Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 143 (2018) 749–756

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Conference on Advances in Computing and Communication (ICACC-2018)

Intelligent Path Discovery for a Mobile Sink in Wireless Sensor Network

Sanu Thomas[a,]* and Thomaskutty Mathew[b]

[a]*School of Technology and Applied Sciences, Pullarikunnu, Kottayam, Kerala, India*
[b]*School of Technology and Applied Sciences, Pullarikunnu, Kottayam, Kerala, India*

Abstract

An intelligent method to discover the optimal path for a mobile sink that collects data from sensors in a wireless sensor network is presented. The method uses a modified travelling salesman problem that provides efficient data collection mechanism. Here the mobile sink is made to travel along the chords of the circles that represent the communication range of the sensor nodes. The travel path of the mobile sink is composed of series of alternating chords and connecting links in between. During the chord traversal period, the mobile sink is within the communication range of the corresponding sensor and the mobile sink exchanges data with that sensor. This data exchange time period is the designer's choice. The main contribution of this work is to determine the optimal locations of these chords along the tour path of the mobile sink. Here non-linear constrained optimization solver is used to find the optimal path.

Keywords: Intelligent path discovery, mobile sink, fmincon, chord visits.

1. Introduction

When the sensor nodes of a Wireless Sensor Network (WSN) are static and sparsely deployed over a wide geographical area, the network may not be fully connected due to large inter sensor geographical distances. That is,

* Corresponding author. Tel.: +919447212567
E-mail address: thomas.sanu@gmail.com

10.1016/j.procs.2018.10.430

it may not be possible for sensors to send data to the sink by multi-hop communication. In such a situation, a mobile sink is employed to collect data from individual sensors [1-6]. When latency in collecting data is acceptable and the network terrain is suitable for the movement of the sink, a single mobile sink is better than multiple static sinks. A mobile sink also avoids the problem of "sink neighborhood problem" [7] which results in the premature expiry of the network.

The path traversed by the mobile sink is a critical factor in the design of the mobile sink WSN. Several methods are available to find the optimal moving sink path [8]. The mobile sink has to visit every sensor node to collect the data from that sensor. Therefore, the sink travel path can be determined by solving the Travelling Salesman Problem (TSP) where the path is a polygon. Travelling paths can be smoothened using Dubins curves [9]. Here alternate segments are replaced by circular arcs. Quaternions Using B-Splines and Bezier curves are described in [10-11]. But these curves are more complex to generate. Therefore, in this paper, travelling paths are smoothened using cubic spline curves. Here, first the basic TSP path is determined and then smoothing is carried out by cubic spline curves. In the proposed work, a constrained non-linear programming technique is used to determine the optimal path for the mobile sink based on TSP.

In section 2, application of Modified Travelling Salesman Problem (MTSP) to the Travelling Sink (TS) is described. Also different possible types of visits by the TS are given in the subsections of Section 2. In section 3, implementation of MTSP using non-linear integer programming to get different types of paths is explained. Section 4 discusses removal of large turning angles of the TS. Section 5 describes the smoothing of the TS polygon path. Section 6 gives comparison with Dubins method and Section 7 contains the Conclusion.

2. Modified Travelling Salesman Problem (MTSP)

The WSN is represented by a planar graph. The sensor nodes are the vertices of the graph. There are N nodes with known locations. Nodes are identified as 1, 2, …, N. Nodes are assumed to be heterogeneous in their communication range. Therefore, the radii of the circles that represent the communication ranges of different sensor nodes are different.

In TSP [12], the traveler visits every target once with minimum round trip travel distance. In the case of WSN, the TSP is modified where the traveler is the mobile sink or Travelling Sink (TS) and the targets are the sensor nodes. The TS visits each sensor to exchange or collect data. After completing the visits, TS returns to its starting point. The visits by the TS should have sufficient opportunity to exchange data with sensor nodes while the total round trip distance should be minimum. This work proposes three types of visits based on the nearness of the TS to the target node.

