An Efficient Algorithm for Constructing Underwater Sensor Barrier

Weiqiang Shen[1], Chuanlin Zhang[1(✉)], Min-Rong Chen[2], Jinglun Shi[3], and Guo-Qiang Zeng[4]

[1] College of Information Science and Technology, Jinan University, Guangzhou 510632, China
weiqshen@gmail.com, tclzhang@jnu.edu.cn
[2] School of Computer, South China Normal University, Guangzhou 510631, China
mrongchen@126.com
[3] School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China
shijl@scut.edu.cn
[4] National-Local Joint Engineering Laboratory of Digitalize Electrical Design Technology, Wenzhou University, Wenzhou 325035, China
zeng.guoqiang5@gmail.com

Abstract. Most existing works on barrier coverage assume that sensors are deployed in a two-dimensional (2D) long thin belt region, where a barrier is a chain of sensors from one end of the region to the other end with overlapping sensing zones of adjacent sensors. However, 2D sensor barrier construction mechanism cannot be directly applied to three-dimensional (3D) sensor barrier construction problem, such as underwater sensor barrier construction, where sensors are finally distributed over a 3D space. In this paper, we investigate how to efficiently construct an underwater sensor barrier with minimum mobile sensors while reducing energy consumption. We first determine the minimum number of sensors needed for an underwater sensor barrier construction. Furthermore, we analyse the relationship between the initial locations of sensors and the optimal location of the underwater sensor barrier, based on which we derive the optimal final locations for all sensors. Finally, we propose an efficient algorithm to move sensors from their initial locations to final locations. Extensive simulations show that, compared with HungarianK approach, the proposed algorithm costs shorter running time and similar maximum movement distance of any one sensor.

Keywords: Underwater sensor barrier · Wireless sensor network Deployment algorithm

1 Introduction

Wireless sensor networks (WSNs) have been widely used in many real life applications, such as battlefield surveillance, environmental monitoring and industrial

B. Li et al. (Eds.): ChinaCom 2017, LNICST 237, pp. 153–164, 2018.
https://doi.org/10.1007/978-3-319-78139-6_16

diagnostics [1]. As an critical issue in WSNs, barrier coverage is garnering more and more attention in recent years [2–5]. Different from area coverage, which aims at gathering the information occurring within the region of interest (ROI), barrier coverage concerns with constructing a barrier for intrusion detection, and has been widely employed in practical security applications. For example, international border surveillance and critical infrastructure protection.

Most existing works on barrier coverage assume that sensors are deployed in a 2D long thin belt region, where a barrier is a chain of sensors from one end of the region to the other end with overlapping sensing zones of adjacent sensors. Fan et al. [6] studied the coverage of a line interval with a set of wireless sensors with adjustable coverage ranges. Liu et al. [7] studied the strong barrier coverage of a randomly-deployed sensor network on a long irregular strip region. Wang et al. [2,8] explored the effects of location errors on barrier coverage. Dobrev et al. [9] studied three optimization problems related to the movement of sensors to achieve weak barrier coverage. He et al. [10] presented a condition under which line-based deployment is suboptimal, and proposed a new deployment approach named curve-based deployment. Ban et al. [11] considered k-barrier coverage problem in 2D wireless sensor networks. Dewitt and Shi [12] incorporated energy harvesting into the 2D barrier coverage problem. Based on the 2D assumption, all sensors composing the barrier finally reside on a 2D plane. This assumption may be reasonable in a terrestrial wireless sensor network where the height of the network is usually negligible as compared to its length and width. However, 2D sensor barrier construction mechanism cannot be directly applied to 3D sensor barrier construction problem, such as underwater barrier construction, where sensors are finally distributed over a 3D space.

In the real life, a wide range of waterside critical infrastructures require the protection of underwater barrier, such as naval base, nuclear power plant and docks. How to protect them from illegal intrusion is an essential problem. A popular defense mechanism from intruders is deploying physical net, which is integrated with sensors as a barrier, along the surrounding waters of these infrastructures. For example, Marinet [13], a physical barrier, whose objective is to prevent swimmer, diver, frogman, floating explosive packages and other water based intruders from illegal intrusion. Despite the physical net barrier provides variety of functions satisfying common application to thwart illegal intruders, there are still some problems which it may not overcome so far. For example, the deployment of physical net barrier usually involves with artificial participation, this may not be a great efficient manner. Especially, when facing a vast and deep underwater space, it will cost a lot of resources whereas the construction progress of physical net is inefficient. Moreover, these physical net barriers may not satisfy the requirement of some special applications, such as submarine intrusion detection which aims to detect illegal submarine intrusion but hopes not to be discovered by the submarine that there are barriers.

