown in the Table 4.2.

Table 4.2. Results for different length of edge of graph with/without rounding

	Length of edge of graph, km	Evaluation of the trajectory	Travel time, hours	Calculation time hour:min:sec	Number of trajectories
Without rounding	32	0,4078	28,29	00:01:50	254
With rounding	32	0,4078	28,29	00:01:12	245
Without rounding	16	0,5612	21,05	00:26:23	1140
With rounding	16	0,5613	21,05	00:19:44	1064
Without rounding	8	0,6148	18,4	06:20:21	2639
With rounding	8	0,6149	18,4	04:04:00	2420

As follows from the table, the characteristics of the generated trajectories did not change to two decimal places, and the calculation time decreased.

In order to reduce the calculation time significantly, one can use a shorter algorithm.

The algorithm for generating Pareto-optimal trajectories can be reduced as follows:

Step 2: instead of constructing sets of Pareto-optimal path characteristics, calculate the length of the optimal path by the main criterion for each level of the graph, for example using the Dijkstra algorithm.

Step 3: For each layer of the graph construct a path from vertex A to vertex B, which is optimal by the main criterion, i.e. with a minimal cost. We denote by N* all the vertices through which constructed paths pass.

The other steps of the algorithm remain unchanged.

The results of the comparison of the algorithm 1 (source) and the algorithm 2 (reduced) are presented in the Table 4.3.

Table 4.3. The results of the comparison of the algorithm 1 and the algorithm 2

	Length of edge of graph, km	Evaluation of the trajectory	Travel time, hours	Calculation time hour:min:sec	Tumber of trajectories
Algorithm 1	32	0,4078	28,29	00:01:12	245
Algorithm 2	32	0,332	27,79	00:00:14	65
Algorithm 1	16	0,5613	21,05	00:19:44	1064
Algorithm 2	16	0,5291	20,67	00:01:33	262
Algorithm 1	8	0,6149	18,4	04:04:00	2420
Algorithm 2	8	0,5968	19,16	00:12:16	553

5.CONCLUSION

The article presents two algorithms for constructing Pareto-optimal trajectories of object motion. Optimization by the two criteria makes it possible to obtain trajectories that are optimal by the main criterion and satisfy a given time limit.

The results of numerical experiments are presented.

The following result is obtained by comparing the two proposed algorithms. In algorithm 2, the number of generated trajectories is reduced significantly, while the accuracy of the best solution decreases insignificantly, and the calculation time decreases very significantly. As can be seen from the Table 3, in the case where the edge of the graph is 8 km, the calculation time has decreased by more than 20 times!

REFERENCES

[1] Bazhenov, S.G., Egorov, N.A., Kulida, E.L., & Lebedev, V.G. (2016). Control of aircraft trajectory and speed to avoid terrain and traffic conflicts during approach maneuvering, *Automation and Remote Control* 77(10), 1827-1837.

[2] Bazhenov, S.G., Korolyov, V.S., Kulida, E.L., & Lebedev, V.G. (2014). Simulation of on-board model of airliner to evaluate capability of trajectories and flight safety. *29th Congress of the International Council of the Aeronautical Sciences, ICAS 2014*.

[3] Diaz Y., Lee S., Egerstedt M., & Young S. (2013) Optimal Trajectory Generation for Next Generation Flight Management Systems. *32nd Digital Avionics Systems Conference*.

[4] Dobrovidov, A.V., Kulida, E.L., & Rudko, I.M. (2015) Path optimization for a moving object in an anisotropic environment using the probabilistic criterion in the passive sonar mode. *Automation and Remote Control*, 76(7), 1271-1281.

[5] Hehn, M., & Andrea R.D. (2011) Quadrocopter trajectory generation and control. *18th IFAC World Congress*, vol. 18, 1485–1491.

[6] Zabarankin, M., Uryasev, S. and Pardalos P. (2002). *Optimal Risk Path Algorithms.* Cooperative Control and Optimization (R. Murphey and P. Pardalos ed.), *Kluwer Academic Publishers, Dordrecht*, vol. 66, 271–303.

