A Recursive Shortest Path Routing Algorithm With Application for Wireless Sensor Network Localization

Juan Cota-Ruiz, Pablo Rivas-Perea, *Member, IEEE*, Ernesto Sifuentes, and Rafael Gonzalez-Landaeta

***Abstract*— In this paper, we present a routing algorithm useful in the realm of centralized range-based localization schemes. The proposed method is capable of estimating the distance between two non-neighboring sensors in multi-hop wireless sensor networks. Our method employs a global table search of sensor edges and recursive functions to find all possible paths between a source sensor and a destination sensor with the minimum number of hops. Using a distance matrix, the algorithm evaluates and averages all paths to estimate a measure of distance between both sensors. Our algorithm is then analyzed and compared with classical and novel approaches, and the results indicate that the proposed approach outperforms the other methods in distance estimate accuracy when used in random and uniform placement of nodes for large-scale wireless networks. Furthermore, the proposed methodology is suitable for the implementation in centralized localization schemes, such as multi-dimensional scaling, least squares, and maximum likelihood to mention a few.**

***Index Terms*— Shortest path, WSN localization, multi-hop Network.**

I. Introduction

WIRELESS sensor networks (WSNs) consist of many sensors or nodes randomly/spatially deployed or distributed over a certain area with the purpose of extracting, processing, transporting, and transmitting information about specific events or variables that could be present in the monitoring area. Being aware about such specific events requires nodes to be equipped with several modules such as sensors (i.e., for gathering environmental information), a processor and a power unit; however, to transport all gathered information through the network, sensors also require to be equipped with radio transceivers. Radio signals, unfortunately, cannot travel long distances, and sensors commonly are not able to reach or communicate with other sensors that are located far away of their own radio ranges. Furthermore, sharing information between two sensors in multi-hop networks, retransmissions among sensors are very common tasks. Therefore, efficient routing is an important issue in the network's performance [1], [2].

Manuscript received January 22, 2016; revised March 12, 2016; accepted March 13, 2016. Date of publication March 17, 2016; date of current version April 26, 2016. The associate editor coordinating the review of this paper and approving it for publication was Dr. Amitava Chatterjee.

J. Cota-Ruiz, E. Sifuentes, and R. Gonzalez-Landaeta are with the Department of Computer and Electrical Engineering, Autonomous University of Ciudad Juarez, Ciudad Juárez 32310, Mexico (e-mail: jcota@uacj.mx; esifuent@uacj.mx; rafael.gonzalez@uacj.mx).

P. Rivas-Perea is with the School of Computer Science and Mathematics, Marist College, Poughkeepsie, NY 12601 USA (e-mail: pablorp80@gmail.com).

Digital Object Identifier 10.1109/JSEN.2016.2543680

Finding the shortest path from a source to a destination place is an essential task in many disciplines such as transportation, computer science, geography, artificial intelligence and WSNs, to mention a few. Commonly, in multi-hop networks, routing algorithms are closely related to finding the shortest distance between a sender node (or source sensor) and a receiver node; therefore, finding the shortest path implies lesser retransmissions which directly affects the energy conservation of the entire network. The research to develop efficient shortest path algorithms has produced a variety of them, whose efficiency primarily depends on the computational performance. Nevertheless, until now, there is not clear which algorithm has the best performance to find the shortest path [3], [4]. For instance, Ji-Xian and Fang-Ling [5] claim that improves Dijkstra's algorithm (DA) by reducing the number of repeated operations and finding the minimum path length intuitively. In [6], an improvement of DA is also presented which reduces computations by about 8% when compared with the original DA. Here, the key point is to introduce an upper bound on the distance between two selected nodes. On the other hand, the authors of [7] propose a heuristic algorithm that chooses the minimum number of nodes between a source and destination nodes by selecting those nodes that are closer to an imaginary Euclidean distance generated by the two selected nodes; however, this approach assumes that positions of the source and destination nodes are known, which in most cases are unknown.

The shortest path can also be used to estimate true distances among non-neighboring sensor nodes in multi-hop networks, which is a fundamental part in centralized range-based localization algorithms. Among the most important advances in this area, we can find a novel distance calculation method among sensors is presented in [8] which improves the basic DA under irregular network topologies. This approach uses as key element the radio range of each sensor and a cosine formula to estimate distances between non-neighboring nodes. Also, Chen *et al.* [9] proposes a signal similarity-based localization (SSL) method. It uses the similarity of received signal strength between two neighboring sensors to obtain a new value (using either the Manhattan SSLM metric or the Euclidean distance SSLE metric) which relates these neighboring sensors. The process is repeated for all neighboring sensors to finally obtain estimated distances among non-neighboring sensors using a shortest path algorithm.