2.1. Types of visits

During the tour of the TS through the network, the path taken by TS to reach the sensor can be designed in several ways. Here, three types of visits are considered which are *Very near visit*, *Circumference visit* and the *Chord visit* as depicted in Figs. 1−4.

2.1.1. *Very near visit*

In this case the TS goes very near to the sensor along the red lines as shown in Fig.1. Here the Communication between the sensor and the TS is highly reliable. The power required by the sensor to transmit data to the TS is the lowest.

2.1.2. Circumference visit

In this case, the traversing path of the TS is designed to contact the circumference of the communication circle of each sensor once as shown in Fig. 2. While departing from a specific contact point or while arriving to a specific contact point, the path may cross and intersect a circle one or more times which is incidental. (See circle 2 in Fig. 2.)

2.2. Chord visit

In this case, the TS path intersects every communication circle and forms a chord at each of these circles as shown in Fig. 3. When the TS is passing through the chord of the communication circle of a specific node, the distance between the TS and that node is less than the communication radius of that circle. Then data exchange can take place between the TS and the node without any halt by the TS. In chord visits we can have either *Constant subtended angle chords* or *Fixed length chords*. It is the designers choice to go for either fixed length or fixed subtended angle chords.

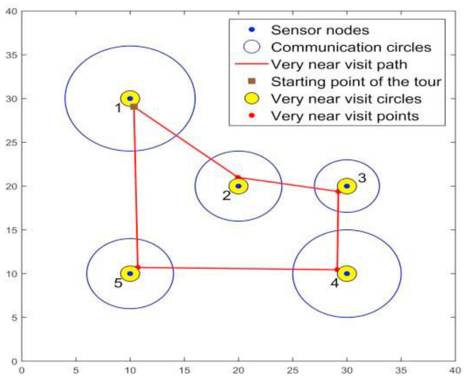

Fig. 1. Very near visit

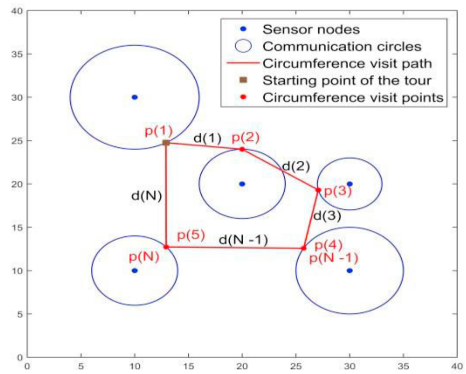

Fig. 2. Circumference visit

2.2.1. Constant subtended angle chords

In this case, the angle θ subtended by the chord at the center of the circle is constant for all the circles as shown in Fig. 3. Here the chord length $= 2*r*\sin(\theta/2)$ where r is the radius of the communication circle. Thus the chord length is directly proportional to the radius of the circle. Therefore, for larger r, the passage period of TS within the circle is longer and hence the TS and that sensor can exchange more data. This design is very much useful for signals having high data rate and low duty cycle like audio/video sensors.

2.2.2. Fixed length chords

In this case, the chord length is fixed irrespective of the radius of the circles as shown in Fig. 4. Here, the total path length of the tour can be reduced by choosing a smaller fixed length for the chords.

