Periodica Polytechnica Electrical Engineering and Computer Science, 68(3), pp. 309–318, 2024

Path Planning for Data Collection Multiagent System with Priority and Moving Nodes in a Sensing Field with Obstacles

Sára Szénási[1*], István Harmati[1]

[1] Department of Control Engineering and Information Technology, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Magyar tudósok körútja 2, 1117 Budapest, Hungary
[*] Corresponding author, e-mail: olasz-szabo.sara@edu.bme.hu

Received: 16 June 2023, Accepted: 25 January 2024, Published online: 30 April 2024

Abstract
Nowadays there is a more and more common need for one-off data collection in a specified area. For example, in case of searching for the survivors of disasters or wars, or for skiers who get into trouble. The simplest way of collecting such data is using wireless sensor network (WSN). However, data transmission within the network uses great amount of energy that may significantly reduce the lifetime and operation time of the battery-operated sensors. For this reason, the data transfer between the given sensor nodes is carried out by robots. In order to minimize the latency of data collection, it is possible to use cooperating robots. It frequently occurs that one of the nodes requires more urgent visit than the other ones, for example, the serious injuries require urgent medical attendance. The sensor nodes can also be moved to collect more usable data in a larger field. To solve this, path planning for data collection from priority and moving nodes are also presented here. The algorithm developed here to perform the one-off data collection operates so that the agents start from preliminary determined or random points and then visit all the sensor nodes, download the data from these nodes and after this arrive at a designated point where they upload the collected information. The goal here is the minimization of cost that include not only the walk-through time but also the consequences of the late visit of the higher priority nodes. The optimization is solved using the ant colony optimization.

Keywords
path planning, mobile robots, obstacle avoidance, multiagent system, data collection, priority nodes, moving nodes

1 Introduction

Nowadays, there is a more and more common need for data collection in a specified area. The simplest way for such data collection is using wireless sensor networks (WSN) [1, 2]. In most applications, a WSN consists of two parts: one data collection unit (also known as a sink or base station) and a large number of tiny sensor nodes. Typically, both sensor nodes and sink remain static after deployment. Sensor nodes, which are equipped with various sensor units, are capable of sensing the physical world and providing data to the sink through single-hop or multi-hope routing [3]. Sensors are usually powered by batteries, which cannot be replaced in some applications, e.g., battlefield surveillance [4]. Since the data loss rate is increasing with the distance and each data transmission rate is associated with an energy consumption rate, which is modeled as a non-decreasing staircase function of the distance [5], the remote data sending uses much more energy and this in turn deteriorates network lifetime. For these reasons, the data transmission is executed by data collection robots [6, 7]. There have been many applications of this technology in literature in recent years. For example, it is reviewed a range of techniques related to mobile robots in WSNs [8]. It is considered deploying a flying robotic network to monitor mobile targets in an area of interest for a specific time period using WSNs [9]. It is investigated the usage of a mobile sink, which is attached to a bus, to collect data in WSNs with non-uniform node distribution [10]. It is raised and solved a problem of viable path planning for data collecting unicycle robots in a sensing field with obstacles [11]. The robots must visit all sensing nodes and then return to the base station and upload the collected data. However, the robots have limited velocity and this way the data delay is significantly increasing. Since transmitting over a short distance is more reliable than long distance, using robots improves the data collection rate. In addition, in terms of security,

Cite this article as: Szénási, S., Harmati, I. "Path Planning for Data Collection Multiagent System with Priority and Moving Nodes in a Sensing Field with Obstacles", Periodica Polytechnica Electrical Engineering and Computer Science, 68(3), pp. 309–318, 2024. https://doi.org/10.3311/PPee.22788

sending mobile sinks to collect data is more secure than transmitting via multi-hop communication [4]. This may be important in some military applications as well.

Viable path planning for data collection by unicycle robots in a sensing field with obstacles problem raised and solved [11]. The robots must visit all sensing nodes and then return to the base station and upload the collected data. Path planning for the robots is a crucial problem since the created paths directly relate to the delivery delay and energy consumption of the system. In a sensing field, there are obstacles, that the robots must not collide with. For successful path planning, it is necessary to determine the criteria for an adequate path. By definition, a viable path is smooth, collision-free with sensor nodes/base station and obstacles, closed, and provides enough contact time with all the sensor nodes [11]. The data collection is carried out by unicycle Dubins-car [12], which can only move with constant velocity and bounded angular velocity, so it can move only on straight lines and turn with bounded turning radius. Because of the kinematic properties of the robots, the path must be smooth. A safety boundary is determined around obstacles and nodes for the sake of collision-free path. All nodes are bounded with a visiting circle with the minimum turning radius of the unicycle robot. The minimum turning radius is determined according to the speed of the robot and its maximum angular velocity. Moreover, all obstacles convex hull is bounded with a safety margin, since in the case of the shortest path the robot should move on the boundary of the convex hull. The path must be closed because of the periodic data collection. The robot downloads data only when it moves around the visiting circle, so it makes round trips around the node as long as it collects all the data from the sensor node. During path planning, it is assumed that the location of all nodes and obstacles, as well as the shapes of the obstacles are known. Between two objects – safety convex hull of nodes and obstacles – there are always defined four tangents, but any tangents that intersect other obstacle is removed. So, when the robot arrives to a node on a tangent it starts downloading data, and during it makes round trips as long as it collects all the data from the node and then it leaves the node on a tangent. So, a path consists of an adequate configuration of tangents and arcs around objects at the safety distance.

It frequently occurs that one of the nodes requires more urgent visit than the other ones, for example the serious injuries require urgent medical attendance. Path planning with priority nodes is solved in many different ways [13, 14]. These articles are focused to visiting priority nodes first and reducing the planned path length is a secondary objective. One-off data collection is solved for multiagent system with omnidirectional robots [15], where the high priority nodes visiting time and the full data collection time is determined and balanced.

In this paper we raise and solve the problem of path planning between periodically moving and priority nodes to one-off data collection for a multiagent system consisting of Dubins-cars in a sensing field with obstacles. In this paper, the created algorithm balances between path planning and priority nodes. Because of the use of dynamic nodes, the sensor nodes can collect data from a larger area.

This paper, is organized in the following way: Section 2 describes the specified problem and introduces the tools of the solution. In Section 3, the created algorithm for path planning is summarized and the main steps are presented with an example. In Section 4, different measurements to analyze the simulation results are presented. Section 5 describes the simulation results for different sensing fields applying different numbers of agents.