In a multi-hop range-based WSN localization, the goal is to estimate the location of all unknown sensors by using

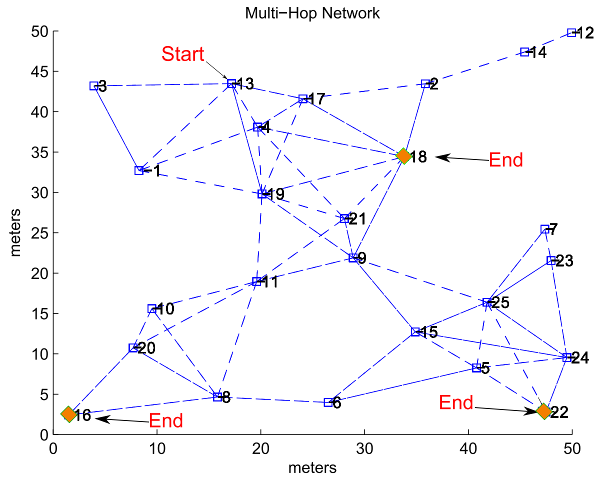

Fig. 1. Connectivity of a WSN with 22 unknown sensors, marked with '□', and three anchors, marked with '♦', all having a radio range of 15 m.

sensors with *apriori* known locations and partial information of the distances between some pair of sensors [10], [11]. For instance, let us assume the homogeneous WSN scenario of Fig. 1 which contains 22 non-located sensors (i.e., unknown sensors) '□' and three anchors (i.e., known sensors) '♦' uniformly and randomly distributed over a 50 by 50 m area where all sensors are of the same type and have a short range of 15 meters. Clearly, partial connectivity among sensors is present and distance estimates between unknown sensors and anchors might be multiple hops. If we consider that each sensor has the capacity to use the Received Signal Strength (RSS) (or any other) method for distance estimation [12], [13], then each unknown sensor would estimate its own location using distance estimates and absolute-positions to at least three anchors.

The sum-dist scheme [14] can be used to estimate distances between anchors and unknown sensors in distributed multi-hop networks. However, in centralized schemes [15]–[17] different shortest path algorithms must be used to estimate distances among non-neighboring sensors [8].

In Fig. 1, we have a static connected WSN with 22 unknown sensors unaware of their own location. To evaluate the location of any sensor, e.g. the unknown sensor s_{13}, it is necessary to estimate the true distances between s_{13} and anchors A_{16}, A_{18} and A_{22}. As described above, an intuitive approach would consists of applying a shortest path distance algorithm. However, such an approach does constraint the number of hops between an unknown sensor and an anchor sensor. Thus, when using noisy range measurements to estimate distances between neighboring sensors, the more hops used from the two-ended sensors, the higher the probability of increasing the distance error due to the propagation of errors in range measurements, assuming an independent identical distribution [18], [19].

The idea of this research is to provide a heuristic method that reduces the propagation error by minimizing the number of hops between two ended sensors and an averaging process. Thus, given two non-neighboring sensors, an unknown sensor and an anchor node, and using only the network connectivity, all shortest paths with the minimum number of hops must be solved where each found path is evaluated using a known weight cost to each hop in the path. Finally, all evaluated paths are averaged to obtain a final distance estimate. For instance, sensor s_{13}, depicted in Fig. 1, has two paths with four hops to anchor A_{16}: $|s_{13}, s_{19}, s_{11}, s_8, A_{16}|$ and $|s_{13}, s_{19}, s_{11}, s_{20}, A_{16}|$; three paths with two hops to anchor A_{18}: $|s_{13}, s_4, A_{18}|$, $|s_{13}, s_{17}, A_{18}|$ and $|s_{13}, s_{19}, A_{18}|$; and only one path with four hops to anchor A_{22}: $|s_{13}, s_{19}, s_9, s_{25}, A_{22}|$. Therefore, to estimate the true distance between sensor s_{13} and anchor A_{16}, the evaluation distance of two paths must be computed using the known weight for each hop in a path (i.e., neighboring range measurements). Then, the estimated distance from s_{13} to A_{16} is calculated as the mean of the two evaluated path distances. This process is repeated for remaining anchors of s_{13}. As a result, the position estimate of s_{13} can be computed using both the distance estimates and the absolute positions of anchors [20], [21]. As can be seen, positioning accuracy is largely dependent on the accuracy of distance estimation between unknown sensors and anchors.

In this paper, our approach is compared with traditional and novel approaches such as the DA [22], the Manhattan method [9] and the heuristic approach IMDS scheme [8]. Our experimental results show that the proposed method can also be used to successfully estimate true distances between non-neighboring sensors in multi-hop networks.

The outline of this paper is as follows. In Section II, we explore the SPP from an integer linear programming point of view. In Section III, we analyze in detail our proposed algorithm. In Section IV, we evaluate and compare the performance of the proposed methodology with novel approaches. Finally, we discuss our conclusions in Section V.