To tackle aforementioned problems, constructing underwater sensor barrier (UWSB[1]) with mobile sensors with capability of intrusion detection in 3D

[1] In this paper, we only consider strong underwater sensor barrier coverage. It will be shortly referred to as UWSB in the following.

underwater space, may be an adequate alternative choice. Compared to the barrier coverage in 2D plane, a barrier coverage in 3D space is not a chain of adjacent sensors any more. Instead, a barrier in 3D space should be a set of sensors with overlapping sensing zones of adjacent sensors that covers an entire (curly) surface that cuts across the space [14].

In this work, we aim to efficiently construct an underwater sensor barrier with minimum mobile sensors while reducing energy consumption. Inspired by [15], in which a Hungarian-based approach named HungarianK was proposed to solve UWSB construction problem, we focus on constructing an UWSB via assigning each sensor to desired final location(i.e., grid point[2]). Furthermore, we recognize that the computational complexity of the Hungarian-based method is at least $O(n^3)$, this may not be a good result in term of large-scale sensor network due to the severe constraint of limited computation capability of individual sensor node. In this case, we are looking forward to proposing another approach to reduce the computational complexity while minimizing the maximum movement distance of any one sensor so as to balance energy consumption of each sensor node. Considering the difference between 2D sensor barrier and 3D underwater sensor barrier, we first determine the minimum number of mobile sensors needed for an UWSB. Furthermore, we analyse the relationship between the initial locations of sensors and the optimal location of underwater sensor barrier, based on which we derive the optimal final location for each sensor. Finally, we propose an efficient algorithm to move sensors from their initial locations to final locations. Extensive simulations show that, compared with HungarianK approach proposed in [15], the proposed algorithm costs shorter running time and similar maximum movement distance of any one sensor.

The rest of the paper is organized as follows. In Sect. 2, we explain the network model, and define some important concepts. Next, in Sect. 3, we show how to efficiently construct an underwater sensor barrier with minimum mobile sensors while reducing energy consumption. Section 4 evaluates the performance of the proposed algorithm through extensive simulations, and finally, Sect. 5 concludes the paper.

2 Model Statement

We consider an underwater wireless sensor network consisting of sensors deployed in a large-scale 3D cuboid of size $l \times w \times h$, where l, w, and h denote the length, the width, and the height of the cuboid, respectively. Without loss of generality, we assume that the illegal objects move along the direction of cuboid length, as shown in Fig. 1, O_1 and O_2 denote object1 and object2, respectively. The following assumptions are made in this work.

- An object (or intruder) may cross the underwater sensor barrier via an crossing path starting at the left face and ending at the right face of the cuboid.

[2] In this paper, in order to make presentation clearer, we use final location instead of grid point.

- Each sensor has the following abilities: localize its own position, to move in all 3-dimensions underwater, communicate with other sensors, and detect intruders. For simplicity, we assume an ideal 0/1 sphere sensing model that an object within (outside) a sensor's sensing sphere is detected by the sensor with probability one (zero).
- In the initial configuration, the locations of all sensors are uniformly and independently distributed in the cuboid. Such a random initial deployment is desirable in scenarios where prior knowledge of the region of interest is not available, and may be the result of certain deployment strategies [15].

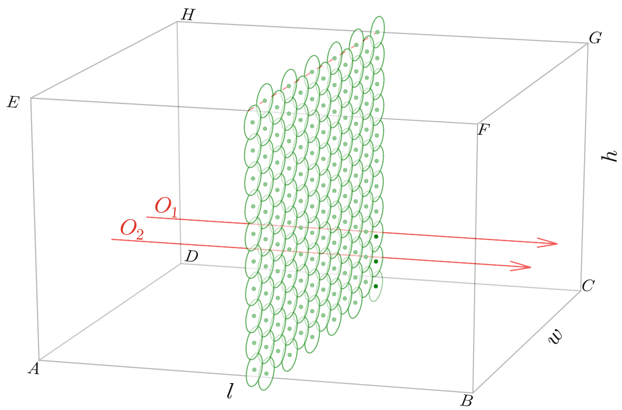

Fig. 1. Illustration of an underwater sensor barrier

In the following, we introduce some important definitions used in this work.

