Stanisław KONATOWSKI, Piotr PAWŁOWSKI

Wojskowa Akademia Techniczna, Wydział Elektroniki, Instytut Radioelektroniki

doi:10.15199/48.2019.07.24

Application of the ACO algorithm for UAV path planning

Abstract. *The ACO (Ant Colony Optimization) algorithm is a bio-inspired metaheuristic used to optimize problems or functions described by graphs, sequences of events, or queues of tasks. It is used, among a variety of other purposes, when routing Internet network packets, determining the shortest routes between designated points (traveling salesman's problem), for the time and cost optimization of production, or setting public transport stops. In the article, the ACO algorithm was used to autonomously construct the optimal route for an unmanned aerial vehicle (UAV). The algorithm establishes the spatial orientation of the UAV, indicating the direction of its transition for each intermediate waypoint. The results of the simulations show the trajectory of the UAV depending on the selected weighting factors, determining the priority of avoiding detected hazards or choosing the shortest path. The quality of each variant is evaluated numerically by the calculated fitness function, the value of which is the sum of the costs of the transition to each intermediate route point. The effect of the algorithm is a set of executable trajectory variants, of which the one with the best fitness value is selected.*

Streszczenie. *Algorytm ACO (ang. Ant Colony Optimization) jest bio-inspirowaną metaheurystyką, wykorzystywaną do optymalizacji problemów lub funkcji opisywanych za pomocą grafów, sekwencji zdarzeń, czy też kolejki zadań. Znajduje on zastosowanie m.in. przy trasowaniu pakietów sieci internetowych, wyznaczaniu najkrótszych tras między wyznaczonymi punktami (problem komiwojażera), optymalizacji czasu i kosztu produkcji, czy też ustalaniu przystanków transportu publicznego. W artykule, algorytm ACO został wykorzystany do autonomicznego wyznaczenia optymalnej trasy dla bezpilotowego statku powietrznego (BSP). Algorytm ustala orientację przestrzenną BSP, determinującą kierunek jego przemieszczenia dla każdego pośredniego punktu docelowego. Wyniki przeprowadzonych symulacji przedstawiają trajektorię BSP w zależności od dobranych współczynników wagowych, określających priorytet ominięcia wykrytych zagrożeń lub wybrania najkrótszej drogi. Jakość każdego wariantu jest określana liczbowo poprzez ustaloną funkcję dopasowania, której wartość stanowi suma kosztów przejścia do każdego pośredniego punktu trasy. Efektem działania algorytmu jest zbiór wykonywalnych wariantów trajektorii, z których wybrany zostaje ten o najlepszej wartości dopasowania* ***[Zastosowaniealgorytmu ACO do wyznaczania trasy BSP]***

Keywords: path planning, UAV, ACO algorithm.
Słowa kluczowe: planowanie trasy, BSP, algorytm mrówkowy.

Introduction

The ACO (Ant Colony Optimization) algorithm is a bio-inspired metaheuristic algorithm developed in 1996 by Marco Dorigo [1]. His research on modeling the collective intelligence of ant swarms began in 1992 as part of his PhD thesis [2].

The ACO algorithm imitates the behavior of ants searching for the shortest path to a food source. Each ant chooses a path for itself, however, it is guided by the amount of perceptible pheromones left by other preceding individuals. When following this path, an ant leaves a certain amount of additional pheromones, making the intensification of the trail greater for the shortest route. Pheromones evaporate during the whole process, thanks to which longer paths with fainter traces are forgotten, and ants are more likely to choose those shorter, more perceptible ones as a consequence (Figure 1). In the initial phase of the search, when the amount of traces left are still small, the ants tend to spread, creating at the same time a set of possible variants, eventually leading to the most effective solution.

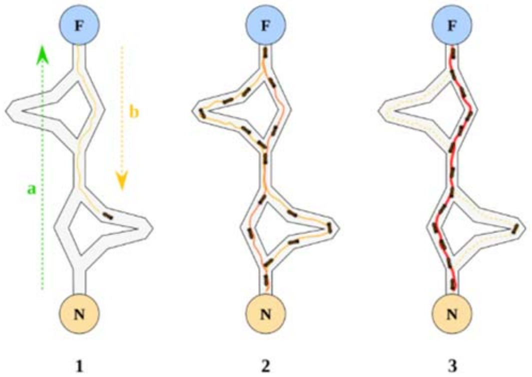

Fig. 1. The principle of choosing the shortest path by ants, where: 1 – random path search, 2 – variant construction, 3 – pheromone accumulation on the best solution and a – movement direction, b – pheromone distribution, N – nest (start), F – food source (target)

The popularity and usefulness of metaheuristic ACO is demonstrated by its wide range of applications such as Resource-Constrained Project Scheduling [3], exploration and data classification [4,11,12], design of efficient computer networks [5], and the allocation and reconfiguration of photovoltaic energy systems [6].