II. Problem Description

Assume a set of N sensors $\mathbf{S} = \{s_1, s_2, \ldots, s_N\}$, randomly deployed in a given geographical area. For range-based localization schemes in a 2-D scenario, it is common to consider that each sensor can calculate the distance to other sensors using the RSS technique [12], [13]. Thus, the range estimate between sensors s_i and s_j can be described as

$$r_{ij} = d_{ij} + \ell_{ij} \tag{1}$$

where d_{ij} represents the true distance, and ℓ_{ij} represents the distance error introduced by environmental factors. Also, due to a limited range of coverage, R, in each sensor (i.e., assumed circular and constant) a multi-hop network is formed where all sensors have a restricted number of neighboring sensors limited by R. Thus,

$$\mathbf{S}_i = \{j \mid d_{ij} < \mathrm{R}\} \tag{2}$$

defines the known neighboring sensors s_j of s_i. A major problem in a range-based multi-hop network localization is how to estimate distances between non-neighboring sensors (e.g., unknown sensors and anchors). A regular method to estimate such estimated distances is done by shortest path algorithms. Thus, we can formulate the SPP using a graph theory point of view, as described next.

A 2-dimensional WSN can be denoted as a weighted geometric graph $(\mathbf{V}, \mathbf{E}, \mathbf{L})$ where $(\mathbf{V}, \mathbf{E})$ is the graph $\mathbf{G}$, and

$\mathbf{L}$ is a map from $\mathbf{V}$ to $\mathbb{R}^2$. $\mathbf{V}$ is the set of vertices or sensor nodes $\mathbf{S}$, $\mathbf{E}$ is the set of edges or links among neighboring nodes, and $\mathbf{L}$ represents the set of locations of sensor nodes. Let $\{e_{ij}, e_{i(j+1)}, \cdots, e_{i(j+n)}\} \in \mathbf{E}$ be the set of edges where e_{ij} is the edge that connects s_i with s_j (or $s_j \in \mathbf{S}_i$), and this edge is associated with a length (or weight) r_{ij}. In this way, if there is one link between s_i and s_j, $r_{ij} = \|\mathbf{L}(i) - \mathbf{L}(j)\|$ $\forall (ij) \in \mathbf{E}$ where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^2$, and if there is no edge between s_i and s_j, $r_{ij} = \infty$ while for $i = j$, we will have $r_{ij} = 0$. Thus, $\mathbf{R}$ is the abutment matrix of the weighted graph. Also, it is common to assume $\mathbf{A}$, i.e., a_{ij} as the initialized adjacency matrix of $\mathbf{G}$ [6], [20], [23].

As it is well known, a path is composed by a set of distinct weighted edges that connect two specific sensors in the network. Thus, the route that provides the shortest path (i.e., less cost weight) between a source sensor s_s and a target sensor s_t in the network defines the shortest path. In a directed graph, when $r_{ij} = a_{ij} \forall (i, j) \in \mathbf{E}$, the SPP with the fewest hops $|\mathbf{P}_{st}|$ is posed as the following integer programming problem:

$$\begin{aligned} |\mathbf{P}_{st}| &= \min \sum_{ij \in \mathbf{E}} r_{ij} y_{ij} \\ & St: \ 0 \le y \le 1 \\ & \sum_i y_{si} - \sum_j y_{js} = 1 \\ & \sum_j y_{jt} - \sum_j y_{tj} = 1 \\ & \forall x \in \mathbf{V} \backslash \{s, t\} \rightarrow \sum_i y_{xi} - \sum_j y_{jx} = 0 \end{aligned} \tag{3}$$

where y is an integer variable, y_{si} represents an edge i leaving s, y_{js} is the edge j entering into s, and so on.

Let us to consider $|\mathbf{P}_{st}^k|$ as the k-th path with the minimum number of hops between s_s and s_t. Thus, the goal is to find M no-repeated paths with the minimum number of hops between s_s and s_t and evaluate the sum of the edge lengths in each path to finally obtain an estimated distance d_{st} as the average of the M found paths between s_s and s_t as follows:

$$d_{st} = \frac{1}{M} \sum_{k=1}^{M} |\mathbf{P}_{st}^k|. \tag{4}$$

III. The Recursive Shortest Path Algorithm (RSPA)

To estimate distances between two non-neighboring sensors s_s and s_t where $s_t \notin \mathbf{S}_s$, our algorithm follows two main steps. First, given the two sensors (or vertices) of a static and directed network (or graph), all possible paths with the minimum number of hops between s_s and s_t need to be obtained. Second, the algorithm evaluates each path assuming that every edge $e_{ij} \in \mathbf{E}$ belongs to the shortest path between s_s and s_t, so the length of the path is obtained based on the known one-hop distance, r_{ij} where $s_j \in \mathbf{S}_i$. Finally, the non-neighboring distance estimate between the two end sensors is calculated as the mean of all evaluated path distances as defined in (4). The proposed algorithm is divided into two sections. The first part uses a recursive approach, shown in Algorithm 1.