Definition 1 (Underwater sensor barrier [14]). *A set of sensors, with overlapping sensing zones of adjacent sensors, that cover an entire (curly) surface which cuts across the underwater space.*

Definition 2 (Crossing path). *A crossing path is a continuous moving trajectory with the start point at one face of the cuboid and the end point at the opposite face.*

Definition 3 (Initial location). *In the initial configuration, all sensors are uniformly and independently distributed in underwater space, the sensor's location at this moment is referred to as initial location.*

Definition 4 (Final location). *The final location is the movement destination of sensor. An UWSB is constructed after all sensors arrive at their final locations.*

3 Constructing Underwater Sensor Barrier

According to aforementioned assumptions, initially, all sensors are uniformly and independently distributed in a cuboid. Our goal is to efficiently construct an underwater sensor barrier with minimum mobile sensors by moving them from their initial locations to optimal final locations. To construct an UWSB effectively and efficiently, we first derive the minimum number of required sensors, by which we can determine whether there are enough sensors to construct an UWSB. Then we implement UWSB construction via following three phases. (1) Find the optimal location of the UWSB. (2) Compute the optimal final locations of all sensors. (3) Propose an efficient algorithm to assign the sensors to their final locations.

3.1 The Minimum Number of Required Sensors

In the context of our work, it is a rectangle of size $w \times h$ after projecting the UWSB to the left face of the cuboid. Thus, the minimum number of required sensors equals to the minimum number of required circles with radius r to full cover a rectangle of size $w \times h$. In this case, the circle is the 2D projection of sensor's sensing sphere, and the radius r is the sensing radius of the sensor.

Actually, in term of the minimum number of circles required for the complete area coverage problem, Kershner [16] had proved that the regular triangular tessellation is the optimal tessellation which results in a set of regular hexagons full cover a 2D plane without any overlap. These sensors finally locate at the center of each regular hexagon with circumradius r, which is the sensing radius of the sensor, as shown in Fig. 2. To obtain the minimum number of required sensors, Theorem 1 is given as follows:

Theorem 1. *The minimum number of required regular hexagons with circumradius r to full cover a rectangle of size $w \times h$ is:*

$$\begin{aligned} f_s(w,h,r) = \lceil \frac{h}{r \times \sqrt{3}} \rceil \times \lceil \frac{\lceil \frac{2\times(w-r)}{3\times r} \rceil + 1}{2} \rceil \\ + (\lceil \frac{h - \frac{r\times\sqrt{3}}{2}}{r \times \sqrt{3}} \rceil + 1) \times \lfloor \frac{\lceil \frac{2\times(w-r)}{3\times r} \rceil + 1}{2} \rfloor. \end{aligned} \tag{1}$$

Proof. Given a rectangle of size $l \times w$, in the length direction, we divide the rectangle into C columns, the first column width $a = r$, 2-th∼ $(C-1)$-th column width $b = \frac{3\times r}{2}$, and the last column width $\in (0, b]$, as shown in Fig. 2. So, the number of columns is:

$$f_c(w,r) = \lceil \frac{2 \times (w-r)}{3 \times r} \rceil + 1. \tag{2}$$

In the width direction, the number of rows of odd-number columns is:

$$f_o(h,r) = \lceil \frac{h}{r \times \sqrt{3}} \rceil. \tag{3}$$

The number of rows of even-number columns is:

$$f_e(h,r) = \lceil \frac{h - \frac{r\times\sqrt{3}}{2}}{r\times\sqrt{3}} \rceil + 1. \tag{4}$$

Combining Eqs. (2), (3), and (4), we have the minimum number of regular hexagons:

$$\begin{aligned} f_s(w,h,r) &= f_o(d,r) \times \lceil \frac{f_c(w,r)}{2} \rceil + f_e(h,r) \times \lfloor \frac{f_c(w,r)}{2} \rfloor \\ &= \lceil \frac{h}{r\times\sqrt{3}} \rceil \times \lceil \frac{\lceil \frac{2\times(w-r)}{3\times r} \rceil + 1}{2} \rceil \\ &\quad + (\lceil \frac{h - \frac{r\times\sqrt{3}}{2}}{r\times\sqrt{3}} \rceil + 1) \times \lfloor \frac{\lceil \frac{2\times(w-r)}{3\times r} \rceil + 1}{2} \rfloor. \end{aligned} \tag{5}$$

In this work, the projection of the UWSB is a rectangle of size $w \times h$, so the minimum number of required sensors is $f_s(w,h,r)$. Thus, if the number of deployed sensor $N >= f_s(w,h,r)$, then at least one UWSB can be constructed. Otherwise, we cannot achieve our goal.