In the proposed solution, the ACO algorithm was used to solve the problem of optimal BSP route planning in three-dimensional space.

ACO algorithm

In the general case, the ACO procedure [7] consists of the selection by ants of subsequent nodes from the n-element set $N = \{u_1, u_2, ..., u_n\}$ and $\tau(t) = \{\tau_1, \tau_2, \dots, \tau_n\}$ with the probability specified by the formula

$$p_j^i(t) = \begin{cases} \dfrac{[\tau_j(t)]^\alpha [\eta_j]^\beta}{\sum_{k \in K}\{[\tau_k(t)]^\alpha [\eta_k]^\beta\}}, & j \in K \\ 0, & j \notin K \end{cases} \tag{1}$$

where $p^i_j(t)$ is the probability of transition to node j by the ant i at time t, which in this case will mean the iteration number of the algorithm. The symbol τ_j represents (numerically) the amount of pheromones left in the u_j node, while η_j is the heuristic factor described by the expression

$$\eta_j = \frac{1}{J_j}, \tag{2}$$

where J_j is the value of the fitness function (transition cost) to node j. The coefficients α and β determine the effect of the pheromone and the heuristic coefficient respectively on the process of selecting the next node. The probability is calculated only for nodes from the set K, which contains all nodes from the set N, for which the transition is not prohibited in any way, thus $K \in N$.

Pheromone trail left on nodes

The principle of the ACO algorithm is based on the traces of pheromones left by each ant after the solution has been constructed (determining the road variant). The value of the pheromones left in node j by all ants after the completion of a given iteration is described by the equation

$$\tau_j(t+1) = \tau_j(t)\cdot\left(1 - \rho\right)+\textstyle\sum_{i=1}^{a} \Delta\tau_i(t), \tag{3}$$

where $\Delta\tau_j$ is the amount of pheromones left in node j by the ant i in the given iteration t, where a is the number of all ants. This value is described by

$$\Delta\tau_i(t)=\frac{Q}{J_{ci,t}}, \tag{4}$$

where Q is a constant value, determining the amount of pheromones that each ant leaves, regardless of the length and cost of its route, while $J_{ci,t}$ is the total cost of the route covered by the ant i in iteration t.

The expression (1-ρ) reflects the process of pheromone evaporation. After each iteration, the value of pheromones found in each node is reduced by the value of ρ (expressed as a percentage), thanks to which weaker solutions - those burdened with higher costs - are forgotten.

Single ant model

Each ant has the ability to remember constructed variants, memorizing the way traveled step by step. Thanks to this, it is possible to reject nodes that have already been visited. This approach prevents the ant from visiting the same node, which can cause the algorithm to loop. The vector describing a single ant in a given iteration ***t*** is presented as follows

$$\boldsymbol{ant}_i^{p,t}=\left[u_i^{p,t},J_i^{p,t}\right], \quad p\in\left(1,\,2,\ldots,\,s_{i,t\text{-}1},s_{i,t}\right) \,, \tag{5}$$

where $u^p{}_i$ is the node selected by the ant in step p, $J^p{}_i$ is the cost of transition to this node, and $s_{i,t}$ is the number of steps made by the ant i in the given iteration t. Every ant is associated with a traveled path saving set

$$\boldsymbol{path}_i^t=\left\{u_i^1,u_i^2,\ldots,u_i^{s_{i,t\text{-}1}},u_i^{s_{i,t}}\right\}. \tag{6}$$

Path planning as an optimization problem for ACO

The problem of path planning can be described as selecting the appropriate sequence of steps leading to reaching the destination point [10]. The process begins when the UAV locates obstacles or hazards. In this situation it is necessary to re-plan the route, considering the amount of fuel remaining (energy) and possible approach to hazardous areas. This ensures shaping the trajectory in such a way that the UAV moves away from the target point when the amount of fuel is insufficient and the level of the given threat is low enough to affect its zone.