Algorithm 1 Obtains the Minimum Number of Hops Between the Source Sensor s_s and All Its Communicating Sensors

Require: s_s, **Hop**
Ensure: **Hop**
1: **Initialize:** static $c = 1, \mathbf{Seq}(j) = 0 \ \forall \ j = 1, 2, 3, \ldots, n$
2: sort($\mathbf{S}_s$) //Sort $\mathbf{S}_s$ from the minimum to the maximum according to hops
3: **for** $j = 1$ **to** length($\mathbf{S}_s$) **do**
4:   **if** ($\mathbf{Hop}(\mathbf{S}_s(j))$==0) **then**
5:     $\mathbf{Hop}(\mathbf{S}_s(j)) = \mathbf{Hop}(s_s) + 1$
6:   **else if** $|\mathbf{Hop}(\mathbf{S}_s(j)) - \mathbf{Hop}(s_s)| \le 1$ **then**
7:     continue
8:   **else if** $\mathbf{Hop}(\mathbf{S}_s(j)) < \mathbf{Hop}(s_s) - 1$ **then**
9:     $\mathbf{Hop}(s_s) = \mathbf{Hop}(\mathbf{S}_s(j)) + 1$
10:   **else**
11:     $\mathbf{Hop}(\mathbf{S}_s(j)) = \mathbf{Hop}(s_s) + 1$
12:     **if** $|\max(\mathbf{Hop}(\mathbf{S}_{\mathbf{S}_s(\mathbf{j})})) - \mathbf{Hop}(\mathbf{S}_s(j))| > 1$ **then**
13:       **Hop** = Algorithm1($\mathbf{S}_s(j)$, **Hop**)
14:     **end if**
15:   **end if**
16: **end for**
17: **if** $s_s \in$ **Seq then**
18:   return // return if the sensor $\mathbf{S}_s$ is already contained in the **Seq** array
19: **else**
20:   $\mathbf{Seq}(c) = s_s$ //Save the the current sensor
21:   $c = c + 1$
22: **end if**
23: **for** $\ell = 1$ **to** length($\mathbf{S}_s$) **do**
24:   **Hop** = Algorithm1($\mathbf{S}_s(\ell)$, **Hop**)
25: **end for**

This algorithm receives as arguments a sensor number s_s (i.e., initially the source sensor) and an array named **Hop** (i.e., initially a vector with n-zero elements) and returns the same vector that contains how many hops away (i.e., the minimum) from the source sensor s_s is every sensor in the network. If a sensor s_m has not wireless connections through multiple hops with the source sensor s_s, the vector **Hop** will contain zero at the m-index position.

To prevent an infinite recursion in Algorithm 1, the recursive process avoids to analyze the same sensor twice. It is done from line 17 to 22 using a vector **Seq** which saves a not analyzed sensor or return the process otherwise. On the other hand, the second section is also a recursive process shown in Algorithm 2. This code receives as arguments the vector **Hop**, obtained from Algorithm 1, a set of adjacent sensors (initially of the target node s_t), a value hop_number (initially the number of hops between the source and target sensor, s_s and s_t respectively), a vector $\mathbf{P}$ (initially a zero vector with hop_number elements) which stores a valid path between s_s and s_t.

As can be seen from Algorithm 2, the proposed approach uses a recursive process to find all possible paths with the minimum number of hops between two communicating end nodes. This algorithm finds all leaving edges starting at the

Algorithm 2 Obtains All Possible Paths With the Minimum Number of Hops Between a Source Sensor s_s and a Target Sensor s_t

Require: **Hops**, $\mathbf{S}_t$, hop_number, **P**, s_t
Ensure: $\mathbf{P}_{st}^{k}$ for $k = 1, \ldots, M$
1: **Initialize:** static $k = 1$
2: **for** $i = 1$ **to** hop_number-1 **do**
3: **for** $j = 1$ **to** length($\mathbf{S}_t$) **do**
4: **if** $\mathbf{Hops}\big(\mathbf{S}_t(j)\big) ==$ (hop_number $- i$) **then**
5: $\mathbf{P}\big((\text{hop\_number}) - i\big) = \mathbf{S}_t(j)$
6: **P** = Algorithm2 (**Hops**, $\mathbf{S}_{\mathbf{S_t(j)}}$, hop_number-$i$, $\mathbf{P}, \mathbf{S}_t(j)$)
7: **end if**
8: **if** (hop_number-i) ==1 **then**
9: $\mathbf{P}_{st}^{k} = \mathbf{P}$ //save the new path
10: $k = k + 1$
11: **end if**
12: **end for**
13: **end for**

target sensor s_t and steps backward by expanding each one of the child nodes in the searching process until the source sensor s_s is found. If a found child sensor has no children, the algorithm returns to the parent sensor to continue with the searching process. A child sensor will be accepted in a path only if it has a lesser number of hops from the source sensor than its parent sensor. When the algorithm finds a valid path, it stores the path in the vector **P**. Finally, all found paths are saved into the matrix $\mathbf{P}_{st}^{k}$, as shown in line 9. All paths of $\mathbf{P}_{st}^{k}$ for $k = 1, \ldots, M$ are evaluated using (4) to obtain an estimated distance between s_s and s_t.