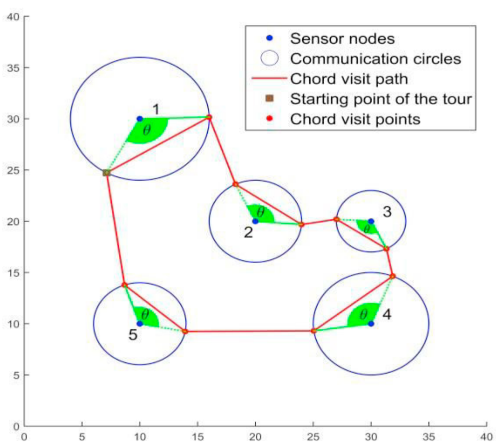

Fig. 3. Constant subtended angle chord visit

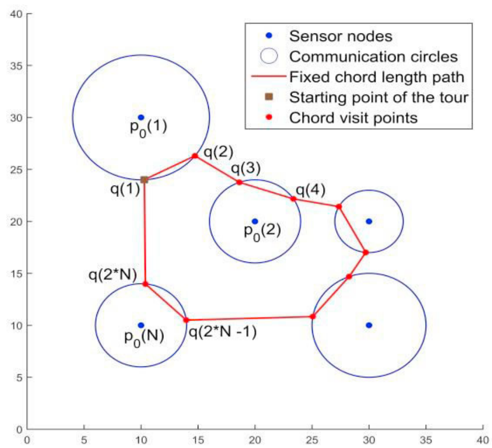

Fig. 4. Fixed length chord visit

3. Implementation of MTSP

Initially, the basic TSP tour for the given set of N nodes is implemented using the standard binary integer programming based algorithm [12]. This gives the optimal permutation of nodes to be traversed. The nodes are reordered and renumbered according to this permutation and the starting (and ending) node is taken as 1. Let the x-y coordinates of these nodes (which are known) be represented by a vector of points as,

$$\boldsymbol{P_0} = [p_0(1), \quad p_0(2), \quad \dots \quad p_0(N)] = \begin{bmatrix} x_0(1) & x_0(2) & \cdots & x_0(N) \\ y_0(1) & y_0(2) & \cdots & y_0(N) \end{bmatrix} \tag{1}$$

Here, $x_0(i), y_0(i)$ give the coordinates of node i among the reordered and renumbered nodes. Actually, these are the coordinates of the center of the communication circle of node i. The point corresponding to node i, for $i = 1$ to N, is represented by

$$p_0(i) = \begin{bmatrix} x_0(i) \\ y_0(i) \end{bmatrix} \tag{2}$$

3.1. Circumference visits

During the round trip tour by TS, the target points are the points on the communication circles (here after referred as 'circles') of the respective nodes. Let $x(i)$, $y(i)$ be the coordinates of the i^{th} target point which lies on the circumference of the i^{th} circle. Then the point vector $\boldsymbol{P}$ of these target points is represented as,

$$\boldsymbol{P} = [p(1), \quad p(2), \quad \dots \quad p(N)] = [\{x(1), y(1)\}, \{x(2), y(2)\}, \dots, \{x(N), y(N)\}] \tag{3}$$

Since the point $x(i)$, $y(i)$ is on the circumference of the i^{th} circle whose center is at $x_0(i)$, $y_0(i)$ with radius $r(i)$, we have,

$$\big(x(i) - x_0(i)\big)^2 + \big(y(i) - y_0(i)\big)^2 = r(i)^2 \tag{4}$$

for $i = 1$ to N. Since points $p(i)$'s constitute the travel path of the TS, the total distance contributed by p(i)'s should be minimum. Therefore, the path is made up of straight line segments. Hence the sum D of the line segments is given by,

$$D = d(1) + d(2) + \dots + d(N-1) + d(N) \tag{5}$$

Here, for i =1 to (N−1),

$$\left.\begin{array}{ll} \text{length of the } i^{th} \text{ segment is,} & d(i) = \sqrt{\big(x(i) - x(i+1)\big)^2 + \big(y(i) - y(i+1)\big)^2} = \|p(i) - p(i+1)\| \\ \text{length of the last, path closing segment} & d(N) = \sqrt{\big(x(N) - x(1)\big)^2 + \big(y(N) - y(1)\big)^2} = \|p(N) - p(1)\| \end{array}\right\} \tag{6}$$

In (6), $\|\dots\|$ symbol is used to represent the Euclidian norm. For example, $\|p(i) - p(i+1)\|$ gives the Euclidian distance between points $p(i)$ and $p(i+1)$. Visit (way) points $p(1)$, $p(2)$,…, $p(N)$ and distances $d(1)$, $d(2)$,…,$d(N-1)$, $d(N)$ form the travel polygon as shown in Fig. 2. The objective is to minimize D given by (5), subjected to the constraint given by (4) which restricts the visit points to lie on the circumferences of the circles.

