On Improving the Efficiency of Breach-Free Scheduling of Reinforced Sensor Barriers

Jorge A. Cobb

Department of Computer Science
The University of Texas at Dallas
Richardson, Texas 75080
Email: cobb@utdallas.edu

***Abstract*—A wireless sensor network consists of an area of interest in which a group of nodes have been randomly located. Furthermore, each node is able to sense activity in the area surrounding it, and thus, the network can be used for intrusion detection. Due to the limited lifetime of sensors, the network lifetime is maximized by organizing sensors into barriers, where each barrier is a subset of sensors that prevents the intruder from crossing the area. However, if not carefully chosen, sensor barriers can have a *breach*, i.e., there is a location between two barriers that are scheduled consecutively that allows the intruder to cross the area undetected. Given a set of barriers, deciding if there is a breach-free schedule of these barriers is intractable, which has led to the development of several heuristics. In earlier work, we introduced *reinforced* sensor barriers, which prevent the crossing of the area of interest in more than one direction, and presented heuristics for obtaining the maximum number of breach-free reinforced barriers. However, the computational complexity of this heuristic is high. In this paper, we present two additional heuristics with lower computational complexity, and compare their performance with our initial heuristic.**

***Keywords*–Sensor networks; Barrier coverage; Security breaches.**

I. INTRODUCTION

A wireless sensor network consists of a collection of computing nodes that may communicate wirelessly and are able to sense the area around them. The sensor nodes are typically spread randomly over an area of interest, such as a battlefield. Furthermore, since they are wireless, we assume they are battery operated, and thus have a limited lifetime [1].

An obvious possible function of a sensor network is intrusion detection, in which the objective of the sensor nodes is to detect an intruder that is attempting to cross the area of interest. Usually, sensors have a sensing range that is smaller than the area of interest. This requires multiple sensing nodes to be operating at the same time.

To maximize the lifetime of the network, the sensors can be divided into disjoint groups. These are then organized in a sleep-wakeup schedule, where one group is active while the remaining groups are in sleep mode. Once the battery is exhausted in the active group, another group is activated. This continues until the battery is exhausted in all nodes.

The amount of coverage of the area of interest is typically divided into two categories: full coverage and partial coverage. In full coverage, every group of sensor nodes must have a collective sensing coverage of the entire area of interest [2]–[5]. In partial coverage, each sensor group may only cover part of the area of interest. In this case, the objective is not to detect the presence of an intruder, but instead the movement of the intruder as it crosses the area [6]–[8]. Our focus is in partial coverage.

A popular form of partial coverage is *barrier coverage*. In this case, each group of sensors forms a continuous barrier from side-to-side across the area such that intruders are prevented from crossing undetected. Sensor barriers have been studied extensively due to their many applications [9]–[16].

In Fig. 1(a), we show four sensor barriers, B_1 through B_4, that prevent an intruder from crossing the rectangular area of interest from top to bottom. Only one of the barriers will be active at any moment, and thus the network lifetime is about four-times that of a sensor node. Typically, a schedule is formed in which the barriers are scheduled consecutively, and a barrier is activated only after the previous barrier is totally devoid of power.

The problem of dividing the sensors into the maximum number of disjoint barriers has been solved in polynomial time [11]. The approach is based on transforming the sensor connectivity graph into a maximum-flow problem.

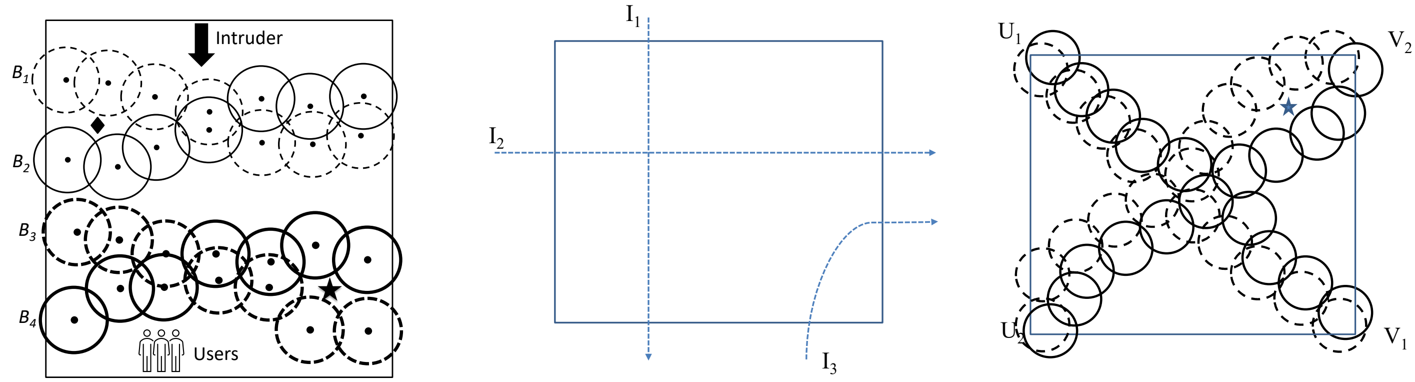
However, care must be taken when choosing the order in which barriers are activated, because some orderings may expose a vulnerability known as a *barrier breach* [17] [18]. For some barrier sleep-wakeup schedules, it is possible for an intruder to cross the area of interest after activating one barrier and deactivating the previous one. The following example illustrates this.