IV. Experimental Results

As described earlier, positioning accuracy in WSNs depends on factors such as the quality of range measurements and the geometry of the network. In this section, we focus the analysis in how our approximation algorithm estimates distances between two non-neighboring sensors in large-scale uniformly and randomly distributed wireless sensor networks. These results are compared with other novel approaches such as the IMDS [8], the SSLM [9], and the traditional DA [15]. We test the performance of each algorithm by using a multi-hop network with noisy range-based measurements, so we simulate RSS noisy pair-wise distances among neighboring sensors as described in [12] with $\eta_p = 2.6$, $\sigma_{SH} = 6$ dB, and $P_0(d_0) = -40$ dBm for all sensors and anchors.

The IMDS method requires a shortest path route to recalculate a new one, so it can be applied to DA. Thus, to modify the shortest route provided by DA, the IMDS scheme follows the next steps:

1.- Given the shortest path route between a sender and a target sensor, each sensor that composes the route estimates distances with non-neighboring sensors using [8], so the connectivity network is modified.

2.- The algorithm DA is applied again to solve the SPP and find the new shortest route.

TABLE I
True Distance Estimation Between RSPA and Novel Schemes

Algorithm	Unknown Sensor	Anchor Node	True distance (m) Estimated distance (m)	Hops	Path(s)
	3	22	59.39		
DA			37.11	7	3-13-4-21-9-25-5-22
DA+IMDS			37.01	6	3-13-4-21-9-5-22
SSLM			60.33	6	3-1-19-9-15-5-22
RSPA			**58.84** (average)	**5**	
			60.04	5	3-1-19-9-25-22
			57.65	5	3-13-19-9-25-22
	7	16	51.33		
DA			54.05	5	7-25-9-11-20-16
DA+IMDS			**50.16**	**3**	7-9-11-16
SSLM			61.23	6	7-23-24-5-6-8-16
RSPA			56.64 (average)	5	
			59.35	5	7-25-5-6-8-16
			56.11	5	7-25-15-6-8-16
			57.05	5	7-25-9-11-8-16
			54.04	5	7-25-9-11-20-16

On the other hand, the SSLM scheme uses RSSI values to estimate relative distances among all neighboring sensors to finally estimate relative distances for non-neighboring sensors with a shortest path algorithm. As it is well known, averaging several measurements will always improve the precision of distance estimates, so for each pair of neighboring sensors, we average a round-trip distance measurements. However, the SSLM scheme compares the similarity of both distance measurements through the Manhattan method to obtain a new value that provides a weighted value for the two neighboring sensors. The process is repeated for all one-hop neighboring sensors, and then, the shortest path between two non-neighboring sensors is calculated with a routing algorithm which is basically the smallest accumulated SSLM along the path between the sender and the target sensor [9]. We use a threshold of 30 if there is not connectivity between two sensors during the SSLM process.

Fig. 1 shows an example of a small sensor network composed of 25 randomly deployed sensors where three are anchor nodes across a 2-D area of 50 m $\times$ 50 m. We assume that each unknown sensor can estimate distances with neighboring nodes using the RSS scheme. Also, an omnidirectional radio range R $= 15$ m is assumed in each sensor. Clearly, there is a path between every unknown sensor and an anchor node; however, we must consider that each path distance between both ended sensors contains errors given by noisy range measurements among neighboring sensors.

Table I shows a basic comparison between RSPA and the other approaches.

There are two cases to estimate true distances between two non-neighboring sensors for the network shown in Fig. 1. The first case is to find the true distance of 59.39 m between nodes 3 and 22. DA solves the SPP using seven hops with an estimated distance of 37.11 m, Dijkstra+IMDS (DA+IMDS) solves the SPP using six hops reducing the estimated distance to 37.01 m, the SSLM algorithm finds a route with also six hops but with a better estimated distance of 60.33 m, and RSPA uses only five hops with an estimated distance of 58.84 m, which is the average of the two routes found.

The second case has a true distance of 51.33 m between nodes 7 and 16. The DA+IMDS method will always take

TABLE II
SENSOR NETWORK TOPOLOGIES TO EVALUATE TRUE DISTANCES UNDER FOUR SCHEMES: DA, DA+IMDS, SSLM AND RSPA

	Number of Sensors	Area Deployment	Radio Range	Type of Distribution
(a)	100	200m x 200m	35	uniform and random
(b)	200	400m x 400m	45	uniform and random
(c)	400	800m x 800m	55	uniform and random

the shortest route of 50.16 m for this scenario while DA finds a shortest route (i.e., an estimated distance) of 54.05 m. For the SSLM scheme, six hops are required to obtain the smallest accumulated SSLM providing an estimated distance of 61.23 m. Finally, RSPA obtains an estimated distance of 56.64 m as the average of four shortest path routes. As can be seen from Table I, finding the shortest path distance does not guarantee to minimize the error between true distances and estimated distances among non-neighboring sensors. However, it will be interesting to analyze which methodology presents better results when the number of hops between non-neighboring sensors has increased in larger multi-hop networks, analyzed in the next paragraphs. To validate the global functionality of each algorithm, we created three WSNs with different network properties. We varied parameters like number of sensors, deployment area, and radio range for each sensor in the network. All nodes were randomly placed over a 2-D area, and we assumed that each sensor was equipped with an omnidirectional radio range; thus, to generate a multi-hop network, any sensor was able to estimate noisy distance measurements with other sensors within its radio range where the RSS parameters were the same as before described. Table II summarizes the three different sensor networks that are used to test the accuracy performance of each analyzed algorithm.

