Complexity of Energy Efficient Localization with the Aid of a Mobile Beacon

Hüseyin Akcan, Cem Evrendilek

***Abstract*—Localization is an essential service in wireless sensor networks. Trilateration is a commonly used solution to range based localization for providing such services. It might be, however, impossible to localize the entire network at once using trilateration due to low connectivity on sparse deployments. In such scenarios, a mobile beacon with a known position is used to move among and locate the nodes with low connectivity to aid trilateration. Given a network graph, finding a minimum energy route traveled by the mobile beacon is a key problem in many real world applications. We prove in this paper that this problem called *Mobile Assisted Trilateration Based Energy Optimum Localization* is NP-hard. To the best of our knowledge, this is the first such result in an attempt to computationally classify this important problem. We also provide a compact integer linear programming formulation for the problem.**

***Index Terms*—Localization, trilateration, NP-hardness, integer linear programming.**

I. Introduction

Given a network graph $G(V, E)$, range based network localization problem in $2D$ is to determine the locations of all the nodes $v \in V$ in $\mathbb{R}^2$ using the distances $\delta_{i,j}$ available between the nodes $\{i, j\} \in E$ [2]. The problem is shown to be strongly NP-hard in k-space for all $k > 0$ in [13]. Aspnes *et al.* [3] proved that localization in Unit Disk Graphs (UDG) is also NP-hard. As Wireless Sensor Networks (WSN) are typically modeled using UDGs, the complexity result has a direct implication for such networks.

A well known polynomial time solution to the range based network localization problem is the trilateration algorithm, assuming that the network graph has a trilateration ordering [1], [6]. A trilateration ordering is defined as a sequence starting with a seed of three nodes with known positions and an ordering on the rest of the nodes such that each node has three edges to the nodes earlier in the sequence. Given the result that 6-connectivity is sufficient for global rigidity, if the wireless range of each node is large enough as suggested in [6] then, with high probability, the network graph is localizable in linear time. However, on sparse graphs, trilateration can be carried out only partially due to the insufficient connectivity among neighbors. One possible solution for sparse graphs is then to use a mobile beacon as an external agent to locate the individual nodes with low connectivity, and help resume trilateration for the rest of the graph [9], [10], [11]. A recent survey paper [8] reports in detail a considerable body of research studying the class of problems in which one or more mobile beacons are used to assist localization.

We state first the assumptions made before a formal definition of the *Mobile Assisted Trilateration Based Energy Optimum Localization* (*MATBOL*) problem can be given. It is assumed that the wireless nodes to be localized form a connected undirected graph. Each edge in such a graph simply indicates that the corresponding pair of nodes is within range of one another, and the weight associated with the edge is a measure of the distance. A special node, called the mobile beacon, has the ability to move. It can travel along the edges from one node to another. Both the static nodes and the mobile beacon are equipped with wireless radios to measure pairwise distances within wireless range, called ranging. Any extra equipment on the mobile might have an effect on the total distance and the pattern traveled in order to localize the entire network. In the current formulation of the problem in this paper, the mobile beacon is assumed to have no extra hardware to support any additional functionality other than the ones required to do ranging and to move (in any direction relative to the local coordinate system of the mobile). It should also be noted that the coordinates of the nodes localized are all relative to the locations of the first three nodes visited by the mobile as dictated by the trilateration ordering. Following this model, once the mobile is deployed right next to some node designated as the base node, it is possible to find the relative locations of all its neighbors with some constant distance traveled. As such, the mobile can keep visiting the nodes in the network by traveling along the direction of the edges as made possible by ranging. The gradient formed along the edges assures that the mobile is not lost throughout its mission.

