Minimum Cost Localization Problem in Three-Dimensional Ocean Sensor Networks

Chao Zhang* Yingjian Liu† Zhongwen Guo† Guodong Sun‡ Yu Wang*

* Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA.

† Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China.

‡ School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China.

***Abstract*—Localization is one of the most fundamental problems in ocean sensor networks. Current localization algorithms mainly focus on how to localize as many sensors as possible given a set of mobile or static anchor nodes and distance measurements. In this paper, we consider the optimization problem, *minimum cost localization problem* in a 3D ocean sensor network, which aims to localize all underwater sensors using the minimum number of anchor nodes or the minimum travel distance of the ship which deploys and measures the anchors. Given the hardness of 3D localization, we propose a set of greedy methods to pick the anchor set and its visiting sequence. Aiming to minimize the localization errors, we also adopt a confidence-based approach for all proposed methods to deal with noisy ranging measurements and possible flip ambiguity. Our simulation results demonstrate the efficiency of all proposed methods.**

I. Introduction

Localization is one of the most fundamental tasks in designing ocean sensor networks (OSNs) [1], [2] or general wireless sensor networks [3]. Location information can be used in many tasks of OSNs such as event detecting, target/device tracking, environmental monitoring, tagging raw sensing data, and network deployment. Moreover, location information can also be used by networking protocols (e.g., position-based routing or geometric topology control) to enhance the performance of OSNs. However, it is more challenging to locate nodes in underwater environments than in terrestrial environments. First, GPS signal does not propagate through water and RF signal cannot be used since it will attenuate very rapidly in water. Thus, acoustic signal is usually the best choice in underwater environments. Second, several alternative cooperative positioning schemes are not applicable in practice due to acoustic channel properties (such as low bandwidth, high propagation delay and high bit error rate). Since the velocity of acoustic signal can change with salinity, pressure and temperature, it is difficult to get quite precise ranges between nodes underwater. Last, 3D deployment of OSN requires more anchor nodes to locate nodes in 3D ocean space. All these make accurate localization in ocean a challenging task.

The work of C. Zhang and Y. Wang was supported in part by the US National Science Foundation under Grant No. CNS-1319915 and CNS-1343355. The work of Y. Liu was partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61074092. The work of Z. Guo was partially supported by the NSFC under Grant No. 61170258 and 61379127. The work of G. Sun was partially supported by the NSFC under Grant No. 61300180. This work was partially done when Y. Liu visited Department of Computer Science, the University of North Carolina at Charlotte, with a scholarship from the China Scholarship Council. Y. Wang (yu.wang@uncc.edu) is the corresponding author.

A large number of localization techniques [4]–[7] have been proposed for OSNs to localize underwater sensors by exchanging information with anchor nodes. Utilizing time of arrival (ToA) [4], [5] or time difference of arrival (TDoA) [6], [7], these methods can estimate distances between nodes and then compute positions of nodes based on these distances. Usually, a certain amount of anchor nodes are deployed, whose positions are known beforehand or can be measured by sea surface buoys or vessels. Most of existing methods try to localize sensors without the guarantee of covering all sensors. They usually rely on one hypothesis that there are enough anchor nodes to achieve the goal. Recently, Huang *et al.* [8] introduce a new localization problem, called *minimum cost localization problem* (MCLP), for 2D sensor networks which aims to localize all nodes in a network using the minimum number of anchor nodes. A set of greedy algorithms using both trilateration and local sweep operations [8] and a genetic algorithm [9] have been proposed to address this problem in 2D networks. In this paper, we further extend and enhance the optimization problem to 3D OSNs (Section II). Note that the cost of manually configuring an anchor node in ocean is expensive and GPS device does not work well underwater. Therefore, we rely on a ship visit to deploy and find the locations of anchor nodes. In addition, we also include a new variation of MLCP where the length of visited path for the ship is considered as the optimization objective. For both versions of MCLP, we propose multiple greedy algorithms to localize all underwater sensors with the minimum number of anchors or the minimum travel distance of the ship (Section III). In addition, to handle localization errors caused by both ranging errors and flip ambiguity, we introduce a confidence-based approach to control errors induced in localization process (Section IV). Our simulation results (Section V) demonstrate the efficiency of all proposed methods.