Consider again the four barriers in Fig. 1(a), and consider scheduling the barriers in order of their number, i.e., B_1, B_2, B_3, and finally B_4. Consider the point highlighted by the dark star. An intruder can remain at the top of the area while barriers B_1 and B_2 are active. Once barrier B_3 becomes active (and the former barriers inactive), the intruder moves to the dark star position. Note that the sensors of B_4 are not yet active. When B_4 is activated and B_3 deactivated, the intruder can cross the area and reach the users undetected. A similar situation occurs if B_4 is scheduled first before B_3.

It is known that, *given* a set of disjoint barriers, finding the longest breach-free schedule of the given barriers is NP-complete [19]. This has prompted several heuristics to be developed [17]–[21], including a probabilistic algorithm in [19]. The complexity of finding the longest breach-free schedule of barriers from a random placement of nodes remains an open problem.

In earlier work, we introduced a stronger form of a barrier, called *reinforced barrier* [22]. To illustrate this barrier, consider Fig. 1(b), in which the area of interest is a rectangle. The

 ISBN: 978-1-61208-998-0



(a) Sensor Barriers (b) Multi-Directional Crossings (c) Breach in Reinforced Barriers

Figure 1. Sensor Barriers and Breaches.

objective is to prevent an intruder from crossing the area by entering from any of its sides and exiting via another side. For example, intrusion I_1 is a vertical intrusion, intrusion I_2 is an horizontal intrusion, while I_3 is a corner intrusion (by turning from vertical to horizontal).

To prevent these intrusions, consider Fig. 1(c), where two reinforced barriers are depicted: the first barrier is depicted with solid lines and the second with dashed lines. Note that each reinforced barrier consists of two diagonal barriers, one barrier from corner U_1 to corner V_1, and another barrier from corner U_2 to corner V_2. Note also that the two diagonal barriers need not be disjoint. By combining two diagonal barriers, none of the above intrusions are possible.

Note that barrier breaches are still possible, as shown in Fig. 1(c). If we schedule the solid line barrier first, then an intruder can arrive to the location marked by the star. Once the next barrier is activated and the first barrier deactivated, the intruder can cross the area and exit via the right side.

Although heuristics to obtain the longest schedule of reinforced barriers were presented earlier in [22], a heuristic for a breach-free schedule of reinforced barriers was not addressed until our recent work [23]. However, the computational complexity of this heuristic is high. In this paper, we present two additional heuristics with lower computational complexity, and compare their performance with our initial heuristic via simulations.

The rest of this paper is organized as follows. Section II presents background and definitions. In Section III, we present our heuristics. Simulation results are presented in Section IV. The conclusion is given in Section V.

II. Background

In this section, we present definitions and discuss earlier methods, before we present our heuristic in Section III.

A. Barrier Schedules

We consider a set S containing n sensor nodes that have been deployed randomly over a rectangular area. A *barrier* consists of a subset B of S arranged in a sequence, $s_1, s_2, \ldots, s_{|B|}$, such that the sensor ranges of s_i and s_{i+1}, $1 \leq i < |B|$, overlap with each other, and furthermore, the sensing range of s_1 overlaps one of the sides of the rectangle, while the sensing range of $s_{|B|}$ overlaps the opposite side of the rectangle.

In Fig. 1(a), four barriers are shown, B_1 through B_4. These barriers are horizontal. A barrier is horizontal if the sides being overlapped are the left and right sides, and is vertical otherwise. We focus first on horizontal barriers; reinforced barriers are presented further below.