We introduce $MATBOL$ as a localization problem for WSNs. The input for a given instance of $MATBOL$ is a graph with the nodes in general position, as well as a designated base node. The objective is to find the shortest path traveled by the mobile beacon, sitting initially at the base, such that all the nodes are localized either by the mobile beacon itself, or using trilateration. As the movement by the mobile beacon is the primary source of the energy dissipation, traveling the shortest distance is essential in minimizing the energy utilized. It should be noted, however, that setting the objective to minimize the total distance traveled by the mobile is not the only viable option. An equally important choice for the objective could be to maximize the accuracy of the localization achieved [4], [12]. In many real world applications, where a mobile is employed to assist the localization, the objective could be either the shortest distance traveled or the most accurate localization achieved, or even a combination of these two. In this paper, we are mainly concerned with the minimization of the energy consumed by the mobile beacon while it is moving. It is worth noting at this point that the problem of energy efficient localization with the aid of one or more mobile beacons is at the top of the list of open research problems in [8]. Therefore, proving $MATBOL$ NP-hard is an essential first step taken to demonstrate the computational difficulty of this

H. Akcan is with the Department of Software Engineering, and C. Evrendilek is with the Department of Computer Engineering, Izmir University of Economics, Balçova, Izmir, Turkey, 35330.
E-mail: huseyin.akcan@ieu.edu.tr, cem.evrendilek@ieu.edu.tr

class of problems.

It is already known that noisy measurements, inevitable in many real world applications, has a huge impact on localization. Such distance measurement errors turn, for example, the complexity of trilateration from polynomial to NP-hard [7]. It should be noted, however, that we assume error-free distance measurements in $MATBOL$. This is a decision made deliberately as the NP-hardness of $MATBOL$ directly implies NP-hardness when the distance measurements are noisy. Moreover, such a result would readily imply the NP-hardness also when the goal is to travel the shortest distance with an additional constraint used to ensure a specified localization accuracy.

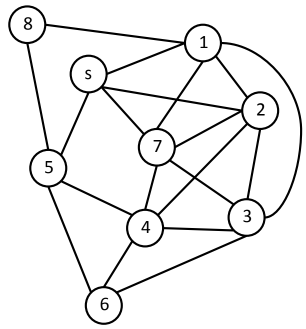

Fig. 1. An example WSN graph. The objective of the mobile at s is to localize the entire graph by travelling the minimum distance possible.

Figure 1 shows an example graph where the mobile beacon rests on the base denoted by s. For the sake of brevity, an unweighted version of the graph is presented in the figure. If the mobile beacon visits the nodes labeled 2, 1, and 8 in this order, then the rest of the nodes could be localized using trilateration only, without the need for the mobile to travel any further. Node 7 is localized using the nodes in $\{s, 1, 2\}$, node 3 is localized using the nodes in $\{1, 2, 7\}$, node 4 is localized using the nodes in $\{2, 3, 7\}$, node 5 is localized using the nodes in $\{s, 4, 8\}$, and finally node 6 is localized using the nodes in $\{3, 4, 5\}$. It should be noted that node 8 does not have enough connectivity to be localized by trilateration only, therefore a visit to it is mandatory to localize it. If the mobile visits the nodes in the alternative order 1, 8, and 5, then another node 4 needs to be visited to kick start the trilateration. It can be easily observed that, when the mobile beacon visits node 6 instead of node 4, trilateration still cannot start. Therefore, assuming that each edge in Figure 1 has the same unit cost, the minimum distance to be traveled by the mobile beacon to localize all the nodes cannot be less than 3 units. The path $\{s, 2, 1, 8\}$ is hence optimal.

The contributions in this paper are listed as:

- We give a formal definition of a problem for the energy efficient use of a mobile beacon for trilateration based localization in WSNs.
- We prove, to the best of our knowledge, for the first time that the $MATBOL$ problem is NP-hard.
- We also give a compact Integer Linear Programming (ILP) formulation for $MATBOL$.

The rest of the paper is organized as follows. Section II presents the formal problem definition and the NP-hardness proof. An ILP formulation for the problem is given in Section III. Finally, Section IV concludes the paper along with a list of open problems.

II. Problem definition

We start by giving the formal definition of the $MATBOL$ problem below.