II. Minimum Cost 3D Localization Problem

A. Models and Assumptions

The 3D OSN can be modelled as a graph $G = (V, E)$, in which V is the set of nodes representing underwater sensor nodes and E is the set of links connecting sensor nodes if they are within each other's sensing range. Here, we assume all sensor nodes are static, ignoring the movement caused by

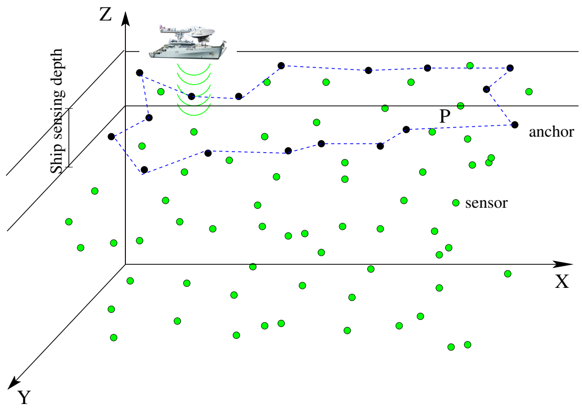


Fig. 1. **Illustration of localization scenario in ocean sensor networks:** black nodes are selected anchors from the shallow sensors, while other green nodes are localized via 3D localization method. P represents the travelling path of the ship to visit all anchors.

subsurface current. We also assume that the sensing range of all sensors are same. A subset of sensor nodes $B \subset V$ will be defined as anchor nodes whose positions need to be known at the beginning of localization (i.e., deployed and measured by a ship visit). Note that GPS devices do not work under water. The remaining widely distributed underwater sensor nodes will rely on distance measurements in E and the positions of anchor nodes B to determine their locations during the localization procedure. Underwater sensor nodes maybe distributed in different depths, but only those shallow sensor nodes within a ship's sensing range can be captured. Thus, we assume that only the shallow sensors can be considered to become anchor nodes. We use $V' \subset V$ to denote the set of shallow sensor nodes. See Fig. 1 for illustration. Note that it is possible that some underwater sensor nodes cannot be localized even though all shallow sensor nodes become anchors. Therefore, we assume that V only contains sensors who are localizable when all shallow sensors are set as anchors.

B. The MCLP Problems

The main purpose of the minimum cost localization problems (MCLPs) in a 3D OSN is to localize all underwater nodes using the minimum cost for anchor nodes. The cost includes the equipment cost and any costs during deployment and measurement. Similar to [8], by assuming a unit cost per anchor node, we can define the following 3D MCLP problems:

Definition 1: **Minimum Cost Localization Problem 1 (MCLP-1):** Given a 3D ocean sensor network G, find a subset B of shallow sensor nodes to be anchor nodes such that (1) all sensor nodes in V can be localized given the graph, the lengths of all links, and positions of all anchor nodes; and (2) the total number of anchor nodes $|B|$ is minimized.

In addition to the cost per anchor node, the cost of deployment and measurement for all anchors may depend on the length of the route P travelled by the ship (shown in Fig. 1). In that case, the MLCP problem can be defined as follows:

Definition 2: **Minimum Cost Localization Problem 2 (MCLP-2):** Given a 3D ocean sensor network G, find a subset B of shallow sensor nodes to be anchor nodes such that (1) all sensor nodes in V can be localized given the graph, the lengths of all links, and positions of all anchor nodes; and (2) the total length of the route $||P||$ taken by the ship to visit all anchors is minimized.

It is clear that both MCLPs always have a feasible solution, since based on our assumption in the worst case every shallow sensors are selected as anchors (i.e., $B = V'$) and all sensors in V will be localized. However, finding the optimal solution of such problems is still very challenging. Since the 2D MCLP [8] is a special case of MCLP-1 and it is NP-hard, MCLP-1 is also NP-hard. In MCLP-2, without considering the localization part, finding the minimum length route to visit all anchors alone is the well-known travel salesman problem (TSP), which is NP-hard. Therefore, MCLP-2 is also NP-hard.