Environment and simulation object

The simulated object is the UAV platform presented as a rigid solid with six degrees of freedom 6DoF [8,10,14]. The state vector, characterizing the state of the UAV in space, is presented in the form

$$\boldsymbol{s}(p)=\left[x,\,y,\,z,\,\boldsymbol{Fa}_x,\,\boldsymbol{Fa}_y,\,\boldsymbol{Fa}_z\right], \tag{7}$$

where p is the number of UAV steps, x, y, z are the Cartesian coordinates of the object, $\boldsymbol{Fa}_x$, $\boldsymbol{Fa}_y$, $\boldsymbol{Fa}_z$ are $\boldsymbol{Fa}$-coordinate system coordinates indicating the direction of UAV rotation (its spatial orientation to the Earth's inertial reference system) by Euler angles [8] φ, θ, ψ. These are the angles respectively: roll describing the rotation relative to the axis $0x$, pitch responsible for the rotation with respect to the axis $0y$ and yaw describing the rotation with respect to the axis $0z$.

The element indicating the direction of UAV movement is the transition matrix $\boldsymbol{V}(p)$ in the form

$$\boldsymbol{V}(p)=\begin{bmatrix} V_x & 0 & 0 \\ 0 & V_x & 0 \\ 0 & 0 & V_z \end{bmatrix} \tag{8}$$

with

$$V_x(p)=\begin{bmatrix} V_x \\ 0 \\ 0 \end{bmatrix},\quad V_y(p)=\begin{bmatrix} 0 \\ V_y \\ 0 \end{bmatrix},\quad V_z(p)=\begin{bmatrix} 0 \\ 0 \\ \mathrm{V_z} \end{bmatrix}. \tag{9}$$

These vectors are permanently attached to the center of the UAV coordinate system and rotate with it, and their direction is consistent with the $0x$ axis.

In the ACO procedure a single ant takes on the role of the UAV, so according to the formula (7), the expression (5) should be extended to

$$\boldsymbol{ant}_i^{p,t}=\left[u_i^{p,t},\,J_i^{p,t},\,\boldsymbol{Fa}_{xi}^{p,t},\,\boldsymbol{Fa}_{yi}^{p,t},\,\boldsymbol{Fa}_{zi}^{p,t}\right],\ p\in\left(1,\,2,\ldots,\,s_{i,t\text{-}1},s_{i,t}\right) \tag{10}$$

and associate an additional speed matrix with the ant

$$\boldsymbol{V}_i^{p,t}=\begin{bmatrix} V_{xi}^{p,t} & 0 & 0 \\ 0 & V_{yi}^{p,t} & 0 \\ 0 & 0 & V_{zi}^{p,t} \end{bmatrix}. \tag{11}$$

The spatial orientation of the unmanned aerial vehicle in relation to the inertial Earth reference system is determined by the rotation of the coordinate system permanently associated with the UAV (Figure 2) during the maneuver.

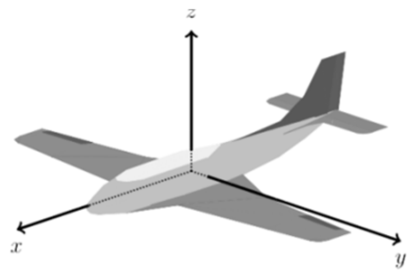

Fig. 2. Coordinate system fixed to an aircraft

Coordinate system transformation is carried out by making three rotations in accordance with the Euler-Rodrigues formula [9]. Each subsequent rotation is made relative to the transformed coordinate system from the previous rotation. Initially, the $\boldsymbol{Fa}$ plane coordinate system is parallel to the inertial system and its orientation can be described by the matrix of three column versors

$$\boldsymbol{Fa}(p)=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

and

$$\boldsymbol{Fa}_x(p)=\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},\ \boldsymbol{Fa}_y(p)=\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ \boldsymbol{Fa}_z(p)=\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{13}$$

which are versors of the $\boldsymbol{Fa}$ coordinate system.

Rotation of the system by an angle γ around the axis defined by the vector w describes the resultant rotation matrix in the form

$$\boldsymbol{w} = \left[w_x,w_y,w_z\right]^T, \tag{14}$$

(15) $$\mathbf{R}(\boldsymbol{w},\gamma) = \begin{bmatrix} w_x^2+(1\text{-}w_x^2)\cos\gamma & \zeta w_x w_y\text{-}w_z\sin\gamma & \zeta w_x w_z+w_y\sin\gamma \\ \zeta w_x w_y+w_z\sin\gamma & w_y^2+(1\text{-}w_y^2)\cos\gamma & \zeta w_y w_z\text{-}w_x\sin\gamma \\ \zeta w_x w_z\text{-}w_y\sin\gamma & \zeta w_y w_z+w_x\sin\gamma & w_z^2+(1\text{-}w_z^2)\cos\gamma \end{bmatrix}$$

for $\zeta = (1\text{-}\cos\gamma)$.
Therefore, the rotation of the UAV in the state of $\boldsymbol{s}(p_1) = [x_1, y_1, z_1, \boldsymbol{Fa}_{x1}, \boldsymbol{Fa}_{y1}, \boldsymbol{Fa}_{z1}]$ by any value of the angles $\Delta\varphi$, $\Delta\theta$, $\Delta\psi$ takes place as follows [16]:
1. Rotation around axis $\boldsymbol{Fa}_{x1}$ by angle $\Delta\varphi$:

(16) $$\boldsymbol{Fa}'_1=\boldsymbol{R}(\boldsymbol{Fa}_{x1},\Delta\varphi)\cdot\boldsymbol{Fa}_1,$$

(17) $$\boldsymbol{V}'_1=\boldsymbol{R}(\boldsymbol{Fa}_{x1},\Delta\varphi)\cdot\boldsymbol{V}_1.$$

2. Rotation around axis $\boldsymbol{F'a}_{y1}$ by angle $\Delta\theta$:

(18) $$\boldsymbol{Fa}''_1=\boldsymbol{R}(\boldsymbol{Fa}'_{y1},\Delta\theta)\cdot\boldsymbol{Fa}'_1,$$

(19) $$\boldsymbol{V}''_1=\boldsymbol{R}(\boldsymbol{Fa}'_{y1},\Delta\theta)\cdot\boldsymbol{V}'_1.$$

3. Rotation around axis $\boldsymbol{F''a}_{z1}$ by angle $\Delta\psi$:

(20) $$\boldsymbol{Fa}'''_1=\boldsymbol{R}(\boldsymbol{Fa}''_{z1},\Delta\psi)\cdot\boldsymbol{Fa}''_1,$$

(21) $$\boldsymbol{V}'''_1=\boldsymbol{R}(\boldsymbol{Fa}''_{z1},\Delta\psi)\cdot\boldsymbol{V}''_1.$$

Finally:

(22) $$\boldsymbol{Fa}_2=\boldsymbol{Fa}'''_1,$$

(23) $$\boldsymbol{V}_2=\boldsymbol{V}'''_1,$$

(24) $$[x_2,y_2,z_2]=[x_1,y_1,z_1]+[V_{x2},V_{y2},V_{z2}],$$

thus $\boldsymbol{s}(p_2)=[x_2, y_2, z_2, Fa_{x2}, Fa_{y2}, Fa_{z2}]$.

In the area of BSP traffic, areas of threats (in the form of spheres) are located, whose parameters will influence the shaping of the trajectory (Figure 3). Each of the hazards is described by a vector of parameters with the form

(25) $$\boldsymbol{thr}_h = [x^h, y^h, z^h, r^h, l^h],$$

where x^h, y^h, z^h are the coordinates of the central point of hazard h, r^h is its radius, and l^h is the level of danger h.

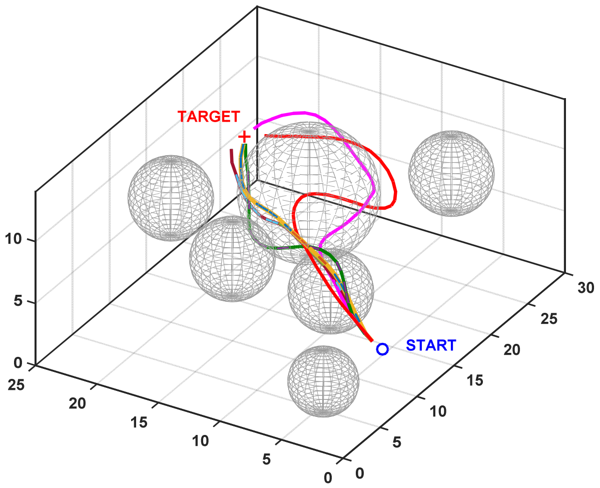

Fig. 3. Threat areas appearing in simulation space

In this work, the problem of route planning has been reduced to determining points in three-dimensional space [12]. The authors do not deal with the dynamics and kinematic relations resulting from the movement of the object; in particular, they do not determine the equations of the object's motion. The algorithm is responsible for constructing the route, which assumes a pattern for the UAV automatic control system.