Definition II.1. *Given an undirected graph $G(V, E)$ with $w(e) \geq 0$ for all $e \in E$ corresponding to the distances as measured by the nodes in V, the objective of the* MATBOL *problem is to localize the nodes in G based on trilateration with the assistance of a mobile beacon situated initially at a designated node $s \in V$ so that the total distance traveled by the mobile gets minimized.*

We will prove that the *MATBOL* problem is NP-hard by a reduction from the All Colors Shortest Path (*ACSP*) problem that has been proved to be NP-hard and also not to lend itself to a constant factor polynomial time approximation unless $NP = P$ in [5]. The input to an instance of the *ACSP* problem is an undirected graph with a color assigned in advance to each node, and a non-negative weight for the edges. The objective of *ACSP* then is to find the shortest, possibly non-simple, path starting from a designated base node such that a node from every distinct color occurs at least once on the path.

We can describe a polynomial time reduction from a given instance of *ACSP* to the corresponding instance of *MATBOL*. Let us assume that the given instance of the *ACSP* problem is identified by the 4-tuple $< G(V, E), s, \kappa, w >$, where $s \in V$ is the base node, $\kappa : V \to C$ is a function assigning a color from $C = \{1, 2, ..., k\}$ to every node in V, and $w(\{i, j\}) \geq 0$ is the weight of an edge for all $\{i, j\} \in E$. We start modifying the given graph $G(V, E)$ to obtain a new graph $G'(V', E')$ by adding two dummy nodes $v^{(1)}$ and $v^{(2)}$ for each $v \in V$ ($V' = V \cup \{v^{(1)}, v^{(2)} | v \in V\}$). The nodes $v^{(1)}$ and $v^{(2)}$ are then both connected to the node v (also referred to as $v^{(0)}$) by an edge for all $v \in V$ ($E' = E \cup \{\{v, v^{(1)}\}, \{v, v^{(2)}\} | v \in V\}$). The color of the newly added nodes are set to be the same as the node to which they are connected ($\kappa(v^{(1)}) = \kappa(v^{(2)}) = \kappa(v^{(0)}), \forall v^{(0)} \in V$). The weights for the newly added edges are all finally set to 0 ($w(\{v^{(0)}, v^{(1)}\}) = w(\{v^{(0)}, v^{(2)}\}) = 0, \forall v^{(0)} \in V$). This transformation is shown in Figure 2 for a single node v.

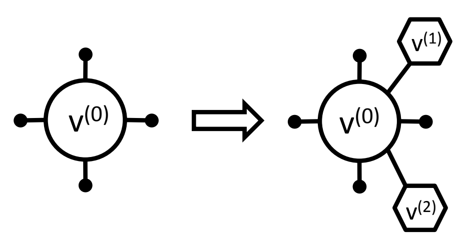

Fig. 2. Each node $v^{(0)}$ (or v) $\in G$ is connected to two dummy nodes $v^{(1)}$ and $v^{(2)}$ via edges of weight (distance) zero. The colors of these dummies are set to be the same as the color of $v^{(0)}$

In the next phase of the transformation process, a random cyclic order is created for every group of nodes with the same distinct color. This order is then used to simply enforce a trilateration ordering to make it possible to start from any node already localized with a particular color and then to localize the rest of the nodes with the same color without any help from the mobile. To this end, a random permutation $\pi_c =$

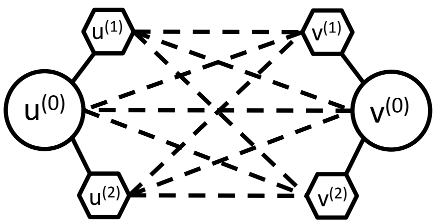

Fig. 3. $u = \pi_c(i)$ and $v = \pi_c((i \mod |g_c|) + 1)$. The dashed lines are the edges newly added. Note that $u^{(p)}$ and $v^{(q)}$ should all have the same color.