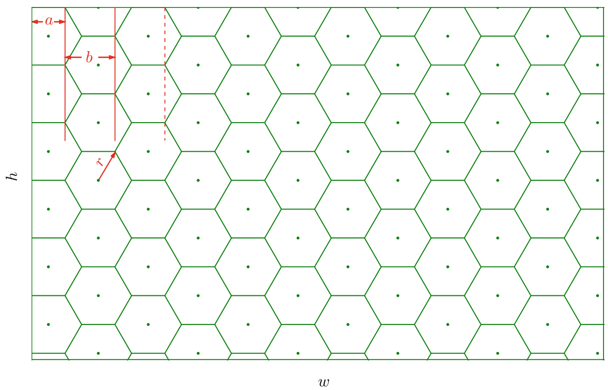

Fig. 2. Fully cover a rectangle of size $w \times h$ with minimum regular hexagons.

3.2 Find the Optimal Location of UWSB

We note that the optimal location of UWSB is similar to the optimal barrier location in [14], where an algorithm was devised to find the optimal sensor barrier location.

Supposing as in Fig. 3 that the plane X is parallel to the coast line to be protected, a set of mobile sensors dropped from an aircraft were floating on water surface. The sensors must move to some plane X, such that X minimizes the energy expended by any one sensor. In this case, we are assuming that the sensors will only move in one direction to approach the X location. It is straightforward to show that X must be somewhere inside the set x-coordinates of the sensors, and as any two sensors move to meet one another, the total distance they travel is the distance between them. Thus, the sensors on either edge move the farthest, and half the distance between them minimizes the maximum distance any one sensor moves. So, to minimize the maximal distance traveled by any sensor, we can calculate the optimal X location as follows [14]:

$$X = (max\ x_i + min\ x_i)/2$$

For example, in Fig. 3, $max\ x_i = 8, min\ x_i = 2$. Thus, all the sensors should move to the line $X = (8 + 2)/2 = 5$ to minimize the movement distance of any one sensor. That means line $X = 5$ is the optimal location of the UWSB.

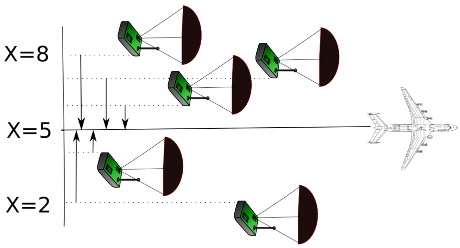

Fig. 3. The first phase: an air drop of sensors along a straight line resulting in scattered placement along that line. The figure is taken from [14].

3.3 Compute the Optimal Final Locations of All Sensors

Since an UWSB is actually a flat surface parallel to the left face of cuboid, the optimal final locations of all sensors are the central points of regular hexagon, as shown in Fig. 2. Thus, for each central point of regular hexagon, the x-coordinate equals to that of optimal location of UWSB, we can derive y, z-coordinates of each central points as follows:

We first get the y-coordinate of each column via Eq. (6),

$$f_y(w, r, i) = \begin{cases} \frac{r}{2}, & \text{i} = 0 \\ \frac{r}{2} + \frac{i \times r}{2}, & 1 \le i < N_{sp} - 1 \\ \frac{r}{2} + \frac{i \times r}{2}, & i = N_{sp} - 1 \ \& \ \frac{r}{2} + \frac{(N_{sp}-1) \times r}{2} < w \\ w, & i = N_{sp} - 1 \ \& \ \frac{r}{2} + \frac{(N_{sp}-1) \times r}{2} \ge w. \end{cases} \tag{6}$$

then we get the z-coordinate of each point column by column. For odd-number columns, h is the cuboid height, r is the sensing radius, R_j denotes the j-th row, and $0 \le j < \lceil \frac{h}{r \times \sqrt{3}} \rceil$, we have

$$f_z(w, r, R_j) = h - \frac{r \times \sqrt{3}}{2} - R_j \times r \times \sqrt{3}. \tag{7}$$