III. Min-Cost 3D Localization Algorithms

In this section, we introduce different greedy algorithms to approximately solve MCLP problems by finding the anchor set. All proposed greedy algorithms use a simple color coding, where each sensor node v will be marked with different colors. We use $s(v)$ to denote v's color and status. A white sensor node represents the node which has not been localized yet; a black sensor node denotes the selected anchor node which can be directly contacted by the ship to get its position; a green sensor node is a non-anchor node whose location can be obtained via localization. Initially, all sensor nodes are white. The purpose of our algorithms is to find the smallest set of anchors (black nodes) or the anchor set with the minimum travelling distance to get the positions of all nodes (coloring all nodes in either black or green) as shown in Fig. 1.

A. Algorithms for MCLP-1

For solving the 3D MCLP-1, we simply extend two proposed methods in [8] for 2D MCLP (one uses trilateration and the other uses local sweep for localization) to 3D networks. Both methods use a general greedy framework as shown in Algorithm 1. The basic idea is as follows. Initially all sensor nodes will be colored as white and with its rank $r(v) = 0$ (Line 1-2). Here, node rank $r(v)$ of a node v indicates the total count of localized neighbors (with color of black or green). In addition, for any sensor nodes with less than four neighbors will be marked as black (Line 3-5), since they cannot be localized by others. Note that based on our assumption, all these nodes are shallow sensor nodes.Then in each step (Line 6-13) the algorithm greedily picks one white sensor node which can benefit the localization procedure most in next step if it is marked as black, and colors it as black. Here we define the benefit of marking a node v black as $g(v)$ which is the number of newly marked green nodes if v is marked as black. This procedure terminates until there is no white node and all black nodes are the selected anchor nodes by the algorithm. In Algorithm 1, when a node is marked as black or green, a function MARK is called, which is a recursive process of localization based on the underline localization method (trilateration or local sweep).

Algorithm 1 Greedy Algorithm for MCLP-1

```
1: for each v ∈ V do
2:     s(v) = white and r(v) = 0.
3: for each v ∈ V do
4:     if degree of (v) ≤ 3 then
5:         MARK(v, black).
6: while ∃v whose s(v)=white do
7:     for each v whose s(v)=white do
8:         Backup current status of all nodes.
9:         MARK(v, green).
10:        g(v) = total number of newly added green nodes.
11:        Restore status of all nodes.
12:    Let v_max is the white node with the maximum g(v).
13:    MARK(v_max, black).
14: return all black nodes as the selected anchors.
```

Algorithm 2 MARK(u, color) based on Trilateration

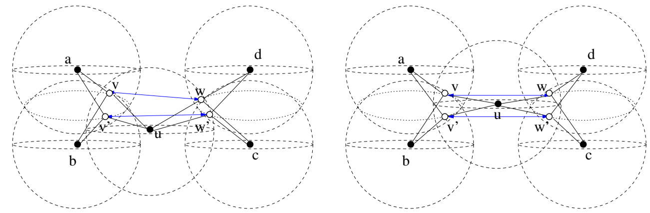
```
1: s(u)=color.
2: for each v of u's white neighbor do
3:     r(v)++.
4: for each v of u's white neighbor with r(v) ≥ 4 do
5:     MARK(v, green).
```

Algorithm 3 MARK(u, color) based on Local Sweep

```
1: s(u)=color.
2: for each v of u's white neighbor do
3:     r(v)++.
   {Phase 1: trilateration}
4: for each v of u's white neighbor with r(v) ≥ 4 do
5:     MARK(v, green).
   {Phase 2: local sweep within two-hop neighborhood}
6: if two white neighbors of u (say v and w) both have ranks
   of 3 and are neighbor to each other then
7:     if both v and w have unique positions which are
       consistent with distance measurement then
8:         MARK(v, green) and MARK(w, green).
   {Phase 3: local sweep within three-hop neighborhood}
9: if a white neighbor v of u and another white neighbor w
   of v both have ranks of 3 then
10:    if both v and w have unique positions which are
       consistent with distance measurement then
11:        MARK(v, green) and MARK(w, green).
```

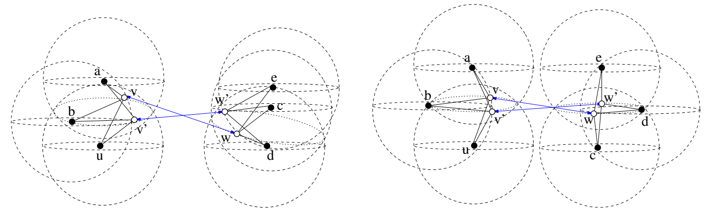
1) *MARK based on Trilateration:* Algorithm 2 shows the mark function for multilateration (iterative trilateration). When a node is marked as black or green, all its white neighbors' rank status need to be increased one. If its white neighbor's rank reaches 4, this neighbor can be marked as green too.