3.2. Determination of the optimal circumference visit path

Here, the constraint given by (4) and D, given by (5) supported by (6), both are non-linear. Therefore the *optimal circumference visit path* problem is solved using the nonlinear constrained minimizer solver '**fmincon(…)**' [13]. Since constraint (4) and D given by (5) are convex functions, the **fmincon(…)** converges and gives the optimum result.

3.2.1. Example 1

In this simple example, number of nodes, $N = 5$ as shown in Fig. 2. The nodes are already reordered and renumbered using a standard TSP algorithm. The coordinates of the nodes, that is, the centers of the circles and their communication ranges (radii of the communication circles) in meters are shown in Table 1. TS visit points are determined using **fmincon(…)** and the path is shown in red in Fig. 2. In Table 1, the length d(i)'s are in meters.

Table 1. Distance values in meters for Example 1

Node ids	Node Locations (x_0, y_0)	Radii	TS visit points (x, y)	Segment length d(i)'s
1	(10, 30)	6	(12.9127, 24.7544)	7.1226
2	(20, 20)	4	(19.9952, 24.0000)	8.5034
3	(30, 20)	3	(27.0825, 19.3012)	6.8612
4	(30, 10)	5	(25.7156, 12.5776)	12.7880
5	(10, 10)	4	(12.9285, 12.7247)	12.0297
			Optimal Total length of the tour =	47.3049

3.3. Realization of very near visit

In, Equation (4), $r(i)$'s give the distance between the visit points and the corresponding nodes (center of the circles). Therefore by choosing $r(i) = e =$ a very small distance for all $r(i)$'s, the visit points are made very close to the nodes. The value of e is the designer's choice. When $r(i)$'s are set to e, Equation (4) is modified as,

$$\left(x(i) - x_0(i)\right)^2 + \left(y(i) - y_0(i)\right)^2 = e^2 \qquad (7)$$

Now D as given by (5), is minimized with constraint (7) to get the *very near visit* points. Thus the *very near visit* problem is a special case of the *circumference visit* problem.

3.4. Chord visits

In chord visits, the TS path intersects each circle at two points to form the chord at that circle as shown in Fig. 4. There are in total $2*N$ chord points. The chord points corresponding to the successive circles are denoted as,

$$Q = [\ \{q(1), q(2)\}, \{q(3), q(4)\},\ldots, \{q(2*i-1), q(2*i)\},\ldots,\{q(2*N-1), q(2*N)\}\ ] \qquad (8)$$

Here, point $q(j)$ is a vector as $q(j) = \{x(j), y(j)\}$ which are the x-y coordinates of point $q(j)$. In (8), the two chord points on the i^{th} circle are $\{ q(2*i-1), q(2*i)\}$. Since, points $q(2*i-1)$ and $q(2*i)$ are lying on the i^{th} circle whose center is at $p_0(i)$ and whose radius is $r(i)$, we have the circle constraint as,

$$\|p_0(i) - q(2*i-1)\| = r(i) \qquad (9)$$

$$\|p_0(i) - q(2*i)\| = r(i) \qquad (10)$$

for $i = 1$ to N.

3.4.1. Chord length constraint

In the case of constant subtended angle θ, the chord length constraint for the i^{th} circle is given by the well-known trigonometric formula,

$$\|q(2*i-1) - q(2*i)\| = 2*r(i)*sin\left(\frac{\theta}{2}\right) \qquad (11)$$

In the case of fixed length chords of length say L, the constraint for all chords is,

$$\|q(2*i-1) - q(2*i)\| = L \qquad (12)$$

3.4.2. Total length of the TS path

The perimeter of the polygon formed by the 2*N chord points (which is also the length of the tour) is given by,

$$D = \sum_{j=1}^{j=2*N-1} (\|q(j) - q(j+1)\|) + \|q(2*N) - q(1)\| \tag{13}$$

The last term on the RHS of (13) gives the polygon closing length from $q(2*N)$ to $q(1)$.