2 Problem statement

During the path planning for data collection Dubins car, the goal is to collect all the data from the sensor nodes. In the sensing field, there are $n' = n + n_{\text{start}} + n_{\text{end}}$ nodes, n sensor nodes, n_{start} start nodes and n_{end} end nodes. The sensor nodes have priority, and these are moving periodically over their data collection circles. In addition to reducing the data collection time, the goal is to visit the high priority nodes as soon as possible. For path planning, we use k agents, which visit all sensor nodes together. Each robot starts on one of the start nodes and ends on one of the end nodes. All sensor nodes store g_i, $i \in [1, n]$ data. In the sensing field, there are also m obstacles, the robots must not collide with them. The data collection path of dynamic nodes and their priority or the location of all obstacles are known. During the data collection all the robots together download the data from all sensor nodes and upload all the collected data to the proper end node. Definition 1 define k-viable sub-paths between priority and moving nodes.

2.1 Definition 1 *k*-viable sub-paths between moving and priority nodes

A path P is k-viable sub-paths between moving and priority nodes if the following 6 conditions are satisfied:

1. smooth,
2. collision-free of obstacles,

3. offers enough contact time to read/write all the data from/to sensor/end nodes;
4. all P_l, $l \in [1, k]$ sub-path starts Start_l, and ends at one of the End nodes;
5. each node is visited by only one viable sub-path;
6. the $cost(P)$ is minimized that consists of the data collection time and the priority cost.

For data collection a multi-agent system with k Dubins cars [12] are used. The Dubins car is a unicycle robot with $X = (x, y, \theta)$ configuration, where $(x, y) \in \mathbb{R}^2$ and θ heading. The robot can move only with constant v velocity and bounded angular velocity

$$\begin{aligned} \dot{x}(t) &= v\cos\theta(t) \\ \dot{y}(t) &= v\sin\theta(t) \\ \dot{\theta} &= u(t) \in [-u_M, u_M], \end{aligned} \tag{1}$$

where u_M denotes the maximum angular velocity. Since the Dubins car can move only with constant velocity and bounded angular velocity, the robot cannot turn with a zero turning radius. The minimal turning radius depends on the robot velocity and the maximal angular velocity

$$R_{\min} = \frac{v}{u_M} \tag{2}$$

Because of the restraints in the possible movement of the Dubins car, it can only move on straight lines and turn with bounded turning radius, also, the planned path must be smooth. To accomplish the smooth path criteria, Definition 2 must be applied. During the data collection, the robot moves around the safety convex hull of obstacles or the data collection circles of nodes and the tangents between them. These arcs and tangents are included to the Tangents Graph as edges, and their tangent points as nodes. Since the Dubins car cannot switch its moving direction immediately, the planned path must satisfy the heading constraint present in Definition 3, as well. The Dubins car's minimal turning radius is $R_{\min} = v/u_M$, so the robot can turn only R radius that satisfied $R \geq R_{\min}$.

2.2 Definition 2 Safety convex hull of obstacles

Any ∂o_j, $j \in [1, m]$ is a smooth curve with the curvature $c(p)$ at any point p satisfying $c(p) \leq \frac{1}{R_{\min}}$ and the minimum distance of the obstacle and the safety convex hull of obstacles is d_{safe}.

2.3 Definition 3 Heading constraint

The heading constraint refers to the fact, that the robot's heading θ at the beginning of an edge in a Tangents Graph should be equal to that at the ending of the last edge.

The second condition of Definition 1 is satisfied when the path does not intersect any safety convex hull of obstacles. In the third condition of Definition 1 offer enough contact time to data transmission. The data collection time depends on the stored data, the data transmission rate and the distance between the node and the robot. The data transfer time can be calculated from the following

$$\delta_i = \frac{g_i}{r(d_i)}, \tag{3}$$

where g_i denotes the data on the i^{th} sensor node, d_i denotes the distance between the robot and the i^{th} sensor node during the data transmission, $r(d)$ denotes the data transmission rate, which is modeled as a non-decreasing staircase function of the distance [5].

$$r(d) = \begin{cases} r_1, 0 < d \leq R_{\min} \\ r_2, R_{\min} < d \leq \frac{3}{2}R_{\min} \\ r_3, \frac{3}{2}R_{\min} < d \leq 2R_{\min} \\ \vdots \\ 0, d > 2R \end{cases}, \tag{4}$$

where

$$r_n = r_1 \frac{R_{\min}}{\frac{n+1}{2}R_{\min}}. \tag{5}$$

The robots download and upload the data only when they move at the data collection circle of the nodes. During the data transmission, the robot and the node move at similar velocity, so the distance between them is constant, and the data transmission rate is also constant (Eq. (4)). Therefore, the data transfer is safe, high quality and energy efficient. In Assumption 1, it is presented the assumption of the moving nodes.

2.4 Assumption 1 Constraints of moving nodes

Each $i \in [1, n']$ node moves periodically on a C_i circle with $R_{\min} \leq R \leq 2R_{\min}$ radius, which is called the data collection circle, with constant v velocity and a pre-defined moving direction.

The problem of determining the clusters and sub-permutations of nodes can be traced back to searching for the shortest k-Hamiltonian path in a graph similarly as a Multiple Travelling Salesman Problem (M-TSP). The k-Hamiltonian path is k tours in the graph, which contains all of the nodes exactly once. To solve this problem Ant Colony Optimization (ACO) is used. Ant Colony Optimization have been inspired by the behavior of real ant colonies, as the ants can find the shortest path between the food source and their anthill [16]. The ACO main idea is the indirect communication between the agents based on pheromone trails. Artificial ants prefer choosing nodes that are close by and connected by edges with a lot of pheromone trails [17]. There are many different methods and ideas to solve TSP with the use of Ant Colony Optimization [18].

To summarize, each $l \in [1, k]$ robot starts at the Start_l node and ends at the End_l points. During the data collection the agents together visit all sensor nodes and load the data from them. During the data transmission, the robots follow the sensor nodes along their data transmission circle and keep a constant distance, while all the data is loaded from them. The agents visit the high priority nodes as soon as possible. The planned data collection path must be collision free. Between two objects, the robots use the common tangent and around the obstacle the robot moves on an arc of obstacle's safety convex hull.

3 Methods

A new method is proposed for planning k-viable sub-paths between priority and moving nodes that fits the conditions of Definition 1. The main steps of the algorithms created are summarized in Algorithm 1.