A *barrier schedule* of length k consists of a sequence of barriers, $B_1, B_2, \ldots, B_k$, such that no two pairs of barriers in the schedule have sensors in common.

Finding the longest barrier schedule has been solved in polynomial time by Kumar et al. [11] with their algorithm known as Stint. The method builds a flow graph F consisting of all sensor nodes in S plus two fictitious nodes, u and v. Node u has an edge with all nodes overlapping the left border of the area, while v has an edge with all the nodes overlapping the right area.

Graph F is constructed in such a way that the maximum flow from u to v corresponds to the number of sensor barriers, and a path with non-zero flow corresponds to a barrier.

B. Barrier Breaches

An ordered pair (B_1, B_2) of horizontal barriers forms a *breach* if there is a point p not covered by either barrier such that a line can be drawn from the top of the area to p without overlapping the sensing area of B_1, and furthermore, a line can be drawn from p to the bottom of the area without overlapping the sensing area of B_2. A barrier schedule $B_1, B_2, \ldots, B_k$ is *breach-free* if every pair of consecutive barriers in the sequence does not form a breach.

Consider the two points marked in Fig. 1(a): the diamond between B_1 and B_2, and the star between B_3 and B_4. Note that the pair (B_3, B_4) forms a breach since the intruder can reach the star from the top side while B_3 is active, and switching to B_4 allows the intruder to reach the users at the bottom of the area. Note also that pair (B_1, B_2) does not form a breach, but pair (B_2, B_1) does.

Given a set of sensors and their location, finding the longest breach-free barrier schedule is desirable. It is known that, given a set of *barriers*, obtaining the longest breach-free schedule of the *given* barriers is NP-complete [19]. Thus, it is likely that the general problem where only the sensors are given is also NP-complete.

Some heuristics, such as those presented in [17] [18], create their schedule of barriers by first obtaining a set of barriers from the Stint algorithm, followed by selecting a subset of these barriers that do not cross each other. Others try all possible schedules obtained from the Stint barriers, or try them at random [19].

Note that Stint is not guaranteed to provide as output a set of barriers that is breach free even though one exists [17].

In [20], we presented another approach to obtain breach-free schedules based on ceilings and floors of barriers, as follows.

A sensor barrier, as shown in Fig. 2(a), divides the area of interest into an upper region and a lower region. The *ceiling* of a barrier B consists of all points p along the border of the sensing radius of each sensor in B such that one can travel from p to any point in the upper region without crossing the sensing area of any sensor. The *floor* is defined similarly.

In Fig. 2(b), we show three barriers. For the schedule B_1, B_2, B_3, a breach-point is marked by a star. Note that the floor of B_3 crosses over the ceiling of B_2, which cases the breach. The approach to build the barriers is thus illustrated in Fig. 2(c), where the barrier construction begins at the top of one side of the area, following the ceiling obtained from the top most nodes until the opposite side is reached. The process is then repeated to obtain subsequent barriers in the schedule.

C. Reinforced Barriers

A *reinforced barrier* R is a set of sensors such that a line cannot be drawn starting from a side of the rectangle and ending at a different side without crossing the sensing area of a sensor. Note that this requires each of the corners to be covered by at least one sensor, and it also implies that there is a subset R' of R such that R acts both as a horizontal and vertical barrier (i.e., a diagonal barrier). By symmetry, R is the union of two diagonal barriers.

Similarly, an ordered pair (R_1, R_2) of reinforced barriers forms a breach if there is a point p not covered by either barrier such that a line can be drawn from some side of the area to p without overlapping R_1, and furthermore, a line can be drawn to p to a *different* side of the area without overlapping R_2. A reinforced-barrier breach is shown in Fig. 1(c).

III. Breach-Free Reinforced Barriers Heuristics

In this section, we present three heuristics for obtaining the longest breach-free schedule of reinforced barriers. We introduced the first one in [23], in which all possible combination of barriers are explored. Due to its complexity, we present below two additional heuristics. One of these is similar to our first heuristic, but uses instead a random walk to find the schedule of barriers. The other is based on a geometrical flooding similar to the one we proposed in [20] for simple horizontal barriers. All three heuristics are then compared via simulations in Section IV.

A. Diagonal Barriers Scheduling

In this sub-section, we present the heuristic that we introduced in [23]. As shown in Fig. 1(b), two diagonal barriers are needed to form a reinforced barrier. Note that this is always the case even when the diagonal barriers are not apparent, which we argued in [23].

Our approach consists in first obtaining the maximum number of disjoint diagonal barriers from U_1 to V_1, and then combining them with the maximum number of disjoint barriers from U_2 to V_2. This is illustrated in Fig. 1(c).