Fig. 2 shows an analysis involving the aforementioned algorithms to estimate true distances for 50 tests using the network of Table II(a). For each test, we randomly choose a source and a target sensor and evaluate distance estimates against true distances for each algorithm. Also, we present an analysis of the number of hops used by each approach to get such estimated distances. Fig. 2(a) shows that, in most cases, both algorithms DA and DA+IMDS tend to agree in distance estimates which are mostly below of the true distances. However, RSPA and SSLM have a better accuracy performance in distance estimates.

The root mean square error (RMSE), is used to test the accuracy performance of each algorithm.

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}\left(d_{st}^{i} - t_{st}^{i}\right)^{2}}, \quad (5)$$

where d_{st}^{i} and t_{st}^{i} represent the i-th estimated distance and true distance between two non-neighboring sensors, respectively; and T is the number of tests ($T = 50$).

Table III shows the RMSE of 50 tests for true distance estimation applied to each algorithm. Clearly, as predicted, DA has the worst accuracy performance to estimate distances between non-neighboring sensors, because it just finds the

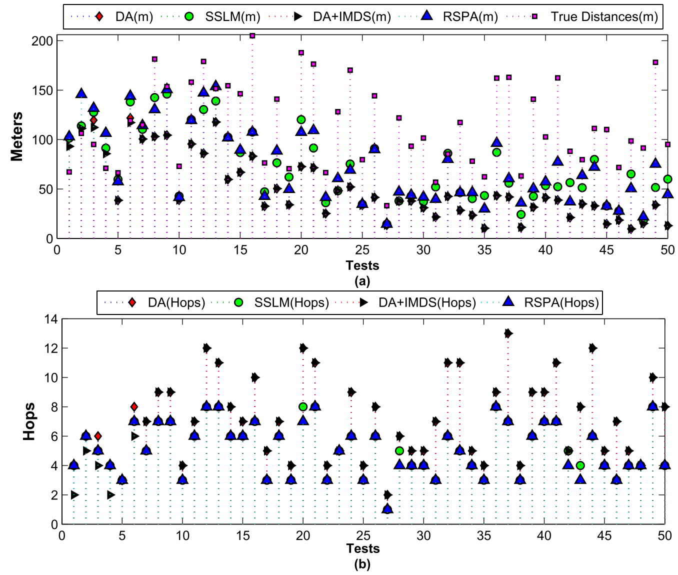

Fig. 2. (a) Estimated distances and (b) Used hops of DA, DA+IMDS, SSLM and RSPA when 50 tests to estimate true distances are applied to a multi-hop network of 100 sensors.

TABLE III
RMSE OF 50 TESTS TO EVALUATE THE ACCURACY PERFORMANCE OF FOUR ROUTING ALGORITHMS: DA, DA+IMDS, SSLM AND RSPA USING THE NETWORK OF TABLE II(a)

Algorithm	RMSE(m)
DA	75.85
DA+IMDS	75.71
SSLM	56.84
RSPA	54.5

shortest path distance without considering noisy in range measurements among sensors; however, this algorithm can be used as a complement to other schemes like IMDS and SSLM. On the other hand, RSPA and SSLM schemes show lower RMSEs than DA and DA+IMDS, but RSPA has slightly better accuracy estimation than SSLM.

Also, as can be observed in Fig. 2(b), RSPA and SSLM use fewer hops estimates than DA and DA+IMDS. For example, in average, DA requires 7.18 hops per test, DA+IMDS uses 7 hops per test, SSLM employs 5.16 while RSPA uses 5.08 per test. Moreover, since RSPA estimates distances using the average of n shortest paths, this is feature useful in other applications. Fig. 3 shows all alternative paths generated by RSPA to estimate true distances of 50 tests shown in Fig. 2. For instance, in the test number 12, RSPA averages 1069 different paths, all with 8 hops, to obtain an estimated distance of 147.2 m, SSLM has an estimated distance of 130.4 m with also 8 hops, DA and DA+IMDS obtains same values of 86.06 m with 12 hops.

We repeated the last procedure to analyze those algorithms with larger networks. Table IV summarizes final accuracy estimates (RMSEs) and number of hops (average) when algorithms solve 50 tests to estimate true distances for each network.

Evidently, adding more sensors in a network makes that combination paths between source and target sensors increase.