The limitation of the movement and maneuverability of the simulation object $\boldsymbol{s}(t)$ is accomplished by narrowing the range of changes in the angles φ, ψ, θ to the value

(26) $$\begin{cases} \varphi_{min} < \varphi < \varphi_{max} \\ \psi_{min} < \psi < \psi_{max}, \\ \theta_{min} < \theta < \theta_{max} \end{cases}$$

with $\varphi_{min} = \psi_{min} = \theta_{min} = \text{-}15$ deg, $\varphi_{max} = \psi_{max} = \theta_{max} = 15$ deg.

Next node selection

In the proposed solution, the ACO algorithm was used to determine the sequence of subsequent route points (nodes). However, in order to approximate the problem of route planning to the problem of UAV control, these points are not directly determined. Each ant constructing its solution does not select target coordinates, but selects a set of three angles about which a UAV should be rotated to move in the appropriate direction.

Due to the fact that ants do not choose the coordinates of the space and the angles of rotation of the object, it is not possible to take angle sets as nodes. This is because a sequence of angles in the form of nodes as $\boldsymbol{path}_{1a} = \{\boldsymbol{u}^1, \boldsymbol{u}^2, \boldsymbol{u}^3, \boldsymbol{u}^4, \boldsymbol{u}^5\}$ and $path_{1b} = \{\boldsymbol{u}^5, \boldsymbol{u}^4, \boldsymbol{u}^3, \boldsymbol{u}^2, \boldsymbol{u}^1\}$ will cause the same distribution of pheromones on the nodes in both cases while the actual $path_{1a}$ and $path_{1b}$ trajectories will be different, which means a critical malfunction in the logic of the algorithm's operation. Therefore, the set of N nodes will be the coordinates in space corresponding to the points visited by the ants during the selection of the sets of angles of rotation and movement in accordance with the directions set by them in all iterations. So let

(27) $$\begin{cases} D_1^t = path'^t_1, \\ D_i^t = D_{i\text{-}1}^t \cup (path'^t_i / D_{i\text{-}1}^t), \end{cases}$$

then D^t_a will mean a set of all different nodes visited in iteration t by all ants and

(28) $$\begin{cases} N^1 = D_a^1, \\ N^m = N^{m\text{-}1} \cup (D_a^m / N^{m\text{-}1}), \end{cases}$$

then N^t will mean the set of all nodes in the iteration t, but the traveled path (6) should be modified to the expression

(29) $$path_i^t = \left\{\boldsymbol{\sigma}_i^1, \boldsymbol{\sigma}_i^2, \dots, \boldsymbol{\sigma}_i^{s_{i,t\text{-}1}}, \boldsymbol{\sigma}_i^{s_{i,t}}\right\},$$

(30) $$\boldsymbol{\sigma}_i^p = [\varphi_i^p, \theta_i^p, \psi_i^p], \quad p \in \left(1, 2, \dots, s_{i\text{-}1,t}, s_{i,t}\right),$$

and also

(31) $$path'^t_i = \left\{\boldsymbol{u}_i^1, \boldsymbol{u}_i^2, \dots, \boldsymbol{u}_i^{s_{i,t\text{-}1}}, \boldsymbol{u}_i^{s_{i,t}}\right\},$$

(32) $$\boldsymbol{u}_i^p = [x_i^p, y_i^p, z_i^p], \; p \in \left(1, 2, \dots, s_{i\text{-}1,t}, s_{i,t}\right),$$

During the process of calculating the probability of going to the node $\boldsymbol{u}_j \in N$, the value of the cost of transition J_j is determined by the fitness function with the form

(33) $$F_{cost,j}(\boldsymbol{u}_j, tw, fw) = fw \cdot fuel(\boldsymbol{u}_j) + tw \cdot threat(\boldsymbol{u}_j).$$

The values of tw and fw are parameters regulating the impact of the level of threats and the length of the route on the total cost value, respectively, for the transition to the node u_j, where $tw + fw = 1$, as well as $tw, fw > 0$.

The fuel energy cost function $fuel(\boldsymbol{u}_j)$ is equal to the Euclidean distance from the point in space describing the node j to the point chosen as the target, while the threat function $threat\,(\boldsymbol{u}_j)$ is described by the expression

(34) $$threat(\boldsymbol{u}_j)=\begin{cases}\sum_{q=1}^{h} l_q(r_q\text{-}d_{jq}), & d_{jq}\le r_q,\\ 0, & d_{jq}>r_q,\end{cases}$$

where l_q is the threat level of hazard q and d_{jq} is the distance between the hazard center q and the position in the Cartesian space to node j from the center of q, while h is the number of threat areas. This approach means that the cost of the threat will be the greater, which is the ant closer to the center of the given threat.