3.4.3. Realization of Chord visit path

The objective is to minimize D given by (13), subjected to the constraints (9), (10) and (11) for constant subtended angle chords or constraints (9), (10) and (12) for fixed length chords. Here also, we use **fmincon(…)** to solve for the points $q(1)$, $q(2)$,…,$q(2*N)$. An example for the constant subtended angel chords visit is shown in Fig. 3, where Θ is set to 120^0. Example for fixed length chord visit is shown in Fig. 4, where the fixed length of the chords is set to 5 meters.

4. Removal of large turning angles

When the TS path polygon has large turning angles, a robot or says a drone or any other vehicle which carries the TS experiences sharp velocity changes. (Large turning angle means sharp curvature). This prevents the smooth travel of the TS carrying vehicle. A sub-path having a large turning angle $\beta(j)$ and a small turning angle $\alpha(j+2)$ is shown in Fig. 5.

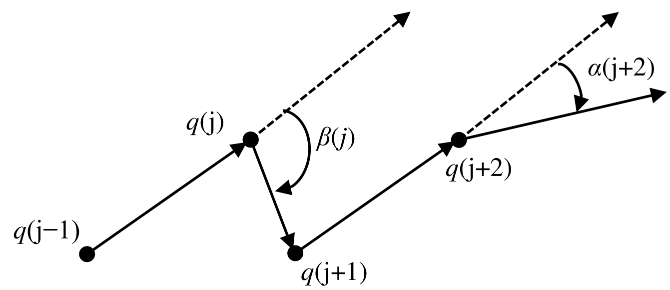

Fig. 5. Turning angles or angles of deviation along the TS path

4.1. Calculation of turning angles

Turning angle (or angle of deviation) is determined using the well-known cosine angle formula using the dot product of the two corresponding vectors as,

$$\beta(j) = cos^{-1}\left\{\frac{\big(q(j) - q(j-1)\big) \bullet \big(q(j+1) - q(j)\big)}{\|q(j) - q(j-1)\| * \|q(j+1) - q(j)\|}\right\} \tag{14}$$

for j = 1 to N. In (14), $q(1)$ is the starting point, $q(N)$ is the ending point and $q(0) = q(N)$ because of the TS path forms closed loop. Let the maximum permissible turning angle be βmax. Then the Angle Constraint (AC) can be expressed as $\beta(j) \le \beta max$. The value of βmax is the designer's choice. This can be expressed in the light of (14) as,

$$cos^{-1}\left\{\frac{\big(q(j) - q(j-1)\big) \bullet \big(q(j+1) - q(j)\big)}{\|q(j) - q(j-1)\| * \|q(j+1) - q(j)\|}\right\} \le \beta max \tag{15}$$

In (14) and (15), point $q(j)$ is a vector as $q(j) = \{x(j), y(j)\}$ which are to be determined to find the optimal path.

4.2. Chord visit path with Angle Constraint(AC)

Here, our objective is to minimize D given by (13), subjected to the constraints (9), (10), (15) and (11) for constant subtended angle chords or constraints (9), (10), (15) and (12), for fixed length chords. Additional constraint (15) provides the AC. Again we use **fmincon(…)** to minimize D with the above mentioned constrains.

4.2.1. Example 2

Here $N = 9$ and we have 18 chord points. The optimal chord visit path without AC is shown in Fig.6. The turning angles at chord points without AC are shown in the first data row of Table 2. After introducing the additional constraint (15) with $\beta max = 80$, the resulting path is shown in Fig. 7. The turning angles with AC are shown in the second data row of Table 2. Angles are shown in degrees nearest to the integer. Turning angles which were greater than 80 are shown in red. With AC, the turning angles are kept less than or equal to βmax (80) as seen in data row 2 of Table 2. With AC the total length of the path increases compared to that without AC.