2) *MARK based on Local Sweep:* Trilateration has its own limitation as discussed in [10]. Similar to 2D methods proposed in [8] and the localization method by [11], we can use sweep operations to improve the localization by checking the consistency of possible positions of nodes in a local neighborhood and localizing more nodes if possible. When two neighboring nodes whose ranks are both 3 (i.e., the possible positions of each of them are limited to two locations), the distance between these two nodes can be used to eliminate the bogus positions. Note that it is possible that a unique match cannot be found, then these two nodes cannot be realized by the local sweep. To reduce the overhead, we limit the sweeps within two-hop or three-hop ranges. Fig. 2 and Fig. 3 illustrate examples for all cases. The detailed MARK procedure is given in Algorithm 3.



(a) unique match (b) no unique match

Fig. 2. **Local sweep within two-hop:** (a) there is a unique match; (b) there is no unique match. Here both v and w are u's neighbors with rank of 3.



(a) unique match (b) no unique match

Fig. 3. **Local sweep within three-hop:** (a) a unique match; (b) no unique match. v is u's neighbor and w is v's neighbor, and both have rank of 3.

Algorithm 4 Greedy Algorithm for MCLP-2

1-9: same as Algorithm 1.

10 : $g(v) = \frac{\textit{total number of newly added green nodes}}{\textit{distance from } v \textit{ to the last selected black node}}$.

11-14: same as Algorithm 1.

B. Algorithms for MCLP-2

In MCLP-1, the ultimate goal is to minimize total number of anchors in 3D OSNs. However, the length of the route P travelled by the ship which deploys and measures all anchors can also lead to time and financial cost. Therefore, in MCLP-2, we focus on minimizing the travelling distance of the ship instead. Obviously, we can still use the greedy algorithms proposed for MCLP-1 and let the selection order of anchors be the visiting sequence of anchors by the ship. Naturally less selected anchors results in shorter travelling distance of the ship. Therefore, the greedy method based on local sweep may give better performance than the one based on trilateration.

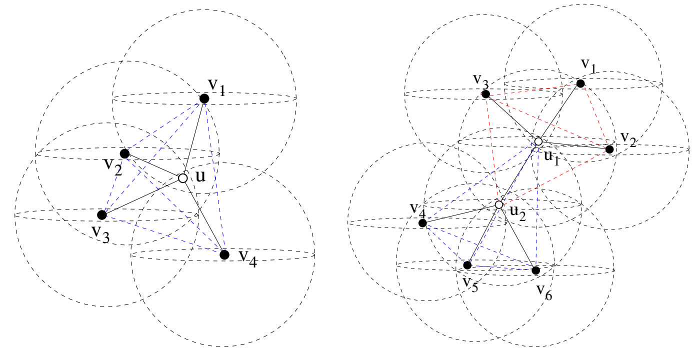
Since all of previous greedy algorithms do not consider the travel distances among selected anchor nodes, one possible improvement could be including the distance metric in the selection criteria of our greedy algorithm. Basically in each step, we can select the next white node, which yields the best ratio between the gain of number of green nodes and the newly added travel distance (from this node to the previous anchor), as the next anchor. See Algorithm 4 for the detailed algorithm.

Moreover, once all anchors (black nodes) are selected by above methods, further optimization on the travel distance can be performed. Recall that travelling salesman problem (TSP) aims to minimize the travel distance while visiting each node exactly once. TSP is a well-known NP-complete problem and cannot be easily solved to find the optimal solution, while there are many existing approximation algorithms or heuristics which work well in practice. We can use all anchors found by our greedy algorithms as the input of TSP algorithm to find the visiting sequence of the ship. This can further reduce the final length of its visiting path.