Let $\mathcal{D}_1$ be a maximal set of disjoint barriers from U_1 to V_1. Similarly, let $\mathcal{D}_2$ be a maximal set of disjoint barriers from U_2 to V_2. To obtain the set of barriers $\mathcal{D}_1$ and $\mathcal{D}_2$, we can take advantage of Stint by running it twice: the first time to obtain $\mathcal{D}_1$, and the second time to obtain $\mathcal{D}_2$, as shown in [23].

The union of any two barriers B_1 and B_2, where $B_1 \in \mathcal{D}_1$ and $B_2 \in \mathcal{D}_2$, forms a reinforced barrier, R. Note in particular that B_1 and B_2 do not need to be disjoint; this is because they will be activated simultaneously. We denote the set of all reinforced barriers with $\mathcal{R}$, i.e., $\mathcal{R} = D_1 \times D_2$.

Our objective is to find the maximum breach-free schedule using the reinforced barriers in $\mathcal{R}$. To accomplish this, we build a graph G whose nodes are elements of $\mathcal{R}$. A directed edge (R_1, R_2) exists in G if the pair (R_1, R_2) does not constitute a breach. In this way, obtaining the longest breach-free schedule is equivalent to the problem of finding the longest simple path in G. Finding the longest path in a directed graph is an NP-Complete problem. Thus, all possible paths have to be examined.

The above approach is similar to the one used in [19], except that the problem considered in [19] is obtaining a maximum breach-free schedule of horizontal barriers.

However, there is a significant difference between reinforced barriers and horizontal barriers: reinforced barriers are not independent of each other. If a barrier $R = B_1 \bigcup B_2$, where $B_1 \in \mathcal{D}_1$ and $B_2 \in \mathcal{D}_2$, is used in the schedule, then for any $B' \in \mathcal{D}_2$, $R' = B_1 \bigcup B'$ cannot appear in the same schedule. This is because the diagonal barrier B_1 takes part in both reinforced barriers. We refer to the pair R and R' as being *incompatible*.

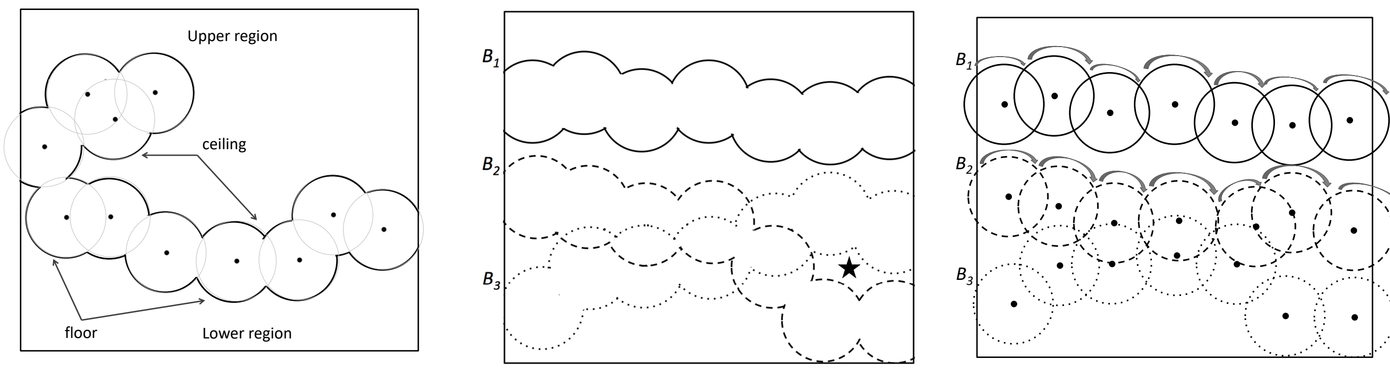
Note that, because incompatible barriers cannot appear in a schedule, then the length of the schedule is upper bounded by $\lambda = \min(|\mathcal{D}_1|, |\mathcal{D}_2|)$. This limits the search for paths of length at most λ.

B. Randomized Diagonal Barriers

Exploring all possible paths in the above heuristic is of the order of $|\mathcal{R}|^\lambda$. Given that $|\mathcal{R}| = |\mathcal{D}_1| \cdot |\mathcal{D}_2|$, this is at least λ^λ, which grows quite quickly with λ. To help mitigate this complexity, we propose a random search of graph G in order to find its longest path. We based our method in the random algorithm from [24] to find the longest path in directed graphs. A similar method is used in [19] to find the longest path for horizontal barriers.