For even-number columns, h is the cuboid height, r is sensor's sensing radius, R_j denote the j-th row, and $0 \le j < \lceil \frac{h - \frac{r \times \sqrt{3}}{2}}{r \times \sqrt{3}} \rceil + 1$, we have

$$f_z(w, r, R_j) = h - R_j \times r \times \sqrt{3}. \tag{8}$$

Combining Eqs. (7) and (8), we have

$$f_z(w, r, R_j) = \begin{cases} h - \frac{r \times \sqrt{3}}{2} - R_j \times r \times \sqrt{3}, & \text{odd-number columns} \\ h - R_j \times r \times \sqrt{3}, & \text{even-number columns.} \end{cases} \tag{9}$$

Therefore, by combining the optimal location x of UWSB, we get the coordinates $(x, f_z(w, r, R_j), f_z(w, r, R_j))$ of optimal final locations of all sensors.

3.4 Movement Algorithm

After determining the optimal final locations of all sensors, an UWSB can be constructed by moving sensors from their initial locations to the final locations. Actually, this movement process is related to the Assignment Problem, which aim to create a one-to-one matching between sensors and final locations. A classic solution to the Assignment Problem is known as the Hungarian Method which can be computed in $O(n^3)$, where n is the number of sensors or final locations.

In [15], the authors proposed a Hungarian-based approach named HungarianK, which can be computed in $O(n^4)$, to solve the sensor assignment problem. However, it may not be a good choice in term of large-scale sensor network due to the severe constraints of limited computation capability of individual sensor node. In this case, we are looking forward to proposing another method to reduce the computational complexity while minimizing the maximum movement distance of any one sensor.

Since all sensors are uniformly and independently distributed in the cuboid, to make control of the maximum movement distance of any one sensor while reduce the computation time, we propose a **height-based match algorithm (HBMA)** to solve our problem. We assume that the cuboid locates at a 3D coordinate system where integer coordinates (x, y, z) represent the location information of a point in underwater space, N_i denotes the number of sensors in the i-th column, all sensors are sorted by x-coordinate in ascending order firstly, this sorted sensor list is denoted as L_{sort}. For each column of the UWSB, N_i sensors are popped out from L_{sort} and assigned to the i-th column, and each sensor will assign to its final location according its height (i.e., z-coordinate). The detail of HBMA is shown in Algorithm 1.

From the pseudo-code of Algorithm 1, we learn that the proposed algorithm consists two main loops. In the outer loop (line 05–line 20), UWSB is constructed column by column, and the outer loop terminates if the last column (i.e., the N_{column}-th column) of UWSB is constructed. Thus, the outer loop runs N_{column} times. In the inner loop, there are two sub-loop (i.e., line 12–line 14 and line 16–line 19) both terminate in $O(n)$. However, line 15 concerns with Quicksort

Algorithm 1. $HBMA(L_{init})$

Input:
The initial locations L_{init} of all sensors.

Output:
The matched list L_{match}, such as $\{((x_0, y_0, z_0), (x_0^{'}, y_0^{'}, z_0^{'})), ((x_1, y_1, z_1), (x_1^{'}, y_1^{'}, z_1^{'})), \ldots\}$. $((x_n, y_n, z_n), (x_n^{'}, y_n^{'}, z_n^{'}))$ means that sensor at the location of (x_n, y_n, z_n) should move to the location of $(x_n^{'}, y_n^{'}, z_n^{'})$.

$L_{sort} \leftarrow Quicksort(L_{init})$;
Compute the optimal final locations of all sensors, return the optimal final location list L_{fl}
$N_{column} \leftarrow f_c(w, r)$;
$seq \leftarrow 0$;
for $col = 0 \rightarrow N_{column} - 1$ **do**
    if $(col + 1)\%2 == 1$ **then**
        $N_{row} \leftarrow f_o(h, r)$;
    else
        $N_{row} \leftarrow f_e(h, r)$;
    end if
    $L_{temp} \leftarrow null$
    for $row = 0 \rightarrow N_{row} - 1$ **do**
        $L_{temp} \leftarrow L_{sort}.pop()$
    end for
    $L_{temps} \leftarrow Quicksort(L_{temp})$;
    for $row = 0 \rightarrow N_{row} - 1$ **do**
        $L_{match} \leftarrow (L_{temps}[row], L_{fl}[seq])$;
        $seq \leftarrow seq + 1$;
    end for
end for

algorithm whose computational complexity is $O(nlog(n))$ in average, and in the worst case, the computation complexity of Quicksort is $O(n^2)$. Therefore, the proposed algorithm can be computed in $O(n^2log(n))$ in average. Even in the worst case, the computation complexity of HBMA is just $O(n^3)$. Compared with the HungarianK approach proposed in [15], whose computational complexity is $O(n^4)$, the proposed algorithm has lower computational complexity.