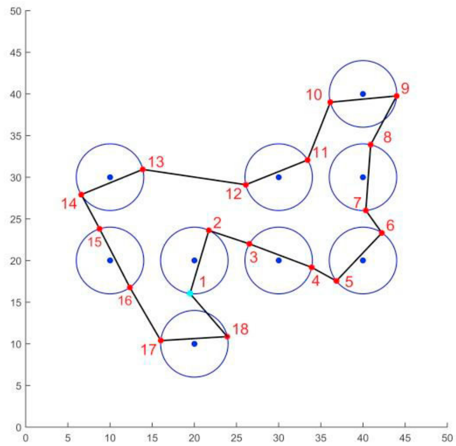

Fig. 6. Optimal path without angular constraint

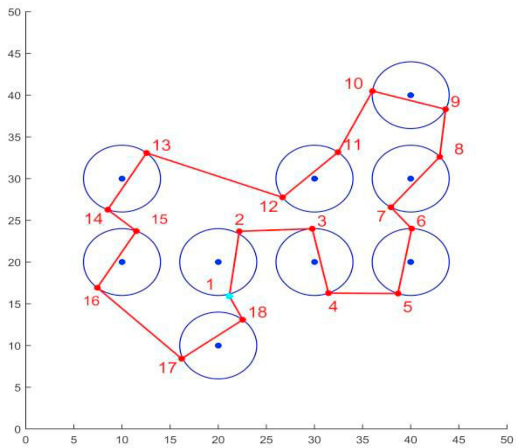

Fig. 7. Optimal path with angular constraint

Table 2. Turning angles (in degrees) without and with angle constraints

AC	β1	β2	β3	β4	β5	β6	β7	β8	β9	β10	β11	β12	β13	β14	β15	β16	β17	β18
No	57	92	2	8	76	78	39	24	123	63	47	31	31	95	1	3	64	127
Yes	34	80	80	77	80	50	80	34	80	80	21	64	80	80	80	77	80	80

5. Smoothing of TS polygon path

If the TS path is gently curved at corners of the polygon, the TS vehicle can smoothly negotiate the corners. In this work, we adopt **Cubic Spline** [14] curve fitting method for smoothing the polygon path of the TS which is shown in

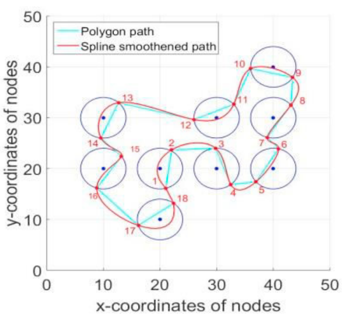

Fig. 8. Spline curve smoothened path

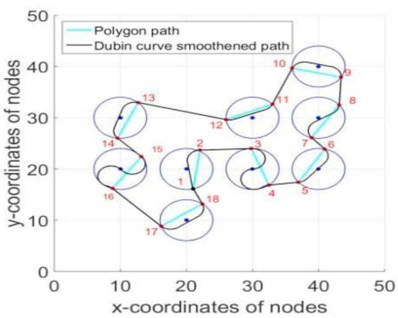

Fig. 9. Dubins curve smoothened path

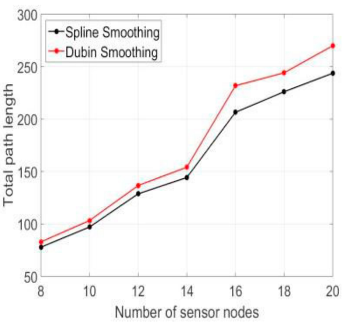

Fig. 10. Spline & Dubins smoothing

In Fig. 8. The spline smoothed curves are derived from the TS path polygon using the standard spline fitting algorithm.