$\pi_c(1)\pi_c(2)...\pi_c(|g_c|)$ of the nodes in $g_c = \{v \in V|\kappa(v) = c\}$ is picked for each distinct color $c \in C$. Specifically, the edges $\{(\pi_c(i))^{(p)}, (\pi_c((i \mod |g_c|)+1))^{(q)}\}$, for a total of all 9 combinations of $p, q \in \{0, 1, 2\}$, are then added in E' with i ranging from 1 through $|g_c|$ for all colors $c \in C$. These newly added edges are drawn in Figure 3 as dashed lines. The weights of these edges are all set to the shortest distance in G between the consecutive nodes in the random order picked for each color $(w(\{(\pi_c(i))^{(p)}, (\pi_c((i \mod |g_c|) + 1))^{(q)}\})$ = the shortest distance between $(\pi_c(i))^{(0)}$ and $(\pi_c((i \mod |g_c|) + 1))^{(0)}$ in G). It should be noted at this stage that the required shortest distances between nodes can be computed by a single run of the All Pairs Shortest Path algorithm.

Finally, the mobile is placed at $s \in V'$, which happens to be the base node in the given instance of the *ACSP* problem.

The construction of the corresponding instance of the new graph G' and hence the entire transformation takes polynomial time.

Lemma II.2. *A given instance of* ACSP *has a solution with cost (length)* Δ *if and only if the corresponding instance of* MATBOL *obtained after the transformation has a solution with cost (total distance traveled)* Δ.

Proof. Let us first prove the only if part. Let us assume that the given instance of *ACSP* has a solution with cost (length) Δ. As this solution in G visits at least one node from every group of nodes with the same distinct color, the mobile starting at s travels over the exact same, possibly non-simple, path to localize all the nodes in G' by the trilateration orderings constructed in the transformation.

In order to prove the if part, we start by assuming that the total distance traveled by the mobile in the corresponding instance of the *MATBOL* is Δ. We can then claim that this very path traversed by the mobile visits every color (that is, a node of this color) at least once. This is justified by the observation that the dummies introduced in the transformation have no option but to be visited by the mobile in order to be localized unless a node in the cyclic order associated with the corresponding color has already been visited. □

Theorem II.3. MATBOL *is NP-hard.*

Proof. It is a direct consequence of Lemma II.2 and the observation that the transformation is polynomial in the size of the given *ACSP* instance. □

Even though the type of the underlying graph employed to represent WSNs might have implications on the intractability and computational characteristics of $MATBOL$, we claim that limiting the graph type to UDGs as generaly used in WSNs will not change the NP-hardness of the $MATBOL$ problem. The proof of that claim is left as an interesting open problem.

III. Integer linear programming formulation of $MATBOL$

In this section, we present an ILP formulation for the $MATBOL$ problem. Given an undirected weighted graph $G(V, E)$, and a designated base vertex $s \in V$ as an instance of the $MATBOL$ problem, we apply the following transformations to obtain a new directed weighted graph $G'(V', E')$ as follows:

- All vertices in G are copied to G' so that $V' = V$, where $|V| = n$.
- For each edge $\{\{i, j\} \in E \mid i, j \in V\}$ with weight $w(\{i, j\})$, an edge $e(i, j)$ with weight $w_{i,j}$, and a new edge $e(j, i)$ with weight $w_{j,i}$ are added to G', with $w_{i,j} = w_{j,i} = w(\{i, j\})$. G' is now a directed graph, obtained by replacing each edge in G by two directed edges with the same weight as the respective edge in G.
- For each vertex $v \in V$ in G, a new sink vertex v', and a new edge $e(v, v')$ with zero weight are added to the graph G'. Therefore, we ensure that a feasible path as a solution exists and always terminates at a sink vertex.
- A new source vertex 0 is added to G' such that $V' = V' \cup \{0\}$, and set the new base $s' = 0$. A new edge $e(0, s)$ with zero weight is also added from vertex 0 to vertex s in G'.