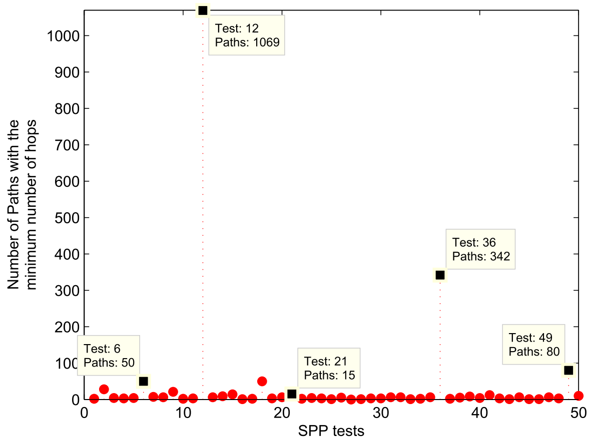

Fig. 3. Alternative paths of RSPA for tests of Fig. 2.

TABLE IV

A COMPARISON OF TRUE DISTANCE ESTIMATES BETWEEN RSPA AND NOVEL SCHEMES WHEN SENSORS ARE RANDOMLY AND UNIFORMLY DISTRIBUTED OVER A WIDE DEPLOYMENT AREA

Algorithm	Network of Table II(b)		Network of Table II(c)	
	RMSE(m)	hops (average)	RMSE(m)	hops (average)
DA	126.80	10.68	232.89	18.04
DA+IMDS	126.83	10.16	232.82	17.08
SSLM	86.57	8.28	177.82	14
RSPA	81.48	8.06	154.67	13.6

Thus, the number of alternative paths with the minimum number of hops offered by RSPA also tends to increase. Hence, the average of all alternative path distances tend to reduce random errors of range measurements. Moreover, the error propagation of noisy range measurements is reduced when the number of hops in a path is also decreased. Hence, our approach provides more accurate distance estimates.

V. CONCLUSIONS

In this research, we have proposed a recursive algorithm to estimate distances between any two sensors. The algorithm finds all possible combination routes with the minimum number of hops between a sender and a target node. To find all possible routes between two sensors, the algorithm uses a data structure in each sensor that contains all neighboring sensors that are at one-hop of distance. In the searching process, each child node is expanded going forward looking for a target node. If an expanded node has no children, the searching process returns back to the parent node to continue exploring new sensors. After that, the algorithm evaluates the path distance of each found route with a weighted distance matrix. Finally, a distance estimate is computed as the mean of all path distances.

The proposed algorithm is analyzed and compared with classical and novel approaches over multi-hop networks when noisy range measurements among neighboring sensors are present. Experimental results indicate that our algorithm provides distance estimates with low estimation error. Moreover, due the nature of this approach to provide all multiple-trajectories between two non-neighboring nodes with the minimum number of hops, our method can be easily applied in a variety of fields, i.e., transportation, vehicle routing, web mapping, communications, geography, artificial-intelligence, and/or GIS-Network analysis, to name only a few.

ACKNOWLEDGMENT

The authors want to express their appreciation to the anonymous reviewers for their useful remarks.

REFERENCES

[1] S. Tomic and I. Mezei, "Improved DV-Hop localization algorithm for wireless sensor networks," in *Proc. IEEE 10th Jubilee Int. Symp. Intell. Syst. Inform. (SISY)*, Sep. 2012, pp. 389–394.

[2] M. S. Elgamel and A. Dandoush, "A modified Manhattan distance with application for localization algorithms in ad-hoc WSNs," *Ad Hoc Netw.*, vol. 33, pp. 168–189, Oct. 2015.

[3] R. Lachowski, M. E. Pellenz, M. C. Penna, E. Jamhour, and R. D. Souza, "An efficient distributed algorithm for constructing spanning trees in wireless sensor networks," *Sensors*, vol. 15, no. 1, pp. 1518–1536, 2015.

[4] D.-B. Zou and Y.-B. Wang, "Adaptive energy-aware routing framework in transmission cost constrained wireless sensor networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 534–538.

[5] J.-X. Xiao and F.-L. Lu, "An improvement of the shortest path algorithm based on Dijkstra algorithm," in *Proc. 2nd Int. Conf. Comput. Autom. Eng. (ICCAE)*, vol. 2. Feb. 2010, pp. 383–385, doi: 10.1109/ICCAE.2010.5451564.

[6] Z. Zhen, W. Jigang, and X. Duan, "Practical algorithm for shortest path on transportation network," in *Proc. IEEE Int. Conf. Comput. Inf. Appl. (ICCIA)*, Tianjin, China, Dec. 2010, pp. 48–51.

[7] I. Banerjee, I. Roy, A. R. Choudhury, B. D. Sharma, and T. Samanta, "Shortest path based geographical routing algorithm in wireless sensor network," in *Proc. Int. Conf. Commun., Devices Intell. Syst. (CODIS)*, Kolkata, India, 2012, pp. 262–265.

[8] B. R. Stojkoska, "Nodes localization in 3D wireless sensor networks based on multidimensional scaling algorithm," *Int. Scholarly Res. Notices*, vol. 2014, Jan. 2014, Art. no. 845027.