At each step, the transition probability will be calculated for points indicated by all combinations of angle sets. In order to reduce the computational complexity of the algorithm, all ranges of angles will be divided with a 7.5 deg resolution, then $\varphi = \psi = \theta = \{-15, -7.5, 0, 7.5, 15\}$deg, creating a set of 125 possible combinations of angles in each step.

For each set $\boldsymbol{\sigma}_{e,f,g}=[\varphi_e, \psi_f, \theta_g]$, where $e \in \varphi$, $f \in \psi$, $g \in \theta$ in step p, locate the node $\boldsymbol{u}_j$ which it indicates according to (14-24) and read the corresponding node value τ_j where

(35) $$\tau_j=\begin{cases}\tau_j, & \boldsymbol{u}_j \in N,\\ 1, & \boldsymbol{u}_j \notin N,\end{cases}$$

then the probability of transition to node $\boldsymbol{u}_j$ should be calculated using the expression

(36) $$p^{i,t}_{e,f,g}(p)=\begin{cases}\dfrac{[\tau_j(t)]^{\alpha}[\eta_j]^{\beta}}{\sum_{k\in K}\{[\tau_k(t)]^{\alpha}[\eta_k]^{\beta}\}}, & j \in K,\\ 0, & j \notin K\end{cases}$$

and select the node and carry out the transition, saving the node to $path'_i$ and $path_i$, and adding the newly visited nodes to the set N. This scheme is repeated until the moment the ant reaches the target.

After completion of the routes by all ants in the given iteration, the pheromone values should be updated on the nodes visited by the ants in accordance with (3-4) and

(37) $$J^t_{c,i}=\sum_{s_{i,t}} J_i^{t,s_{i,t}}.$$

Then the best ant is chosen

(38) $$\boldsymbol{ant}^t_i \xrightarrow{min(J^t_{c,i})} \boldsymbol{ant}_{best}.$$

Simulation results

The simulations were carried out with the following parameters: $\rho = 0.8$, $\alpha = 1$, $\beta = 3$, $Q = 50$, iteration limit $itlim = 180$, starting point $\boldsymbol{s}_p = [7\ \ 1\ \ 5]$, destination point $\boldsymbol{s}_t = [20\ 20\ 10]$ and the initial transition vector

(39) $$V(p_0)=\begin{bmatrix}2 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix},$$

and also, initial orientation

(40) $$\boldsymbol{Fa}(p_0)=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}.$$

The flight direction is determined by the UAV's $0x$ axis, which is initially compatible with the $0x$ axis in the MATLAB simulation environment. The transition vector has a length of 2 units, which is the length of every step in the UAV's movement. The level of all threats has been set to 0.5.The simulations were carried out for three variants of weights:

1. $fw_1 = 0.2$, $tw_1 = 0.8$ (Figures 4, 7),
2. $fw_2 = 0.5$, $tw_2 = 0.5$ (Figures 5, 8),
3. $fw_3 = 0.8$, $tw_3 = 0.2$ (Figures 6, 9),

which allows for the observation of how different weights affect the ACO path building process.