IV. Min-Cost 3D Localization with Errors

So far, we assume that there is no error in distance measurements and the calculated positions via trilateration or other localization methods are accurate once the node is localizable. However, this is not the case in real world, especially for OSNs. First, due to high delay and error rate of acoustic channel and various velocities of acoustic signals with changing salinity, pressure and temperature, it is difficult to get precise distance measurements between underwater nodes in the ocean. Second, even when the distance measurement error is minimized, flip ambiguity can still result in totally contrary results as demonstrated by [12]. Last, since most of the underwater nodes need to be localized via other nodes using iterative localization methods, any local errors can propagate to a large region, accumulate through the localization process, and induce huge deviations from the correct positions. This is especially true for large-scale networks. Therefore, we need to consider how to handle the localization errors caused by both ranging errors and flip ambiguity.

Similar to [12] and [13], we adopt a confidence-based approach to control errors induced in localization process. For each sensor node v, we define a node confidence $c(v)$, which indicates the probability of its accurate localization. The confidence $c(v)$ depends on *the confidences of reference nodes* of v and *the layout of the reference nodes*. Here, we use reference nodes to denote the localized nodes (those marked as green or black) based on which we can locate other nodes by trilateration. For example, in Fig. 4(a), a white node u can be localized via four reference nodes ($v_1, \cdots, v_4$, all marked as green or black nodes). Similar to [13], we use the width of the reference node set to estimate the confidence of trilateration, since a narrow width of reference nodes may lead to higher probability of ambiguity and errors. Here, the width of four reference nodes can be defined as the minimal height of tetrahedron $v_1v_2v_3v_4$ (marked as blue in the figure) which depends on the layout of reference nodes. To normalize the confidence of trilateration into range $[0, 1]$, we divide the width of reference nodes to the height of a regular tetrahedron with its size length equal to the sensing range. Let $c_t(u, v_1, v_2, v_3, v_4)$ represent the confidence of trilateration at u by using $v_1, \cdots, v_4$ as the reference nodes. Then the confidence of u is given by $c(u) = \min\{c(v_1), c(v_2), c(v_3), c(v_4), c_t(u, v_1, v_2, v_3, v_4)\}$. In



(a) trilateration (b) local sweep

Fig. 4. **Calculation of Node Confidence:** (a) u's confidence depends on its four references' confidences and the confidence of trilateration (from blue tetrahedron); (b) confidences of u_1 and u_2 have the same value and depend on confidences of six references and the confidences of trilateration (from both red and blue tetrahedrons).

other words, the confidence of node u is the minimum value among the confidences of its reference nodes and the confidence of trilateration. By this definition, the selection of its reference nodes influences the confidence of a node. We always assume that the confidence of an anchor node is 1.

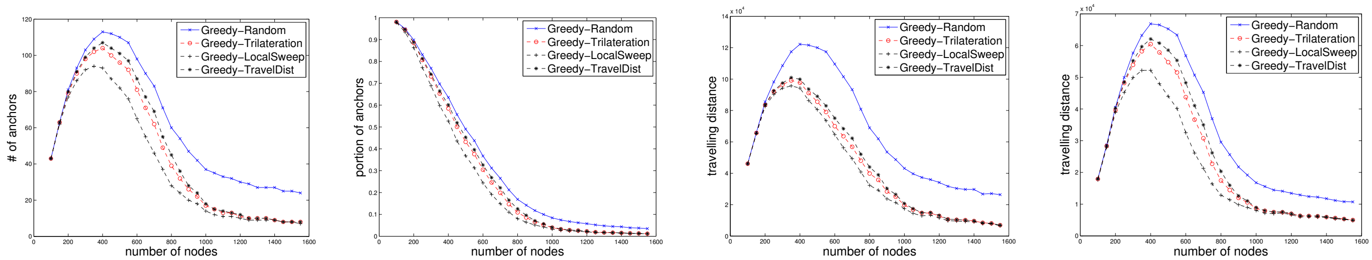
During a local sweep, two nodes are localized via five to six reference nodes. Fig. 4(b) shows an example for local sweep within three-hop[1]. In this example both u_1 and u_2 can be localized via six references $v_1, \cdots, v_6$. Since the positions of u_1 and u_2 depend on each other, thus $c(u_1) = c(u_2)$. The confidence of trilateration $c_t(u_1, u_2, v_1, v_2, v_3, v_4, v_5, v_6) = min\{c_t(u_1, u_2, v_1, v_2, v_3), c_t(u_2, u_1, v_4, v_5, v_6)\}$, i.e., the minimum one of the confidences of two tetrahedrons ($u_2v_1v_2v_3$ in red and $u_1v_4v_5v_6$ in blue). Then the node confidences of u_1 and u_2 are given by $c(u_1) = c(u_2) = \min\{c(v_1), \cdots, c(v_6), c_t(u_1, u_2, v_1, \cdots, v_6)\}$.