The first step of the method is determining the clusters and the visiting sequence of nodes that is presented in Section 3.1. The second step of the Algorithm 1 plan path between dynamic sensor nodes in a dynamic sensing field with obstacles is summarized in Section 3.2.

3.1 Create clusters and a visiting sequence of nodes

To create clusters and a visiting sequence of nodes Ant Colony Optimalization is used. In this Section 3.1 the obstacles are ignored during the path planning, only the cluster and sequence creation is achieved. The main steps of path planning between priority sensor nodes are outlined in Algorithm 2. In this method, the centers of C_i, $i \in [1, n']$ data collection circles. We use the path planning algorithm between priority sensor nodes using ACO [15]. The key steps are initialization, path generalization at predefined N_c times, and choosing best tour from the best tours of the cycles.

In each cycle, m ant teams build paths at the same time on the same pheromone trails. Each ant team consists of k ants, and each ant creates one sub-path that starts in one of the start nodes and ends in one of the end nodes. The sub-paths of the same ant team together contain all sensor nodes. The path generalization consists of six steps. The first step is updating the $\boldsymbol{\tau}_{ij}$ pheromone trails between all i and j nodes, where $i, j \in [1, n']$ based on

$$\tau_{ij}\left(t+1\right)=\sum_{h=1}^{m}\tau_{ij}^{h}\left(t+1\right), \tag{6}$$

where $\tau_{ij}^{h}\left(t+1\right)$ is the pheromone update according to the h ant team

$$\tau_{ij}^{h}\left(t+1\right)=\left(1-\rho\right)\tau_{ij}^{h}\left(t\right)+\Delta\tau_{ij}^{h}. \tag{7}$$

The pheromone update consists of two parts, the pheromone trail evaporation and the pheromone emission of the ants based on the previous cycle. The parameter $\rho \in [0, 1]$

Algorithm 1 Planning k-viable sub-paths between priority and moving nodes

1	Run Algorithm 2 to determine the clusters and the visiting sequency of nodes → $\Sigma_1, \Sigma_2, \ldots, \Sigma_k$ sub-permutations of nodes
2	Run Algorithms 3 for all $\Sigma_1, \Sigma_2, \ldots, \Sigma_k$ sub-permutations of nodes → $P_1, P_2, \ldots, P_k$ sub-paths

Algorithm 2 Path planning between priority sensor nodes using ACO

1. Initialize $\boldsymbol{d}_{ij}$, $\boldsymbol{\eta}_{ij}$, $\boldsymbol{\tau}_{ij}$, determine candidate lists and priority filter $\boldsymbol{P}_i^F$ (Eq. (12))
2. Update the pheromone trails according to the Eqs. (6), (7).
3. Place each $l \in [1, k]$ ants at Start_l node.
4. for every $l \in [1, k]$ ant of $h \in [1, m]$ ant team repeat:

   If the tour of l ant is finished, then jump to the next iteration.

   If the tour of l ant contains any sensor nodes then $\boldsymbol{\mathcal{N}}_i^{h(t)} = \boldsymbol{\mathcal{N}}_i^{h(t)} \cup \text{End}$

   If all s, $s \in [1, k] \setminus l$ are finished their tour then $\boldsymbol{\mathcal{N}}_i^{h(t)} = \boldsymbol{\mathcal{N}}_i^{h(t)} \setminus \text{End}$

   4.1 Calculate $\boldsymbol{p}_{ij}^h$ from the actual city i to every city j (Eq. (13))

   4.2 Select next node about q_0 random number or $\boldsymbol{p}_{ij}^h$ probabilities and $\boldsymbol{\mathcal{N}}_i^h$ (Eq. (14)).

   4.3 Update tabu list.
5. Repeat step 4 until all cities are chosen.
6. Determine the cost of the planned paths of ant teams (Eqs. (10), (11)).
7. Save the best – minimal cost – tour of the cycle.
8. Calculate $\Delta\boldsymbol{\tau}_{ij}^h$ from Eq. (8)
9. Repeat step 2–8 N_c times.
10. Choose the minimal cost tour from the best tours of the cycles.

denotes the value of the pheromone trail's evaporation. It can be used to avoid the uncontrolled growth of the pheromone trails left by the ants and it allows the algorithm to forget previous suboptimal routes. The $\Delta\tau_{ij}^{h}$ denotes the pheromone emission of $h \in [1, m]$ ant team in the previous cycle

$$\Delta\tau_{ij}^{h} = \begin{cases} \frac{Q_1}{J^h} + \frac{Q_2}{D_s^h(j)} + \Delta\tau_{ij}^{gb}, & \text{if } h = \text{best} \\ \frac{Q_1}{J^h} + \frac{Q_2}{D_s^h(j)}, & \text{else} \end{cases}, \tag{8}$$

where Q_1 and Q_2 are predefined constants and $\Delta\tau_{ij}^{gb}$ denote the global pheromone update based on the best tour that is the smallest cost tour

$$\Delta\tau_{ij}^{gb} = \frac{Q_{\text{best}}}{J_{\text{best}}}, \tag{9}$$

where Q_{best} is a predefined constant and J_{best} the cost of the best tour of the cycle. The best tour means the minimal cost tour of the cycle. The pheromone update needs to be done for all used edges of each built sub-tour of the cycle. The pheromone emission consists of two parts for each ant team: the J^h cost of the ant team in Eqs. (10), (11) and the $D_s^h(j)$ priority cost to the j^{th} nodes of the s ant's tour in h ant team with Σ_s^h permutation of nodes:

$$\begin{aligned} J_s^h &= L_s^h + \frac{\sum_{i\in\Sigma_s^h} L_s^h(i)\, \boldsymbol{P}^F_{i\in\Sigma_s^h(i)}}{L_s^h} \\ &= L_s^h + \sum_{i\in\Sigma_s^h} D_s^h(i) = L_s^h + D_s^h, \end{aligned} \tag{10}$$

where L_s^h is the time of the data collection of the s^{th} ant of the h^{th} ant team converted to path length. L_s^h consist of the moving time between the nodes and the data collection times according to Eq. (3). $D_s^h(i)$ is the priority cost to i^{th} nodes in Σ_s^h permutation of nodes that is the planned sub-tour of s ant of h ant team. This depends on the required time to visit the i^{th} node and the priority filter $\boldsymbol{P}^F_{i\in\Sigma_s^h(i)}$ (Eq. (8)) of i^{th} node in Σ_s^h.