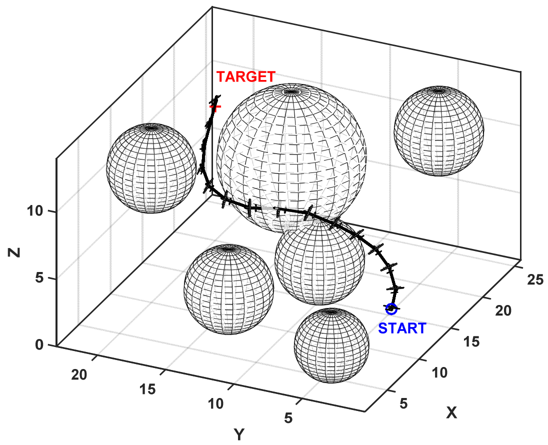

Fig. 4. UAV trajectory for parameters: $fw = 0.2$, $tw = 0.8$

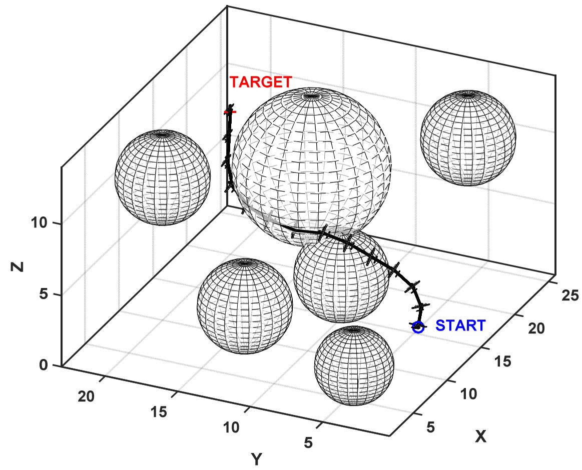

Fig. 5. UAV trajectory for parameters: $fw = 0.5$, $tw = 0.5$

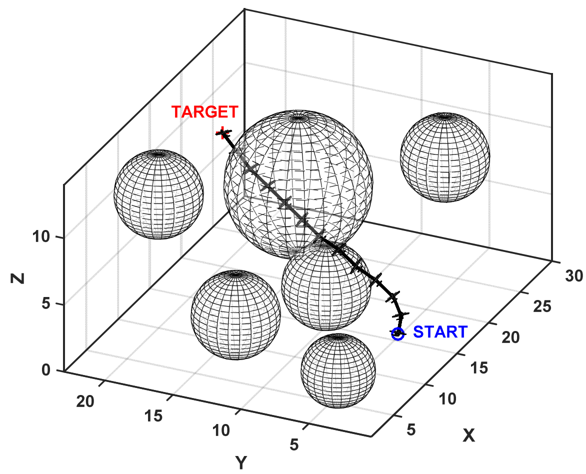

Fig. 6. UAV trajectory for parameters: $fw = 0.8$, $tw = 0.2$

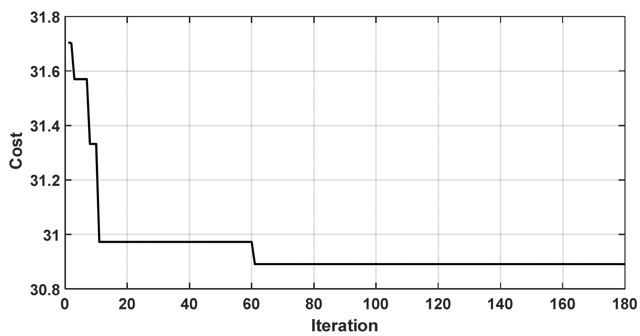

Fig. 7. Path cost optimization plot for parameters: $fw = 0.2$, $tw = 0.8$

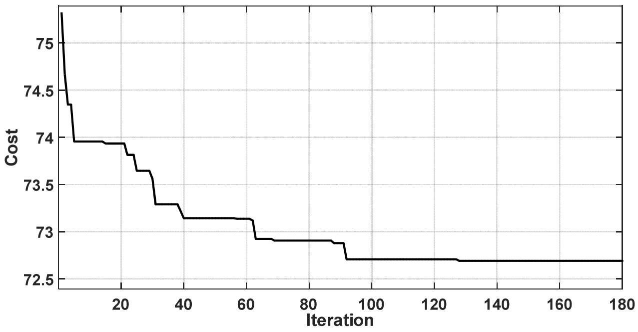

Fig. 8. Path cost optimization plot for parameters: $fw = 0.5$, $tw = 0.5$

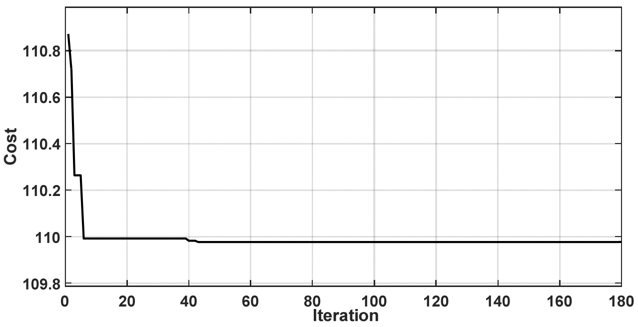

Fig. 9. Path cost optimization plot for parameters: $fw = 0.8$, $tw = 0.2$

Conclusion

In the designed solution, the ACO algorithm operated according to the assumptions. In the initial phase, the program chose the main route variant (it took the main direction of the movement), which was subject to optimization and correction during subsequent iterations.

For the variant $fw_1 = 0.2$, $tw_1 = 0.8$, the algorithm selected a route for the UAV avoiding the threats, only slightly violating the zones. Violation of the zones is acceptable due to setting $tw < 1$. The initial cost of the route without considering the weight parameters was $J_{01} = 55.24$, which with the final result of $J_{best1} = 30.89$, provides a 44.1% improvement.