By having the node confidence, we can modify all proposed methods by requiring a minimum confidence threshold α. A node v can be localized via either trilateration (Line 4 in Algorithm 2 and Line 4 in Algorithm 3) or local sweep (Line 6 and Line 9 in Algorithm 3) if and only if $c(v) \geq \alpha$. A confidence calculation and check procedure is introduced into those lines before the node(s) can be marked as localized. If $c(v) < \alpha$, this node needs to wait for more reference nodes (green nodes) in its neighbors. If a node is selected as an anchor (marked as black), its confidence is set to be 1.

V. Simulations

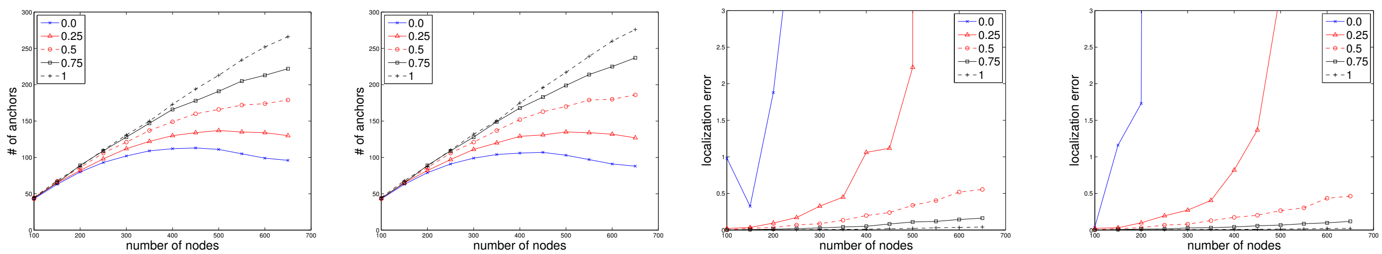
In order to evaluate the performance of proposed methods, extensive simulations are conducted on random generated 3D OSNs. In all simulations, $100 \sim 1050$ sensor nodes are randomly deployed within a $(2000m)^3$ cubic region. The sensing range of each node is set to be $300m$, i.e., if the distance between two nodes is less than or equal to $300m$, we assume that there is a distance measurement between them. The sensing range of the ship is set as $900m$, i.e., all sensor

[1] In local sweep with two-hop, five reference nodes are used. It just like the case when v_2 and v_6 are the same node.



(a) number of anchors (b) portion of anchors (c) travel distance before TSP (d) travel distance after TSP

Fig. 5. Performances of different methods for MCLP-1 and MCLP-2.



(a) # of anchors/random (b) # of anchors/trilateration (c) localization error/random (d) localization error/trilateration

Fig. 6. Performances of *Greedy-Random* and *Greedy-Trilateration* with different confidence thresholds α.

nodes within $900m$ depth are the shallow sensor nodes. A certain number of sensor nodes' coordinates are generated within the cubic region. Then using all the shallow nodes as anchors and trilateration, all sensor nodes which cannot be localized will be deleted. We only consider all sensor nodes who are localizable when all shallow sensor nodes are set as anchors. For each set of simulations, we perform the simulations for 100 times (i.e. over 100 random networks), and report the average results. We implement the three proposed methods: *Greedy-Trilateration* (Algorithms 1 and 2), *Greedy-LcoalSweep* (Algorithms 1 and 3), *Greedy-TravelDist* (Algorithms 4 and 2). In addition, we also test a random method *Greedy-Random*. In *Greedy-Random*, trilateration is recursively used to localize as many nodes as possible, and when there is no more nodes can be localized, a random node is picked to be the next anchor. This procedure is repeated until everyone are localized. In the following sections, we will test these four methods in different experiments.