4 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm through extensive simulations by using Python program language. We setup the simulation environment as follows:

1. The underwater space is modeled as a cuboid with length $l = 4000$ m, width $w = 3600$ m, and height $h = 2500$ m, respectively.
2. We use the minimum number of required sensors to construct an UWSB in the simulations. Sensors are uniformly and independently distributed in the cuboid.
3. We vary the sensing radius r_s of the sensor from 400 m to 200 m. From the Eq. (5), we learn that the shorter the sensing radius, the more the number

of required sensors. Thus, as the variation of sensing radius r_s, the minimum number of required sensors varies from 31 to 104.
4. All experiments are repeated by 100 runs.

The following performance metrics are used to evaluate the performance of the proposed algorithm.

– Running time: The lower Running time, the lower requirement of computation power and the less energy to be consumed for the sensor.
– Maximum movement distance of any one sensor: For each sensor, the movement distance indicates the straight-line distance between the initial location and the final location. Generally, the movement distance of a sensor is proportional to its energy consumption, and the longer the movement distance, the larger the energy consumption. Thus, in order to balance energy consumption, we hope to minimize the maximum movement distance of any one sensor.

In the simulations, we evaluate the performance of the proposed algorithm by constructing an UWSB with the minimum mobile sensors. The sensing radius r_s varies from 400 m to 200 m, accordingly, the minimum number of required sensors increases from 31 to 104. As shown in Fig. 4, the running time increases with the increasement of the number of sensors, and the proposed algorithm HBMA costs shorter running time than the HungarianK proposed in [15]. Figure 5 depicts the maximum movement distance of any one sensor increases with the increasement of the number of sensors, the proposed algorithm HBMA is comparable to the HungarianK in term of the maximum movement distance of any one sensor. Overall, compared with the HungarianK, the proposed algorithm costs shorter running time and similar maximum movement distance of any one sensor.

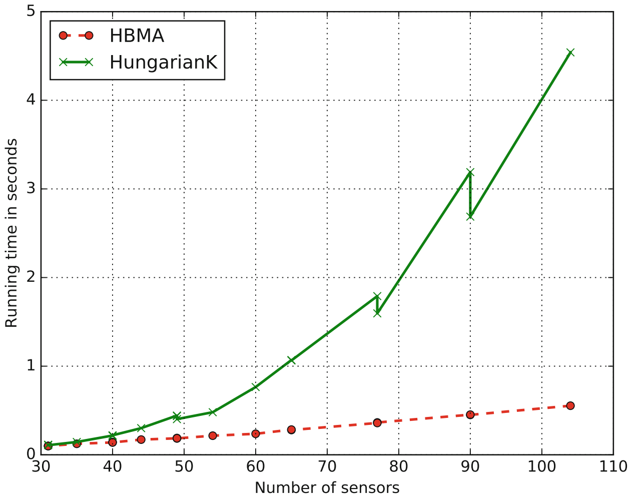

Fig. 4. Comparison of running time between the HungarianK and the proposed algorithm HBMA.

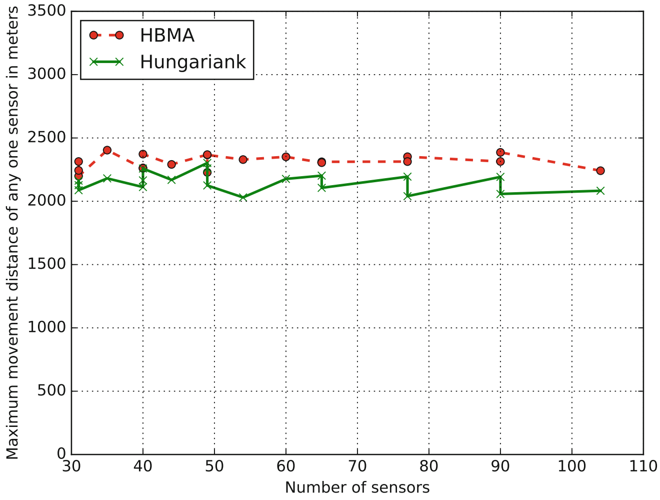

Fig. 5. Comparison of maximum movement distance of any one sensor between the HungarianK and the proposed algorithm HBMA.