Variant $fw_2 = 0.5$, $tw_2 = 0.5$, indicates a greater tendency of the UAV to violate dangerous zones when increasing the energy weight, while still keeping the largest possible distance from the hazard center, where according to the relationship (34) the cost of the hazard is the highest. The initial cost at level $J_{02} = 84.27$, with the final cost of $J_{best2} = 72.69$, means an improvement of 13.7%.

For the last variant $fw_3 = 0.8$, $tw_3 = 0.2$ the UAV also behaved as predicted. The trajectory was guided directly by the danger zone, however, a slight detour from the center of the threat area is visible because of the small cost of danger due to the non-zero coefficient tw_3. At the initial cost of $J_{03} = 114.16$, with $J_{best3} = 109.97$, an improvement of 3.7% was achieved.

A low percentage improvement does not necessarily indicate improper operation of the algorithm. In the second and third variant, the decrease in improvement can be explained by the fact that the designated new routes (their resulting shapes) differ from the base route to a lesser extent than in the first attempt. This means that the base route already largely met the weight requirements of the second and third variants and did not require major modifications. The difference between the base shape and the resulting shape of the trajectory is the highest for the first variant, which results in the largest percentage change.

The correct operation of the algorithm allows for further development of the proposed solution. The next step may be the application of known algorithm modifications to improve its performance such as Ant System Rank (ASR) or Max Min Ant System (MMAS) [15].

***Authors**: dr inż. S. Konatowski, e-mail: skonatowski@wat.edu,pl; mgr inż. Piotr Pawłowski, e-mail: piotr_pawlowski@icloud.com, Wojskowa Akademia Techniczna, Instytut Radioelektroniki, ul. Gen. Witolda Urbanowicza 2, 00-908 Warszawa.*

REFERENCES

[1] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B. Vol. 26, No 1, pp. 29-41, 1996.
[2] M. Dorigo, Optimization, Learning and Natural Algorithms. PhD thesis, Politecnico di Milano, Italy, 1992.
[3] D. Merkle, M. Middendorf, H. Schmeck, Ant colony optimization for resource-constrained project scheduling. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 4, 2002.
[4] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, B. Baesens, Classification with Ant Colony Optimization. IEEE Transactions on Evolutionary Computation, Vol. 11, No.5, 2007.
[5] W. Chen, J. Zhang, An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Carious QoS Requirements. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 39, No 1, 2009.
[6] H. B. Tolabi, M. H. Ali, M. Rizwan, Simultaneous Reconfiguration, Optimal Placement of DSTATCOM, and Photovoltaic Array in a Distribution System Based on Fuzzy-ACO Approach. IEEE Transactions on Sustainable Energy, Vol. 6, Issue: 1 , January 2015.
[7] S. Konatowski, P. Pawłowski, Ant Colony Optimization for UAV path planning. 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraina, 2018.
[8] Krzyzanowski, A., Flight mechanics, MUT (2009).
[9] J. S. Dai: Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections. Elsevier, October 2015.
[10] T P. Kaniewski, T. Kraszewski, Drone-based system for localization of people inside buildings. 14th International Conference on Advanced Trends in Radioelecrtronics Telecommunications and Computer Engineering 2018, Lviv-Slavske, Ukraine, pp. 46-51, 2018.
[11] J. Matuszewski, Identification of radar signals using discriminant vectors. XVII International Conference on Microwaves, Radar and Wireless Communications (MIKON), Wroclaw, Poland, MAY 19-21, 2008, Vols 1 and 2, pp. 433-436.
[12] J. Matuszewski, A. Dikta, Emitter location errors in electronic recognition system. Proc. of SPIE Digital Library, XI Conference on Reconnaissance and Electronic Warfare Systems, Vol. 10418, No. 4/2017, pp. C1-C8, 2018.
[13] T. Pietkiewicz, B. Wajszczyk, Fusion of identification information from ELINT-ESM sensors. Proc. SPIE 10715, 2017 Radioelectronic Systems Conference, 107150F.
[14] M. Łabowski, P. Kaniewski, Motion Compensation for Unmanned Aerial Vehicle's Synthetic Aperture Radar. Signal Processing Symposium SPS 2015, Debe, Poland, pp.184-188.
[15] N. Sakthipriya, T. Kalaipriyan, Variants of Ant Colony Optimization – A state of an Art. Indian Journal of Science and Technology, Vol 8(31), November 2015.
[16] R. Róziecki, Bifurkacyjna analiza dynamiki lotu samolotu supermanewrowego z wektorowaniem ciągu. PhD thesis, Politechnika Wrocławska, Wrocław, 2006.