The cost of the h ant team is the maximum cost of the s ants of h team

$$J^h = \max_s J_s^h. \tag{11}$$

The priority filter is determined by the priority of nodes. The most important nodes have priority 1. The priority filter assigns the largest value to the most important nodes

$$\boldsymbol{P}_i^F = \frac{\max_{j\in n'} \boldsymbol{P}_j + 1 - \boldsymbol{P}_i}{\max_{j\in n'} \boldsymbol{P}_j}. \tag{12}$$

The second step of path creation method is initialization, which consists of emptying the tabu list and placing the k ants at the start nodes. The next step is to build the m paths, consisting of s sub-paths from node to node. The sub-paths are built in parallel to ensure that the more important nodes are visited earlier. At each step, ant s chooses the next place using the probability function depending on the pheromone trails, the distance from the other nodes and the priority filter. The probability that ant s, currently placed at node i, chooses to go to node j at the t^{th} iteration is

$$\boldsymbol{p}_{ij}^{h} = \frac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\boldsymbol{\eta}_{ij}\right]^{\beta}}{\sum_{l\in\mathcal{N}_i^h}\left[\tau_{il}(t)\right]^{\alpha}\left[\boldsymbol{\eta}_{il}\right]^{\beta}} \boldsymbol{P}_j^F \text{ if } j \in \mathcal{N}_i^h, \tag{13}$$

where $\boldsymbol{\eta}_{ij} = 1/\boldsymbol{d}_{ij}$ is an a priori available value that depends on $\boldsymbol{d}_{ij}$ which is the distance between the center of the data collection circle of node i and j. The α and β are two parameters that determine the relative influence of the pheromone trail and the heuristic a priori information. $\mathcal{N}_i^h$ denotes the set that is the feasible neighborhood of ant placed at node i in ant team h, that is, the set of nodes which ant team h has not visited. For better convergence, it can be used as a candidate list that contains the $k_{\text{candidate}}$ closest nodes for each node. The start nodes also have a candidate list, but the candidate list of sensor nodes does not contain the start nodes. The end nodes do not have a candidate list, but the candidate list contains the end nodes. The effect of candidate list takes place inside the set $\mathcal{N}_i^h$. The $\mathcal{N}_i^{h(t)}$ set contains the nodes that are not in a tabu list, and $\mathcal{N}_i^{h(c)}$ set contains the nodes in the candidate list from node i. The set $\mathcal{N}_i^h$ can be determined by the form

$$\mathcal{N}_i^h = \begin{cases} \mathcal{N}_i^{h(t)} \cap \mathcal{N}_i^{h(c)}, & \text{if } \mathcal{N}_i^{h(t)} \cap \mathcal{N}_i^{h(c)} \neq \varnothing \\ \mathcal{N}_i^{h(t)}, & \text{else} \end{cases} \tag{14}$$

thus if there are unselected nodes in the candidate list, the $\mathcal{N}_i^h$ is determined based on the candidate list and the tabu list otherwise, it is solely based on the tabu list.

A random factor is used to find the global optimum [17]. A $0 < q_0 < 1$ constant number is determined and before choosing the next node, a number q is generated randomly. If $q_0 < q$ the next node from set $\mathcal{N}_i^h$ is chosen randomly, or else the next node is determined according to the probability of Eq. (13). After this, choose the next node and add it to the tabu list. In the next, step determine the cost of the ant teams and then choose and save the best tour. The final step of the path creation is to calculate the $\Delta\boldsymbol{\tau}_{ij}^{h}$ based on Eq. (8).

Finally, the best tour is chosen with the smallest cost from the best tours of cycle.

3.2 Path planning between dynamic sensor nodes

The second step of Algorithm 1 is determining a viable path for Dubins car from the created sub-permutations. For this, we create a new method, outlined as Algorithm 3. The Algorithm 3 is running for each sub-permutation independently of each other.

The first step of Algorithm 3 is determining the direction of correct e_{ij} tangents between the data collection circles of all neighborhoods i and j nodes in Σ permutation. The second step is determining all possible Σ_{ij} permutations of blocking obstacles for all e_{ij} direction correct tangents. The third step is creating T_{ij} tree-like graphs between all neighborhood i and j nodes in Σ permutation that satisfy the heading constraint according to Definition 3. The T_{ij} tree-like graphs represent the e_{ij} tangents and their blocking obstacles. The main steps in creating a tree-like graph are summarized in Algorithm 4. The next step in Algorithm 3 is to find all possible $\boldsymbol{P}_{ij}$ paths in the T_{ij} tree-like graphs and calculate their lengths. If the tangents between the data collection circle of nodes i and j do not block obstacles the T_{ij} tree-like graphs contain just the tangent point of the direction correct tangent as nodes and the edge that represents the tangent. The determined path length does not contain the time of data transmissions, only the path length between the data collection circles. The data transmission length can be determined only when the time to arrive to the data collection circle is known, and this way the distance between the robot and the node are known. So, the last step is to find the shortest path in a dynamically changing graph.

Algorithm 3 Path planning between dynamic sensor nodes and obstacles

1. Determine the direction correct tangents between the data collection circles of all neighbourhood i and j nodes in Σ.
2. Determine the blocking obstacles for the tangents
3. Create tree-like graphs which represented to each tangents
4. Find all possible path in the tree-like graphs and calculate the path lengths
5. Find the shortest path

Algorithm 4 Create a tree-like graph between $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ data collection circles

Run for all neighbourhood elements for all Σ_{ij} permutations

1. Add 4–4 arrival and departure tangent points for all obstacle as nodes to T_{ij} the tree-like graph.
2. Add the arc length as edges along the obstacles between the arrival and departure tangent point pairs changed by Definition 3
3. Add the proper tangents as edges between the departure and arrival points.