6. Comparison with Dubins method

The TS smooth path can be obtained by Dubins curve smoothing as described in [5-6]. In Dubins method, Circular arcs are used inside the communication circles while straight line segments are used outside, as shown in Fig. 9. In cubic spline smoothing method, the whole path is smoothly curved throughout. The optimal distances of the paths calculated by Dubins smoothing [9] and by cubic spline smoothing, for different number of nodes are shown in Fig. 10. For small number of nodes, there is not much difference between the two methods. In a sparsely populated WSN with large number of nodes, the proposed method is better compared to the Dubins method.

7. Conclusion

The work presents a new method of finding the best path for mobile sink when the sensors are heterogeneous with variable communication ranges. The Chord visit method described in this paper is a novel technique for better communication between the travelling sink and the sensors. This method can collect higher volume of data compared to other methods. The designer can use constant subtended angle chord lengths, fixed chord lengths and variable chord lengths. Variable chord lengths can be selected based on the heterogeneity of sensor nodes. An upper bound can be set for the turning angles of the vehicles carrying the Travelling Sink. The maximum turning angle depends on the type and velocity of the vehicle. The Minimum curvature requirement is satisfied using Cubic Spline Interpolations. The chord visit method is a centralized one and this can be extended to three dimensional WSNs which are deployed to monitor underwater and terrestrial regions.

References

[1] Tunca, C., Isik, S., Donmez, M. Y., and Ersoy, C. (2014) "Distributed Mobile Sink Routing for Wireless Sensor Networks: A Survey." *IEEE Communications Surveys & Tutorials* **16** (**2**): 877−897.

[2] Mario Di Francesco, Das, S. K., and Anastasi, G. (2011) "Data collection in wireless sensor networks with mobile elements: A survey." *ACM Trans Sensor Networks* **8** (**1**): 1–31.

[3] Liang, W., Luo, J., and Xu, X. (2010) "Prolonging network lifetime via a controlled mobile sink in wireless sensor networks." *Global Telecommunications Conf. (GLOBECOM 2010), IEEE*: 1−6.

[4] Wang, Z., Basagni, S., Melachrinoudis, E., and Petrioli, C. (2005) "Exploiting sink mobility for maximizing sensor networks lifetime." *Proc. of the 38th Annu. Hawaii Int. Conf. on System Sciences (HICSS '05)*: 1−9.

[5] Hailong Huang. (2017). "Performance Improvement by Introducing Mobility in Wireless Communication Networks." *arXiv*: 1712.02436.

[6] Le Ny, J., Frazzoli, E., and Feron, E. (2007) "The curvature-constrained traveling salesman problem for high point densities." *46th IEEE Conference on Decision and Control*, New Orleans, LA: 5985−5990.

[7] Xu, X. and Liang, W. (2011) "Placing the minimum number of sinks in sensor networks for network lifetime maximization." *Proc of ICC'11, IEEE*: 1−6.

[8] Rao, J., and Biswas, S. (2008) "Data harvesting in sensor networks using mobile sinks," *IEEE Wireless Commun*, **15** (**6**): 63–70.

[9] Savla, K., Frazzoli, E., and Bullo, F. (2005) "On the point-to-point and traveling salesperson problems for Dubins' vehicle." *Proc. Amer. Control conf*: 786−791.

[10] Neubauerand, M., and Mller, A. (2015) "Smooth orientation path planning with quaternions using B-splines." *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*: 2087–2092.

[11] Faigl, J., and Vana, P. (2018) "Surveillance Planning With Bézier Curves." *IEEE Robotics and Automation Letters* **3** (**2**): 750-757.

[12] Diaby, M. (2007) "The Travelling Salesman Problem: A Linear Programming Formulation." *WSEAS Transactions on Mathematics* **6** (**6**): 745-754.

[13] Rajesh Kumar Arora. (2015). "*Optimization : Algorithms and Applications."* Hoboken, New Jersy : CRC Press.

[14] de Boor, Carl, A. (1978). "*Practical Guide to Splines*. New York: Springer"-Verlag.