5 Conclusion

In this work, to efficiently construct an UWSB with minimum mobile sensors while reducing energy consumption, we first analyse the relationship between the optimal location of UWSB and the initial positions of sensors, and determine the minimum number of mobile sensors needed for constructing an UWSB. Moreover, we derive the optimal final locations of all sensors, based on which we propose an efficient algorithm to move sensors from their initial locations to final locations. Extensive simulations show that, compared to the HungarianK approach, the proposed algorithm costs shorter running time and similar maximum movement distance of any one sensor.

Acknowledgements. The authors would like to thank the referees whose insightful comments have helped improve the presentation of this paper significantly. This work was supported in part by Major Omnibus Reform Project (Information and Computing Science) of China under Grant No. 82616611, National Science Foundation of China under Grant Nos. 61373158 and 61671213, Guangzhou Key Lab of Body Data Science under Grant No. 201605030011, and Zhejiang Provincial Natural Science Foundation of China under Grant Nos. LY16F030011 and LZ16E050002.

References

1. Wang, B.: Coverage problems in sensor networks: a survey. ACM Comput. Surv. (CSUR) **43**, 1–53 (2011)
2. Wang, Z., Chen, H., Cao, Q., Qi, H., Wang, Z., Wang, Q.: Achieving location error tolerant barrier coverage for wireless sensor networks. Comput. Netw. **112**, 314–328 (2017)

3. Silvestri, S., Goss, K.: MobiBar: an autonomous deployment algorithm for barrier coverage with mobile sensors. Ad Hoc Netw. **54**, 111–129 (2017)
4. Paul, B., Béla, B., Amites, S.: Barrier coverage. Random Struct. Algorithms **49**, 429–478 (2016)
5. Wu, F., Gui, Y., Wang, Z., Gao, X., Chen, G.: A survey on barrier coverage with sensors. Front. Comput. Sci. **10**, 968–984 (2016)
6. Fan, H., Li, M., Sun, X., Wan, P.J., Zhao, Y.: Barrier coverage by sensors with adjustable ranges. ACM Trans. Sens. Netw. (TOSN) **11**, 1–20 (2014)
7. Liu, B., Dousse, O., Wang, J., Saipulla, A.: Strong barrier coverage of wireless sensor networks. In: Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 411–420 (2008)
8. Wang, Z., Chen, H., Cao, Q., Qi, H., Wang, Z.: Fault tolerant barrier coverage for wireless sensor networks. In: INFOCOM, 2014 Proceedings IEEE, pp. 1869–1877 (2014)
9. Dobrev, S., et al.: Weak coverage of a rectangular barrier. In: Fotakis, D., Pagourtzis, A., Paschos, V.T. (eds.) CIAC 2017. LNCS, vol. 10236, pp. 196–208. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57586-5_17
10. He, S., Gong, X., Zhang, J., Chen, J., Sun, Y.: Barrier coverage in wireless sensor networks: from lined-based to curve-based deployment. In: 2013 Proceedings of the IEEE, INFOCOM, pp. 470–474 (2013)
11. Ban, D., Yang, W., Jiang, J., Wen, J., Dou, W.: Energy-efficient algorithms for k-barrier coverage in mobile sensor networks. Int. J. Comput. Commun. Control **5**, 616–624 (2016)
12. DeWitt, J., Shi, H.: Barrier coverage in energy harvesting sensor networks. Ad Hoc Netw. **56**, 72–83 (2017)
13. Marinet: Underwater Port Security System - Intrusion Detection Barrier. http://www.rbtec.com/products/Maritime-Underwater-Security/marinetunderwatersecuritynet
14. Barr, S.J., Wang, J., Liu, B.: An efficient method for constructing underwater sensor barriers. JCM **6**, 370–383 (2011)
15. Barr, S., Liu, B., Wang, J.: Barrier coverage for underwater sensor networks. In: Military Communications Conference MILCOM, pp. 429–478 (2008)
16. Kershner, R.: The number of circles covering a set. Am. J. Math. **61**, 665–671 (1939)