The final step of Algorithm 3 is to find the shortest path in a dynamic graph based on Ant Colony Optimalization. The main step of this is summarized in Algorithm 5. The first step of Algorithm 5 is determining the $\boldsymbol{d}_{ij}$ distance vector of all possible $\boldsymbol{P}_{ij}$ of all i and j neighborhood nodes of Σ and initializing the $\boldsymbol{\eta}_{ij} = 1/\boldsymbol{d}_{ij}$ reciprocity distance vector and $\boldsymbol{\tau}_{ij}$ pheromone vector variables. Next, begin the path planning iteration N_c times. The first step is generating a q random number, that is greater than the q_0 constant, then the next nodes are generated randomly from the $\boldsymbol{P}_{ij}$ paths. Otherwise, first the $\boldsymbol{\tau}_{ij}$ is normalized for better convergence

$$\tau_{ij}^{(n)}\left(t\right) = \frac{\tau_{ij}\left(t\right)}{\max \tau_{ij}\left(t\right)}. \tag{15}$$

Next the Algorithm 5 determine the $\boldsymbol{p}_{ij}$ probability of each $\boldsymbol{P}_{ij}$ paths

$$\boldsymbol{p}_{ij}^{h} = \frac{\left[\tau_{ij}\left(t\right)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l \in \mathcal{N}_i^h}\left[\tau_{il}\left(t\right)\right]^{\alpha}\left[\eta_{il}\right]^{\beta}} \quad \text{if } j \in \mathcal{N}_i^h, \tag{16}$$

where $\mathcal{N}_i^h$ contain the $\boldsymbol{P}_{ij}$ paths. The cost function of Ant Colony Optimalization is the t_{full} data collection time of each sub-path. The $\Delta\boldsymbol{\tau}_{ij}$ consist of three parts: the first is proportional with the $t_{ij}^{(t)}$ traveling time of the path between the i and j nodes data collection circles. The second and the third parts are proportional with the $t_i^{(v)}$ and $t_j^{(v)}$ visiting time of the i and j data collection circles

$$\Delta\boldsymbol{\tau}_{ij} = \frac{1}{t_{ij}^{(t)}} + \frac{1}{t_i^{(v)}} + \frac{1}{t_j^{(v)}}. \tag{17}$$

Algorithm 5 Shortest path in dynamic graph with ACO

1. Initialize $\boldsymbol{d}_{ij}$, $\boldsymbol{\eta}_{ij}$, $\boldsymbol{\tau}_{ij}$.
2. Run for N_c times
   - 2.1 Do for all i and j neighbourhood element of Σ and all m ants
     - generate q random number
     - if $q > q_0$
       - Choose random path from $\boldsymbol{P}_{ij}$
     - else
       - Normalise pheronome
       - Calculate $\boldsymbol{p}_{ij}$
       - Choose the path from $\boldsymbol{P}_{ij}$ tours according to $\boldsymbol{p}_{ij}$
   - 2.2 Calculate t_{full}
   - 2.3 Calculate $\Delta\boldsymbol{\tau}_{ij}$ and update pheromone
   - 2.4 Save the best tour of the cycle
3. Choose the minimal cost tour from the best tours of the cycles

Then the pheromone is updated

$$\tau_{ij}\left(t+1\right)=\tau_{ij}^{(n)}\left(t\right)+\Delta\tau_{ij}. \tag{18}$$

At each iteration the minimal cost tour is saved, and after the N_c iteration, the best tour with the smallest cost is chosen from the best tours of cycles.

The Example 1 shows an illustrative example of the Algorithm 3.

Example 1: An illustrative example of Algorithm 3: *On Fig. 1 (a) can be seen the example sensing field with* N_{start}, N_1 *and* N_{end} *nodes with positive, negative and positive rotating direction respectively. The permutation of nodes is* $\sum = \{\boldsymbol{C}_{\text{Start}}, \boldsymbol{C}_1, \boldsymbol{C}_{\text{End}}\}$. *On Fig. 1 (b) can be seen the*

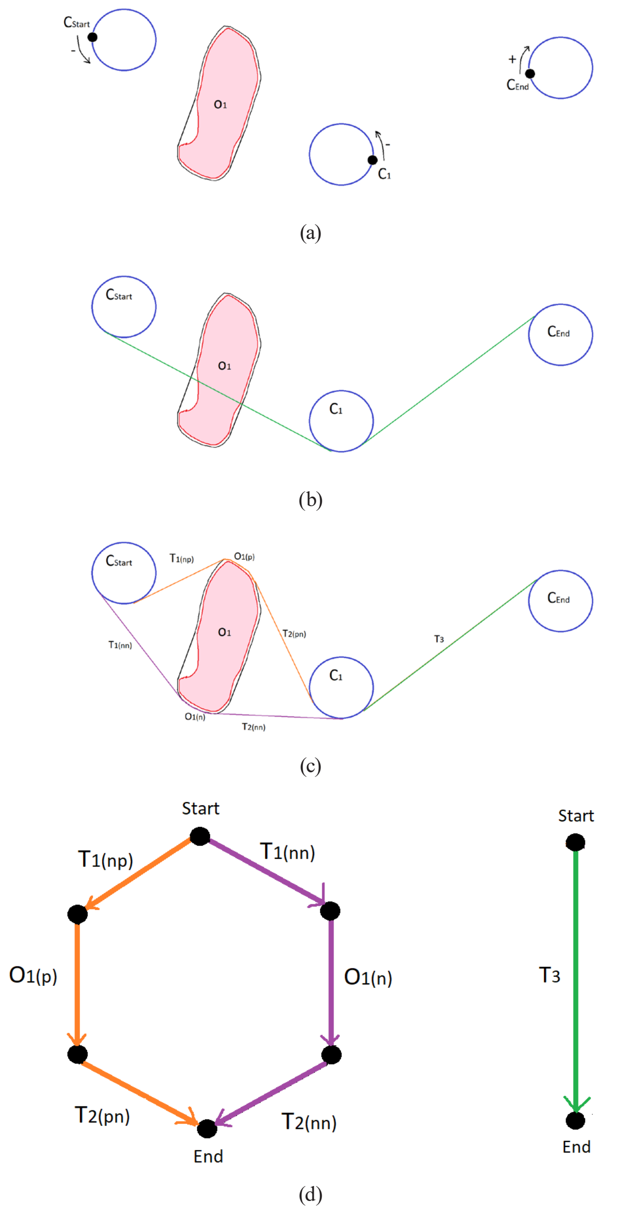

Fig. 1 The illustrative example on Algorithm 3: (a) The sensing field with $\sum = \{\boldsymbol{C}_{\text{Start}}, \boldsymbol{C}_1, \boldsymbol{C}_{\text{End}}\}$; (b) First step of Algorithm 3; (c) Third and forth step of Algorithm 3; (d) The tree like graph between $\boldsymbol{C}_{\text{Start}}$ and $\boldsymbol{C}_1$ or $\boldsymbol{C}_1$ and $\boldsymbol{C}_{\text{End}}$