A. Experiments for MCLP-1

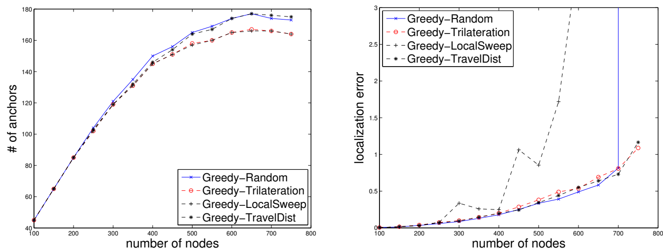
In MCLP-1, our objective is to minimize the number of anchors. Fig.5(a) shows the total number of anchors selected by each algorithm. It is obvious that the less anchor nodes selected the better. Among all methods, the random method needs the most number of anchors. *Greedy-LocalSweep* yields the best performance, since it can localize more nodes at each step. All the greedy algorithms will converge after the sensor nodes become denser. Overall, the total number of anchors first increases with the node number and then drops down sharply around 400. Fig.5(b) shows the percentage of anchors to the total number of shallow sensor nodes. Clearly, the percentage of anchors drops while the number of sensors increases. When the sensor network is sparse, most of shallow sensors have to be anchors. When the sensor network is very dense, only very small percentage of nodes needs to be anchors. This shows that our proposed algorithms can achieve better performances for large-scale 3D sensor networks.

B. Experiments for MCLP-2

In MCLP-2, we aim to minimize the travelling distance of the ship due to high cost of travelling in the sea. Fig.5(c) shows the travelling distance of each algorithm based on the visiting sequence of the greedy algorithms (i.e. the ordering of selected anchor nodes). As expected, fewer anchor nodes results in shorter travelling distance. However, the random method has much longer travelling distance than other greedy methods. Moreover, *Greedy-TravelDist* has similar performances as *Greedy-Trilateration*. It means that the anchor number plays more important role in the randomly deployed network. This maybe due to the similar distance among sensor nodes in a uniformly random deployment. As discussed in Section III-B, we can also use existing TSP algorithm to optimize the travelling distance of each algorithm. We feed the positions of all selected anchors in a genetic TSP algorithm, and use the ouput path as the course of the ship. Fig.5(d) shows the results. Roughly the total travelling distances are shorten to half of original ones. Moreover, the difference between the random method and other greedy methods become smaller and *Greedy-LocalSweep* still yields the best result in term of travelling distance.

C. Experiments for MCLP-1 with Errors

Last, we consider the MCLP problem under measurement errors. Similar as [13], we introduce a scaled zero mean Gaussian noise $\beta \times N(0, \sigma^2)$ to the distance measurements. In our simulations, we let $\sigma = 1$ and 10 as the default value of



(a) number of anchors (b) localization error

Fig. 7. Performances of different methods with $\alpha = 0.5$.

the scale factor β. All proposed algorithms now must consider the nodes' confidences during the localization process. As explained in Section IV, a node is localized only when its confidence $c(v) \geq \alpha$ where α is the confidence threshold. The location of the localized node is calculated using the least square estimation, since it cannot be absolutely solved due to measurement errors. In the following experiments, we also measure the location errors of each algorithm in addition to the number of selected anchors. Here, we use the ratio between the error of estimated distance and the actual distance based on accurate positions to represent the location error.

1) The impact of confidence threshold: We first change the confidence threshold α from 0 to 1.0. Note that when $\alpha = 0$, all methods do not consider the node confidence at all during the localization process. Due to space limit, we only show the results for *Greedy-Random* and *Greedy-Trilateration* in Fig. 6. As expected, more anchors are needed when confidence threshold increases because higher threshold means stricter selection rules for anchor nodes. Due to stricter selection rules of anchor nodes with higher confidence thresholds, the localization accuracy increases with confidence threshold as shown in Fig. 6(c) and (d). Note that the selection of confidence threshold may limit the localization capacity. For instance, in Fig. 6(d) if α is set to be 0.25, the maximum size of localized sensor set is around 400. For a network with more than 400 sensors, the localization error increases dramatically and the localization results are not reliable. However if the value of α is large enough (such as 0.75), the portion of localizable nodes can increase with little error increases.