[9] P. Chen, H. Ma, S. Gao, and Y. Huang, "SSL: Signal similarity-based localization for ocean sensor networks," *Sensors*, vol. 15, no. 11, pp. 29702–29720, 2015. [Online]. Available: http://www.mdpi.com/1424-8220/15/11/29702

[10] B. D. O. Anderson, I. Shames, G. Mao, and B. Fidan, "Formal theory of noisy sensor network localization," *SIAM J. Discrete Math.*, vol. 24, no. 2, pp. 684–698, 2010. [Online]. Available: http://dx.doi.org/10.1137/100792366

[11] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128606003227

[12] J. Cota-Ruiz, J.-G. Rosiles, P. Rivas-Perea, and E. Sifuentes, "A distributed localization algorithm for wireless sensor networks based on the solutions of spatially-constrained local problems," *IEEE Sensors J.*, vol. 13, no. 6, pp. 2181–2191, Jun. 2013.

[13] S. Tomic, M. Beko, R. Dinis, G. Dimic, and M. Tuba, "Distributed RSS-based localization in wireless sensor networks with node selection mechanism," in *Technological Innovation for Cloud-Based Engineering Systems* (IFIP Advances in Information and Communication Technology), vol. 450, L. M. Camarinha-Matos, T. A. Baldissera, G. Di Orio, and F. Marques, Eds. Springer, 2015, pp. 204–214. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16766-4_22

[14] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Comput. Netw.*, vol. 43, no. 4, pp. 499–518, 2003.

[15] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proc. ACM Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, Annapolis, MD, USA, 2003, pp. 201–212. [Online]. Available: http://dx.doi.org/201-212.10.1145/778415.778439

[16] A. A. Kannan, G. Mao, and B. Vucetic, "Simulated annealing based wireless sensor network localization," *J. Comput.*, vol. 1, no. 2, pp. 15–22, 2006.

[17] C. Alippi and G. Vanini, "A RSSI-based and calibrated centralized localization technique for wireless sensor networks," in *Proc. 4th IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PERCOMW)*, Mar. 2006, pp. 301–305. [Online]. Available: http://dx.doi.org/10.1109/PERCOMW.2006.13

[18] B. Huang, C. Yu, and B. D. O. Anderson, "Analyzing error propagation in range-based multihop sensor localization," in *Proc. 48th IEEE Conf. Decision Control, Held Jointly 28th Chin. Control Conf. (CDC/CCC)*, Dec. 2009, pp. 865–870.

[19] T. Kim, M. Shon, M. Kim, D. S. Kim, and H. Choo, "Reducing error propagation on anchor node-based distributed localization in wireless sensor networks," in *Ubiquitous Computing and Multimedia Applications*. Berlin, Germany: Springer, 2011, pp. 165–174.

[20] J. Aspnes *et al.*, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, Dec. 2006.

[21] A. ElAssaf, S. Zaidi, S. Affes, and N. Kandil, "Low-cost localization for multihop heterogeneous wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 472–484, Jan. 2016.

[22] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.

[23] B. D. Anderson *et al.*, "Graphical properties of easily localizable sensor networks," *Wireless Netw.*, vol. 15, no. 2, pp. 177–191, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11276-007-0034-9

Juan Cota-Ruiz received the Ph.D. degree in electrical and computer engineering from the University of Texas at El Paso in 2011. He has been a Full Research Professor with the Autonomous University of Ciudad Juárez since 2003. He has national and international publications in the field of electrical engineering, and he holds a national patent. His current research interests include wireless sensor networks, smart sensors, numerical optimization, and digital signal processing.

Pablo Rivas-Perea (S'03–M'12) received the Ph.D. degree in electrical and computer engineering from the University of Texas at El Paso in 2011. He was with the industry for eight years. He was a Post-Doctoral Fellow and an Adjunct Professor of Computer Science at Baylor University. He was with the NASA Goddard Space Flight Center, and he is currently an Assistant Professor of Computer Science at Marist College. His current research interests include data science, machine learning, soft computing, big data analytics, and image processing.

Ernesto Sifuentes was born in Durango, Mexico, in 1976. He received the B.S. degree from the Technological Institute of Durango, Durango, Mexico, in 2000, the M.S. degree from the Technological Institute of Chihuahua, Chihuahua, Mexico, in 2002, and the Ph.D. degree from the Universitat Politcnica de Catalunya, Barcelona, in 2009, all in electronics engineering. Since 2002, he has been a Titular Professor with the Department of Computer and Electrical Engineering, Autonomous University of Ciudad Juárez, Juárez, Mexico. His current research interests include sensor interface circuits, virtual instrumentation, embedded systems, autonomous sensors, and wireless sensor networks.

Rafael Gonzalez-Landaeta received the Electronics Engineering degree from Rafael Belloso Chacín University, Maracaibo, Venezuela, in 1997, and the Ph.D. degree in biomedical engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 2008. He has been with the Francisco de Miranda University, Coro, Venezuela, since 1999, teaching courses in analog electronic and biomedical sensors. He is currently with the Department of Computer and Electrical Engineering, Autonomous University of Ciudad Juárez, Mexico. His current research areas include biomedical sensors and analog signal processing, noise and interference in electronic circuits, and noninvasive physiological measurement.