first step of Algorithm 3, so the direction correct tangents between the data collection circles of all neighborhoods i and $j \in \{\text{Start}, 1, \text{End}\}$ *nodes in* Σ. *The tangent between* $\boldsymbol{C}_{\text{Start}}$ *and* $\boldsymbol{C}_1$ *data collection circle is blocked by the* o_1 *obstacle. The third step of Algorithm 3 is creating a tree-like graph that represents the tangents. On Fig. 1 (c) can be seen the two possible paths between the* $\boldsymbol{C}_{\text{Start}}$ *and* $\boldsymbol{C}_1$ *data collection circle that contain the o1 obstacle. However, there is only one direction proper tangent between* $\boldsymbol{C}_{\text{Start}}$ *and* $\boldsymbol{C}_1$ *data collection circles, that is blocked by the o1 obstacle, with two ways to get around: positive, denoted orange on Fig. 1 (c) and negative, denoted purple on Fig. 1 (c) directions also. On Fig. 1 (d) can be seen the created* $T_{\text{start},1}$ *and* $T_{1,\text{end}}$ *tree-like graphs representing the tangents between* $\boldsymbol{C}_{\text{Start}}$ *and* $\boldsymbol{C}_1$ *data collection circles, denoted orange and purple, or* $\boldsymbol{C}_1$ *and* $\boldsymbol{C}_{\text{End}}$ *data collection circles, denoted by green according to the Fig. 1 (c).*

The last step of Algorithm 3 is finding the shortest path according to $T_{\text{start},1}$ *and* $T_{1,\text{end}}$ *tree-like graphs and the data collection times. There are two different viable paths on Fig. 1 (c) sensing field. The first possible path full data collection time is:* $t_{\text{full}}^{(1)} = t_{\text{Start}}^{(v,1)} + |T_{1(np)}| + |O_{1(p)}| + |T_{2(pn)}| + t_1^{(v,1)} + |T_3| + t_{\text{End}}^{(v,1)}$, *where |edge| denotes the length of the edge and* $t_i^{(v,1)}$ *denotes the visiting time of i node in the first path. The second possible path full data collection time is:* $t_{\text{full}}^{(2)} = t_{\text{Start}}^{(v,2)} + |T_{1(nn)}| + |O_{1(n)}| + |T_{2(nn)}| + t_1^{(v,2)} + |T_3| + t_{\text{End}}^{(v,2)}$. *The Algorithm 3 choose the minimal full data collection time path.*

Remark: *If there are more than one blocking obstacle, the number of possible paths significantly increases. That is the reason why we use the ACO algorithm to find the shortest path.*

4 Measurement

We defined three different measurements to analyze the results. First, we investigated the data collection time of all nodes, this is the maximum of the t_{full}^{l}, $l \in [1, k]$ data collection times of agents. Second, it is investigated the visiting time of all high priority nodes. Third, we analyzed the number of the nodes in each data transmission zone.

5 Simulation results

Simulations are run on a 200 m × 200 m virtual field with 40 sensor nodes and 15 obstacles. The first ten serial numbers used for sensor nodes have high priority nodes, while others have low priority nodes. The start nodes are located in a predefined, randomly chosen place in a sensing field, while the end node is located in the lower left corner. Simulations are run with k agents $k \in [2, 5]$.

The robot velocity is v = 4 m/s and the maximum angular velocity is u_M = 1 rad/s, therefore the minimum turning radius is 4 m.

Each sensor node stores g_i = 0.5 MB data and all collected data is uploaded by the robots to the end node for further analysis. The base data transmission rate is r_1 = 250 kB/s.

The radius of data collection circles are R_{min} = 4 m ≤ R ≤ $2R_{min}$ = 8 m, but R is an integer, so $R \in$ {4 m, 5 m, 6 m, 7 m, 8 m}. There are 8 sensing nodes for all R set elements. The zones of data transmission and the data transmission rates (Eqs. (4), (5)) can be seen on Table 1.

The candidate list is determined by the closest 20% of the number of sensor nodes. The parameters of Ant Colony Optimalization like α, β, ρ, q_0, Q, Q_{best} can be determined in many ways [15, 17–19], the classical parameters are used [17]: $\alpha = 1$, $\beta = 2$, $\rho = 0.1$, $q_0 = 0.9$, $Q = Q_1 = Q_{best} = 1$. $Q_2 = Q_1/5$ is determined by experience method based on the pheromone update balance between the whole and the sub-path length. The count of cycles is 2000 in Algorithm 2 and 50 in Algorithm 5.

It is important to carefully choose the number of applied ants. If the number of ant teams are too low in a cycle, then the path planning will not be deterministic enough based on the actual pheromone values. On the contrary, in the case of too many ant teams, new paths will be overrepresented after the pheromone update compared to the previous cycle's pheromone values. This is caused by the greatly increased computational values. Also, with a large number of ant teams, the probability of uniform path creation is increased, causing some edges to continuously updates during pheromone update and thus distorting convergence. The m count of the applied ant teams is determined based on the number of sensor nodes $m = n'/2$ in Algorithm 2 and $m = n'/4$ in Algorithm 5.

The data collection time of Algorithm 2 and Algorithm 3 can be seen on Fig. 2. On Fig. 2, it can be seen that the data collection time of Algorithm 2 is significantly larger than the data collection time of Algorithm 3, because of

Table 1 The data transmission zones

Zone	d_{min} [m]	d_{max} [m]	$r(d)$ kB/s
Zone 1	0	4	25.000
Zone 2	4	6	16.667
Zone 3	6	8	12.500
Zone 4	8	10	10.000
Zone 5	10	12	8.333
Zone 6	12	14	7.143
Zone 7	14	16	6.250

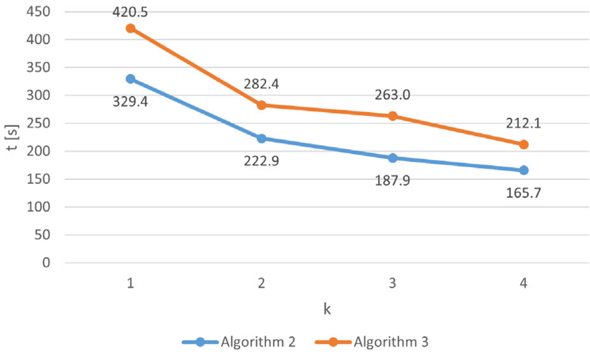

Fig. 2 The data collection time of Algorithm 2 and Algorithm 3

the pre-defined direction around the data collection circle and the blocked obstacles or the decreasing data collection time, because of the Algorithm 2 calculate the data collection time according to the base data transmission rate. The data collection time is decrease of number of agents.