2) Performances of different methods: We then study the performances of different proposed greedy algorithms. In this set of simulations, we set $\alpha = 0.5$. In this case, the anchor selection strategies will play a critical role in minimizing total number of anchor nodes. In Fig. 7(a), *Greedy-LocalSweep* and *Greedy-Trilateration* can achieve the better performances. Moreover, similar as experiments for MCLP-1, initially when sensor network is sparse, most of the shallow nodes have to be anchors. Gradually, as sensor network become denser, only small portion of shallow nodes need to be anchors. However localization errors induced by different methods are similar. This confirms that the confidence threshold plays more critical role in controlling localization errors as shown in Fig. 7(b).

3) The impact of measurement errors: At last, we investigate the influences of measurement errors in our localization problem. Here we fix the total number of nodes as 300 and $\alpha = 0.5$. Different methods are compared for different levels of measurement errors (generated by various values of β) as shown in Fig.8. Generally, larger value of β leads to larger measurement errors. As expected, average localization errors increase steadily with larger measurement errors. However the increase rate changes dramatically when $\beta = 25$. The trend will increase and become out of control if the measurement errors keep increasing after certain threshold value.

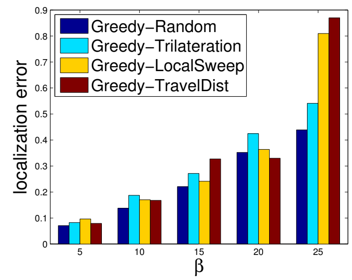


Fig. 8. Performances under different levels of errors (various values of β).

VI. Conclusion

In this paper, we extend the minimum cost localization problem (MCLP) to 3D OSNs in order to find the optimal anchor set to localize all sensor nodes in the network. Two versions of MCLP are introduced: one tries to minimize the number of anchors, while the other focuses on optimizing the travelling distance of a ship to visit all selected anchors. Both problems are computationally challenging. We propose three different greedy algorithms to find the anchor set for a given 3D OSN. We also consider how to handle measurement errors and flip ambiguity by adopting a confidence-based approach. Extensive simulations are conducted and demonstrate the efficiency of these algorithms. We leave the problem to handle the node mobility as our future work.

References

[1] M. Erol-Kantarci, H. Mouftah, and S. Oktug, “A survey of architectures and localization techniques for underwater acoustic sensor networks,” *IEEE Communications Surveys Tutorials*, 13(3):487–502, 2011.

[2] Y. Wang, Y. Liu, and Z. Guo, “Three-dimensional ocean sensor networks: A survey,” *J. of Ocean University of China*, 11(4):436–450, 2012.

[3] Y. Wang and L. Li, “Localization in wireless sensor networks,” Book Chapter in “RFID and sensor networks: architectures, protocols, security and integrations” (eds. Y. Zhang et al.), Auerbach Publications, 2009.

[4] M. Erol, F. Vieira, et al., “Multi stage underwater sensor localization using mobile beacons,” in *Proc. of SENSORCOMM*, 2008.

[5] M. Erol, F. Vieira, et al., “Localization with Dive’N’Rise beacons for underwater acoustic sensor networks,” in *Proc. of ACM WuWNet*, 2007.

[6] H. P. Tan, A. F. Gabor, Z. A. Eu, et al., “A wide coverage positioning system (WPS) for underwater localization,” in *Proc. of IEEE ICC*, 2010.

[7] X. Cheng, H. Shu, et al., “Silent positioning in underwater acoustic sensor networks,” *IEEE TVT*, 57(3):1756–1766, 2008.

[8] M. Huang, S. Chen, and Y. Wang, “Minimum cost localization problem in wireless sensor networks,” *Ad Hoc Networks J.*, 9(3):387–399, 2011.

[9] A. Assis, L. Vieira, et al., “A genetic algorithm for the minimum cost localization problem in wireless sensor networks,” in *IEEE CEC*, 2013.

[10] Z. Yang, Y. Liu, and X.-Y. Li, “Beyond trilateration: On the localizability of wireless ad-hoc networks,” in *Proc. of IEEE INFOCOM*, 2009.

[11] D. K. Goldenberg, P. Bihler, et al., “Localization in sparse networks using sweeps,” in *Proc. of ACM MobiCom*, 2006.

[12] Z. Yang and Y. Liu, “Quality of trilateration: Confidence-based iterative localization,” *IEEE TPDS*, 21(5): 631–640, 2010.

[13] X. Wang, Y. Liu, et al., “OFA: An optimistic approach to conquer flip ambiguity in network localization,” *Comp. Net.*, 57(6):1529–1544, 2013.