We are now ready to give an ILP formulation for $MATBOL$. The ILP formulation uses four classes of decision variables given as follows:

- $x_{i,j}$ represents whether the corresponding directed edge is visited or not, such that,

$$x_{i,j} = \begin{cases} 1, & \text{if edge } (i, j) \text{ is visited as part of the solution,} \\ 0, & \text{otherwise.} \end{cases}$$

- y_i represents whether the corresponding node is visited or not, such that,

$$y_i = \begin{cases} 1, & \text{if node } i \text{ is visited as part of the solution,} \\ 0, & \text{otherwise.} \end{cases}$$

- $f_{i,j}$ represents the flow value for the corresponding edge, such that,

$$f_{i,j} = \begin{cases} 1, \ldots, n+1, & \text{if edge } (i, j) \text{ is part of the solution,} \\ 0, & \text{otherwise.} \end{cases}$$

- $l_{i,k}$ represents the trilateration order of node i, such that,

$$l_{i,k} = \begin{cases} 1, & \text{if node } i \text{ is trilaterated at stage } k, \\ 0, & \text{otherwise.} \end{cases}$$

We give the objective function and the constraints of the ILP model as follows:

$$\text{minimize} \sum_{(i,j):(i,j) \in E} x_{i,j} * w_{i,j} \qquad (1)$$

subject to

$$\sum_{j:(j,i)\in E'} x_{j,i} = \sum_{j:(i,j)\in E'} x_{i,j} \quad , \forall i \in V \tag{2}$$

$$y_j \geq x_{i,j} \quad , \forall (i,j) \in E' \tag{3}$$

$$\sum_{i:(i,j)\in E'} x_{i,j} \geq y_j \quad , \forall j \in V' \setminus \{0\} \tag{4}$$

$$\sum_{j:(j,i)\in E'} f_{j,i} = y_i + \sum_{j:(i,j)\in E'} f_{i,j} \quad , \forall i \in V \tag{5}$$

$$x_{i,j} \leq f_{i,j} \leq (n+1) * x_{i,j} \quad , \forall (i,j) \in E' \tag{6}$$

$$y_i + \sum_{k=0}^{n-4} l_{i,k} \geq 1 \quad , \forall i \in V \tag{7}$$

$$3 * l_{i,k} \leq 3 * y_i + \sum_{j:(j,i)\in E} y_j \leq M + (2-M) * (1 - l_{i,k}) \quad , \forall i \in V, k = 0 \tag{8}$$

$$3 * l_{i,k} \leq 3 * y_i + \sum_{j:(j,i)\in E} l_{j,k-1} \leq M + (2-M) * (1 - l_{i,k}) \quad , \forall i \in V, k > 0 \tag{9}$$

$$x_{0,s} = 1 \tag{10}$$

$$x_{i,j} \in \{0,1\} \quad , \forall (i,j) \in E' \tag{11}$$

$$y_i \in \{0,1\} \quad , \forall i \in V' \setminus \{0\} \tag{12}$$

$$f_{i,j} \in \{0,1,\ldots,n+1\} \quad , \forall (i,j) \in E' \tag{13}$$

$$l_{i,k} \in \{0,1\} \quad , \forall i \in V, \forall k \in \{0,\ldots,n-4\} \tag{14}$$