Consider searching for a simple path of length L, $L \leq \lambda$, in G. First, randomly color the nodes of G using L colors. The existence of a path where all nodes have distinct colors, i.e., a *monochromatic* path, can be obtained using dynamic programming as follows. Let C be a subset of the L colors,

 ISBN: 978-1-61208-998-0



(a) Ceiling and Floors (b) Barrier Breach (c) Following the Ceiling

Figure 2. Barrier Breach Avoidance.

and R be a reinforced barrier, i.e., R is a node in G. Let $path(C, R)$ be a boolean function that returns true if there is at least one monochromatic path of length $|C|$ using exactly the colors in C whose last node is R. Let c be a color, $c \notin C$. Then,

$$path(C \cup \{c\}, R) = \langle \exists R', (R', R) \in G \wedge path(C, R') \rangle \wedge color(R) = c.$$

One issue remains to be addressed: incompatible barriers. It is possible that the path found for $path(C, R')$ contains diagonal barriers that are part of R. To remedy this, we change $path(C, R)$ from type boolean to type set, and it contains the set of diagonal barriers that have been used in creating the path found for $path(C, R)$. Thus, if $R = \langle B_1, B_2 \rangle$, where B_1 and B_2 are its diagonal barriers, then

$$\begin{aligned} &path(C \cup \{c\}, R) = path(C, R') \cup \{B_1, B_2\} \\ &\quad \text{for some } R' \text{ if} \\ &\quad\quad (R', R) \in G \wedge path(C, R') \neq \emptyset \wedge color(R) = c, \\ &path(C \cup \{c\}, R) = \emptyset \text{ otherwise.} \end{aligned}$$

Finally, since λ is an upper bound on the size of the breach-free schedule, the actual schedule may be smaller. Hence, we begin searching with $L = \lambda$, and decrease L if no path is found. In addition, since the search is random, it has to be run multiple times for each value of L to increase the likelihood of success. We chose to run the algorithm $2 \cdot L$ times whenever we search for a path of length L. This generated good results during our evaluations.

C. Flooding

Our final heuristic is based on the ceiling heuristic that we presented in [20], and is illustrated in Fig. 2(c). Recall that we begin at the top-most node, and follow the ceilings of the sensors until the opposite side of the area is reached. Since we are doing reinforced barriers, this is not sufficient. We therefore continue following the ceilings, all around the area, until we reach the original node. This is illustrated in Fig. 3. The first barrier consists of the dark circles (sensor areas). These nodes are then removed, and the process is repeated to obtain the next barrier, shown as gray circles.

We refer to this method as the flood method, in the sense that if we picture the sensor areas as being sandbags and the surrounded area flooded with water, then the sensor areas in contact with the water will form the initial barrier. These sensors are removed and the next sensor areas in contact with the water form the next barrier, and so on.

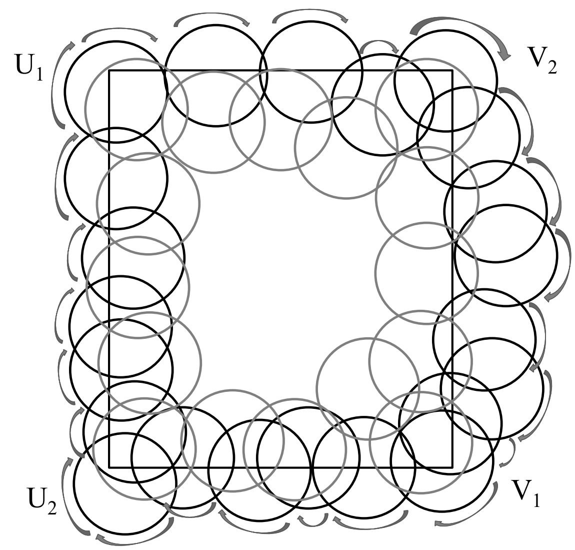


Figure 3. Flood Reinforced Barriers.

A couple of observations must be highlighted.

- The resulting barrier must cover all four corners of the area of interest. If this is not the case, then the barrier has to be discarded.
- It is possible to optimize the resulting barrier by removing some sensor nodes in a manner similar to what we proposed in [21]. For example, if three consecutive sensors in the barrier, s_1, s_2, s_3, are such that the sensing areas of s_1 and s_3 overlap, then s_2 is redundant and can be removed. As mentioned in [21], care must be taken that by removing sensor nodes we do not cause the next barrier to cause a breach with the previous one.

