readme

Post-Disaster Operations Improvement: Route Planning and Optimization for Swarm UAVs in Flight-Restricted Areas to Enhance Rescue Efforts

Muhammed Mirac Ozer

Since the obstacles are random, in some cases (for example, when they are close to each other) the solution to the problem is not optimal. In this case, the problem is tried to be solved by applying the Ant Colony Algorithm to the swarm, taking into account the status of all obstacles. The rectangular area containing points X and Y and the obstacles within the area are marked on the map. The line segment was chosen to be the diagonal of the rectangle. Starting from top A, the rectangle is divided into m×n squares and R(m×n) clusters are created as the coordinates of the centers of these squares. It is assumed that each ant can move from one center to another in constant time. In other words, the rectangular area is divided into sections according to the distance the ant can travel in a single time. The starting point (X point) of the ants is shown as and the end point (Y point) is shown as . By noting the rectangles consisting of the cells inside the obstacles, the "tabulist" prohibited data set consisting of the coordinates of the cells and border lines in the obstacles is created. When there are no obstacles, the optimal route will be the line segment connecting the start and end point. With the method in the above study, when there are obstacles, a straight line connecting the points and is specified (along the main line of the rectangles within the obstacles). As a result, a fracture line was formed. If the shortest distance from any () point of the plane to this broken line is marked with , Equation 4 expresses the ant's desire to achieve its goal or the attractiveness of its path.

|  |  |
| --- | --- |
|  | (4) |

At the beginning of the algorithm, the number of ants involved in the L period and the maximum value of the T-iteration are given. The situation of and in the jth iteration of each ith ant is represented by three parameters as shown in Equation 5.

|  |  |
| --- | --- |
|  | (5) |

Here, the first parameter (X) indicates the number of ants in the cell, the second parameter indicates the current amount of pheromone in the cell (σ), and the third parameter indicates whether entry into the cell is free or prohibited (α). It should also be noted that, although the input and output parameters are as shown in Equation 6, is constant when the algorithm works. Initially, all cells except cell are empty as shown in Equation 7.

|  |  |
| --- | --- |
|  | (6) |

|  |  |
| --- | --- |
|  | (7) |

This indicates that all ants are in cell (). The number of ants in cell () in each jth iteration is calculated as shown in Equation 8.

|  |  |
| --- | --- |
|  | (8) |

The number of ants in any cell at the same time cannot exceed L (). Here, the collection is carried out on the ants that choose the cell () in the jth iteration. On the other hand, as shown in Equation 9, the same situation cannot be achieved in two consecutive iterations.

|  |  |
| --- | --- |
|  | (9) |

In other words, if the ith ant leaves the cell in the jth iteration, it cannot enter the cell in the th iteration. Initially, the amount of pheromone in all cells is 0. Thus, Equation 10 occurs in the initial state. In the ith iteration, the probability of moving from cell () to cell () to one of the 8 cells in its neighborhood is calculated as shown in Equation 11.

|  |  |
| --- | --- |
|  | (10) |
|  | (11) |

Here, with the evaporation rate of the pheromone being θ, at the beginning, any of the 8 cells is chosen randomly, with . If the probability is 0 or θ as shown in Equation 12, it means that this cell is not selected.

|  |  |
| --- | --- |
|  | (12) |

It is initially calculated as the inverse value of the length of the path leading to cell (). Equation 9 adjusts the ants' movements according to the obstacles and the broken line . With Equation 11, the ant calculates the probabilities for the cells around each cell, taking into account the situation of Equation 9, and chooses the largest of them. The change of the pheromone occurs when it enters the () cell of each i-th ant, as shown in Equation 13. The evaporation rate of the pheromone is as shown in Equation 14.

|  |  |
| --- | --- |
| = | (13) |
|  | (14) |

The ant colony algorithm, which is tried to be explained with mathematical expressions, is expressed in the pseudo-code shown in Algorithm 2.

**Algorithm 2** Ant Colony Algorithm.

**Begin**

A rectangle is created on the map, with the vertices and .

cluster is created.

sets of obstacles are created.

broken line is created.

The number of ants and the maximum number of iterations, , are given.

All ants are placed in cell .

It is initialized with the formula () and ().

**repeat**

**for** each ant **do**

The ant (), the route is taken in the initial state.

**repeat**

With the formula (), the probabilities are calculated for the cells around each () cell, considering the () condition, and the largest one is selected.

The coordinates of the selected cell are added to the route.

Calculate the length (longitude) of the route .

The pheromone is replaced by the formulas (Eq. 14) and ().

**until** the selected cell is not a border cell.

**end do**

**if** The ant arrived at cell **then**

The ant returns to its initial state and the length of the route is saved.

**end if**

**if** the ant is on the border **then**

The ant returns to its initial state and the length of the route is reset.

**end if**

**end for**

The shortest route from the saved routes is displayed on the screen.

**until** number of iterations 𝑡 ≤ 𝑇 [129].