The feasible solution to the $MATBOL$ problem is a, possibly non-simple, path that ensures all the vertices are localized either by a visit of the mobile, or by trilateration using already localized neighbours. The objective is to minimize the total distance as shown in (1). Constraints (2) assure that the number of the incoming and the outgoing edges on a feasible path are equal for all the vertices visited with the exception of the source vertex and the sink vertices. Therefore, the constraints are defined only on the vertices in V. Constraints (3) guarantee that a vertex j is visited if an incoming edge (i,j) is part of the solution. Constraints (4) assert that if the feasible path visits vertex j, then there should be at least one incoming edge (i,j) that gets visited. Constraints (5) and (6) eliminate subtours by introducing flow values for each edge visited as presented in [5]. Constraints (5) make sure that each vertex visited consumes a unit flow while Constraints (6) bound the maximum flow value carried on an edge. The rule to localize each vertex is either to visit it as part of the solution, or the vertex should have at least three already visited, hence localized, neighbors allowing for the trilateration of this vertex. The decision variable $l_{i,k}$ represents the trilateration order for vertex i such that $l_{i,k} = 1$ when vertex i is localized at stage k. Constraints (7) guarantee that all the vertices in V are localized either by a visit or by trilateration. As $|V| = n$, and trilateration needs at least three visited vertices to initiate, a maximum of $n-3$ stages are required by the trilateration algorithm. Constraints (8) and (9) ensure that each vertex is either visited by the mobile or has at least three visited neighbors. Therefore, vertices localized in the early stages of the trilateration are used to locate the unlocalized vertices in the later stages of the trilateration. M is a large value not any less than the maximum degree of all the nodes plus three. Constraint (10) initiates the path from the source node towards the base node so that the base node is always visited as part of the feasible path. Constraints (11), (12), (13), and (14) are integrality constraints for the decision variables used in the model.

The feasible solution to this ILP for $MATBOL$ gives a possibly non-simple path depicting the optimal path of the mobile beacon used in the problem.

IV. Conclusion

The difficulty of energy efficient trilateration with help from a mobile beacon in WSNs is formalized and investigated in this paper. This version of the localization problem introduced as $MATBOL$ is proved to be NP-hard. A compact ILP formulation is also provided for $MATBOL$. The paper is expected to pave the way for the development of new heuristic algorithms possibly with known approximation bounds. Proving the NP-hardness of $MATBOL$ when the underlying graph is an UDG is left as an interesting open problem.

V. Acknowledgement

This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) Career Grant no: 112E099.

References

[1] H. Akcan and C. Evrendilek, "GPS-free directional localization via dual wireless radios," *Computer Communications*, vol. 35, no. 9, pp. 1151 – 1163, 2012.

[2] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, "A theory of network localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, 2006.

[3] J. Aspnes, D. K. Goldenberg, and Y. R. Yang, "On the computational complexity of sensor network localization," in *First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS'04)*, Turku, Finland, July 2004, pp. 32–44.

[4] S. Bartoletti, A. Giorgetti, M. Z. Win, and A. Conti, "Blind selection of representative observations for sensor radar networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1388–1400, April 2015.

[5] Y. C. Bilge, D. Çagatay, B. Genç, M. Sari, H. Akcan, and C. Evrendilek, "All colors shortest path problem," *CoRR*, vol. abs/1507.06865, 2015.

[6] T. Eren, O. Goldenberg, W. Whiteley, Y. Yang, A. Morse, B. Anderson, and P. Belhumeur, "Rigidity, computation, and randomization in network localization," in *INFOCOM*, vol. 4, Hong Kong, China, March 2004, pp. 2673–2684.

[7] C. Evrendilek and H. Akcan, "On the complexity of trilateration with noisy range measurements," *Communications Letters, IEEE*, vol. 15, no. 10, pp. 1097 –1099, October 2011.

[8] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2220–2243, thirdquarter 2016.

[9] D. Koutsonikolas, S. M. Das, and Y. C. Hu, "Path planning of mobile landmarks for localization in wireless sensor networks," *Computer Communications*, vol. 30, no. 13, pp. 2577 – 2592, 2007.

[10] P. Perazzo, F. B. Sorbelli, M. Conti, G. Dini, and C. M. Pinotti, "Drone path planning for secure positioning and secure position verification," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[11] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. J. Teller, "Mobile-assisted localization in wireless sensor networks," in *INFOCOM*, Miami, FL, USA, March 2005, pp. 172–183.

[12] S. Rai B and S. Varma, "An algorithmic approach to wireless sensor networks localization using rigid graphs," *J. Sensors*, vol. 2016, pp. 3 986 321:1–3 986 321:11, 2016.

[13] J. B. Saxe, "Embeddability of weighted graphs in k-space is strongly np-hard." in *17th Allerton Conf. Commun. Control Comput.*, 1979, pp. 480–489.