The full and high priority node data collection time of Algorithm 3 can be seen on Fig. 3. It can be seen that the full data collection time and the data collection time of high priority nodes decrease with the number of agents. But the data collection time of high priority nodes depends on the location of the start nodes and the location of the high priority nodes. For example, with the use of k = 4 agents, the data collection time of the high priority nodes significantly decrease compared to the data collection time of the high priority nodes in the case of k = 3 agents, because of the location of $Start_4$ node on Fig. 4 (d).

On Fig. 4, it can be seen that the planned path applies $k \in [2, 5]$ agents. In a sensing field, the N9 sensor nodes is closed by the End node, which causes high data collection times for the high priority nodes. On Fig. 4 (a), the planned path for k = 2 data collection robots. The path of the first robot, denoted by blue, makes a long trip around the o_3 obstacle because of the predefined direction around the data collection

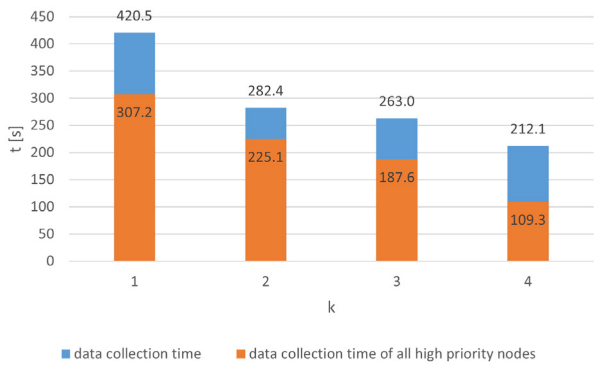

Fig. 3 The full and high priority nodes data collection time of Algorithm 3

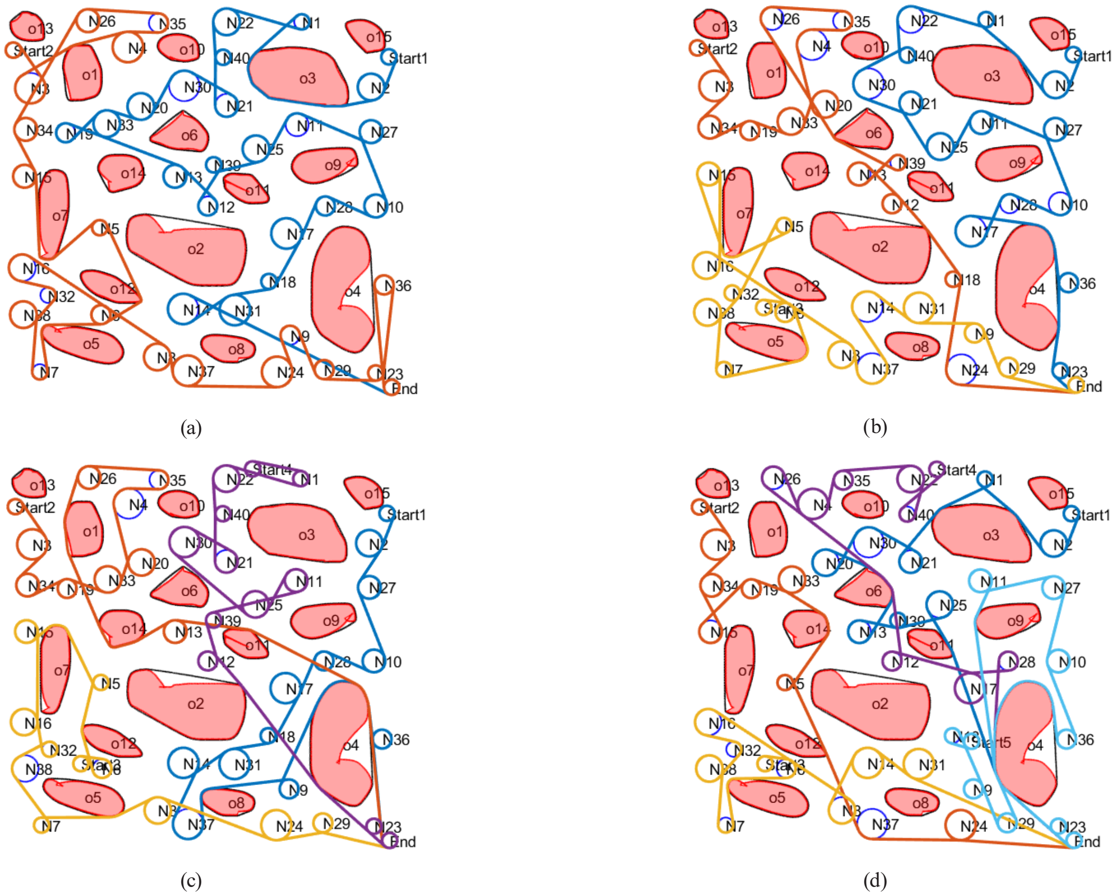

Fig. 4 The planned path with use $k \in [2, 5]$ agents: (a) $k = 2$; (b) $k = 3$; (c) $k = 4$; (d) $k = 4$

circles and to reduce the path length around the next nodes. For example, around the data collection circles of N1, N21, N30, N11 nodes, the robot does not make an extra round trip and the path is not blocked by other obstacles. The second path denoted by red, intersects itself two times to visit the high priority nodes as soon as possible. On Fig. 4 (b), the planned path for $k = 3$ data collection robots. The path of the first robot, denoted by blue, gets around the o_3 obstacle the shorter way. The second sub-path, denoted by red, intersects themselves to visit the high priority N4 node as soon as possible. The third sub-path, denoted by yellow, intersects itself many times and the o_7 and o_5 blocking obstacles make the sub-path length longer. On Fig. 4 (c), the planned path applied $k = 4$ robots. The second sub-path, denoted red, also intersects itself to visit the N4 high priority nodes as soon as possible, and the sub-path contains sensor nodes just the left top corner of the sensing field, then the path blocks about o_{11} and o_4 obstacles. On Fig. 4 (d), the planned path with use of $k = 5$ robots. The Start_5 node is close to the End node, so the N9 sensor nodes will be visited sooner, and this way the data collection time from high priority nodes will be shorter.

In Table 2 shows the number of the nodes in the different data transmission zones. Since most of the data transmissions are done in the first three data transmission zones in all cases, if the data transmission rate is not smaller than the half of the base data transmission rate, the data collection task is properly solved.

Table 2 The number of the nodes of the data transmission zones

Zone/k	2	3	4	5
Zone 1	16	15	15	20
Zone 2	6	6	7	5
Zone 3	7	11	8	8
Zone 4	7	4	6	7
Zone 5	1	6	6	4
Zone 6	7	4	4	6
Zone 7	0	0	2	0

6 Conclusion

In our paper, a path planning algorithm for a multiagent system with Dubins-cars in a sensing field with priority and moving nodes has been proposed. A new algorithm based on Ant Colony Optimalization is developed for handling periodically moving nodes. We run simulations for multiagent system with different numbers of robots. New measurements were defined to analyze the result. The full data collection time and the data collection time of all high priority nodes decrease with the number of the applied robots. But the data collection time of all high priority nodes significantly depends on the location of the start and high priority nodes.

Acknowledgement

Supported by the ÚNKP-22-2-III-BME-233 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

Project no. TKP2021-NVA-02 has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme.

References

[1] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R. "Wireless sensor networks for habitat monitoring", In: The 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, USA, 2002, pp. 88–97. ISBN 978-1-58113-589-3 https://doi.org/10.1145/570738.570751

[2] He, T., Krishnamurthy, S., Stankovic, J. A., Abdelzaher, T., Luo, L., Stoleru, R., Yan, T., Gu, L., Hui, J., Krogh, B. "Energy-efficient surveillance system using wireless sensor networks", In: MobiSys04: The 2nd International Conference on Mobile Systems, Applications, and Services 2004, Boston, MA, USA, 2004, pp. 270–283. ISBN 978-1-58113-793-4 https://doi.org/10.1145/990064.990096

[3] Al-Karaki, J. N., Kamal, A. E. "Routing techniques in wireless sensor networks: a survey", IEEE Wireless Communications, 11(6), pp. 6–28, 2004. https://doi.org/10.1109/MWC.2004.1368893

[4] Gu, Y., Ren, F., Ji, Y., Li, J. "The Evolution of Sink Mobility Management in Wireless Sensor Networks: A Survey", IEEE Communications Surveys and Tutorials, 18(1), pp. 507–524, 2015. https://doi.org/10.1109/COMST.2015.2388779

[5] Ren, X., Liang, W., Xu, W. "Data Collection Maximization in Renewable Sensor Networks via Time-Slot Scheduling", IEEE Transactions on Computers, 64(7), pp. 1870–1883, 2015. https://doi.org/10.1109/TC.2014.2349521

[6] Chatzigiannakis, I., Kinalis, A., Nikoletseas, S. "Sink mobility protocols for data collection in wireless sensor networks", In: The 4th ACM International Workshop on Mobility Management and Wireless Access, Terromolinos, Spain, 2006, pp. 52–59. https://doi.org//doi.org/10.1145/1164783.1164793

[7] Yun, Y., Xia, Y. "Maximizing the Lifetime of Wireless Sensor Networks with Mobile Sink in Delay-Tolerant Applications", IEEE Transactions on Mobile Computing, 9(9), pp. 1308–1318, 2010. https://doi.org/10.1109/TMC.2010.76

[8] Huang, H., Savkin, A. V., Ding, M.,Huang, C. "Mobile robots in wireless sensor networks: A survey on tasks", Computer Networks, 148, pp. 1–19, 2019. https://doi.org/10.1016/j.comnet.2018.10.018

[9] Huang, H., Savkin, A. V. "Reactive 3D deployment of a flying robotic network for surveillance of mobile targets", Computer Networks, 161, pp. 172–182, 2019. https://doi.org/10.1016/j.comnet.2019.06.020

[10] Huang, H., Savkin, A. V. "An energy efficient approach for data collection in wireless sensor networks using public transportation vehicles", AEU-International Journal of Electronics and Communications, 75, pp. 108–118, 2017. https://doi.org/10.1016/j.aeue.2017.03.012

[11] Huang, H., Savkin, A. V. "Viable path planning for data collection robots in a sensing field with obstacles", Computer Communications, 111, pp. 84–96, 2017. https://doi.org/10.1016/j.comcom.2017.07.010

[12] Lester, E. D. "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents", American Journal of Mathematics, 79(3), pp. 497–516, 1957. https://doi.org/10.2307/2372560

[13] Doan, T. T., Bostel, N., Hà, M. H. "The vehicle routing problem with relaxed priority rules", EURO Journal on Transportation and Logistics, 10, 100039, 2021. https://doi.org/10.1016/j.ejtl.2021.100039

[14] Phuong, H. N., Nhat, H. T. N., Hà, M. H., Langevin, A., Trépanier, M. "Solving the clustered traveling salesman problem with d-relaxed priority rule", International Transactions in Operational Reserch, 29(2), pp. 837–853, 2022. https://doi.org/10.1111/itor.12797

[15] Olasz-Szabó, S., Harmati, I. "Path Planning for Data Collection Multiagents System in a Sensing Field with Priority Nodes", In: Kiss, B., Szirmay-Kalos, L. (eds.) Proceedings of the Workshop on the Advances of Information Technology 2023, Budapest, Hungary, 2023, pp. 79–85.

[16] Dorigo, M., Maniezzo, V. Colorni, A. "Positive feedback as a search strategy", Politecnico di Milano, Milan, Italy, Report No. 91–016, 1991.

[17] Dorigo, M., Gambardella, L. M. "Ant colonies for the travelling salesman problem", Biosystems, 43(2), pp. 73–81, 1997. https://doi.org/10.1016/S0303-2647(97)01708-5

[18] Thomas, S., Dorigo, M. "ACO algorithms for the Traveling Salesman Problem", Evolutionary Algorithms in Engineering and Computer Science, 4, pp. 163–183, 1999.

[19] Gaertner, D., Clark, K. "On Optimal Parameters for Ant Colony Optimization Algorithms", In: Proceedings of the 2005 International Conference on Artificial Intelligence, ICAI 2005, Las Vegas, NV, USA, 2005, pp. 83–89.

[20] Olasz-Szabó, S., Harmati, I. "Path Planning for Data Collection Multiagent System in a Sensing Field with Obstacles", In: 25th International Symposium on Measurement and Control in Robotics (ISMRC), Houston, TX, USA, 2022, pp. 1–7. https://doi.org/10.1109/ISMCR56534.2022.9950570