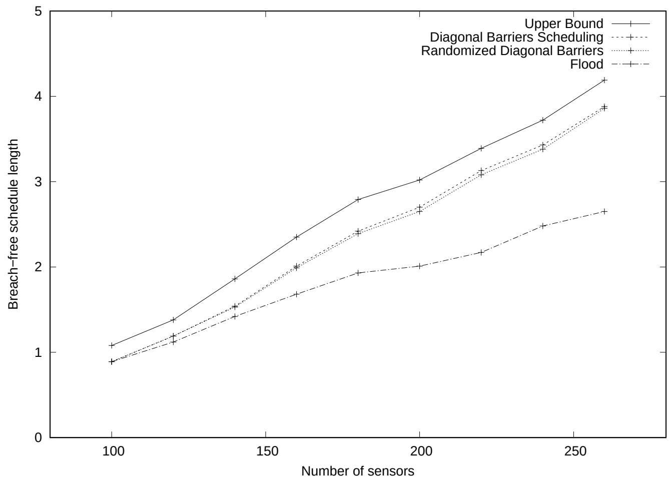


Figure 4. Number of sensors vs schedule length in square area.

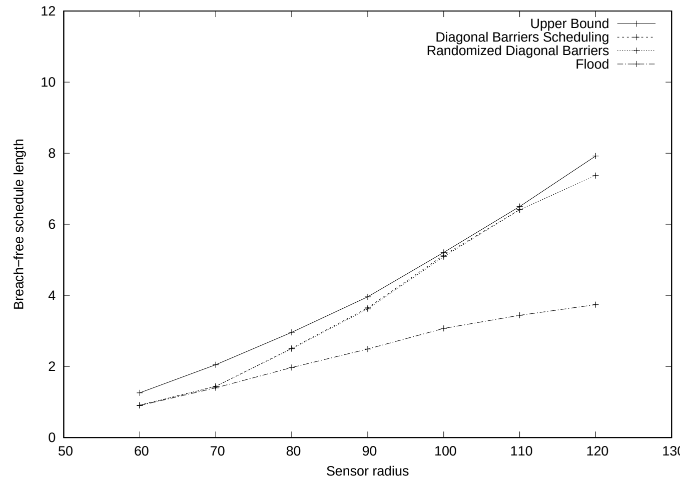


Figure 5. Radius vs. schedule length in square area.

IV. Simulation Results

In this section, we compare the performance of the three heuristics: Diagonal Barriers Scheduling (DBS), Randomized Diagonal Barriers (RDB), and Flooding (FL). Because the RDB and FL heuristics have lower complexity than the DBS heuristic, our objective is to determine if these lower complexity heuristics are capable of providing results that are similar to the higher-complexity DBS heuristic. In addition, we compare all of these against the upper bound λ, where $\lambda = \min(|\mathcal{D}_1|, |\mathcal{D}_2|)$.

The area of interest is a square of size 500×500 meters. We also simulated a rectangular area of 400×600 meters. Sensor nodes are randomly deployed in each area, ranging from 100 to 260. In addition, the radius of the sensing area of sensors ranges from 60 to 120 meters. Every point in our plots corresponds to the average of 100 simulations.

We start with a sensing area of 500×500 meters. Figure 4 plots the number of sensors vs. the resulting reinforced breach-free schedule length. The sensor radius is maintained at 90 meters. As the number of sensors increases, the size of $\mathcal{D}_1$ and $\mathcal{D}_2$ also increase, along with the upper bound λ. Note that the upper bound is oblivious to the existence of barrier breaches, and the maximum breach-free schedule is expected to be less than this.

Heuristic DBS is able to generate schedules close to the upper bound. This is at the expense of significant computation time, in the order of several days for the largest case of 250 sensors, vs. just a few hours for RDB, and a few minutes for FL. RDB matches closely the results of DBS, and thus the random algorithm is able to find schedules close to those of a full search of all paths. Heuristic FL, on the other hand, returns significantly smaller barrier schedules.

In the next figure, we plot the sensor radius vs. the resulting reinforced breach-free schedule length. The number of sensors is maintained constant at 250. As the radius increases, the diagonal barrier sets $\mathcal{D}_1$ and $\mathcal{D}_2$ increase in size, and therefore, so does the total number of reinforced barriers from which a schedule can be obtained. The results are similar to those of the previous figure. That is, DBS is close to the upper bound,

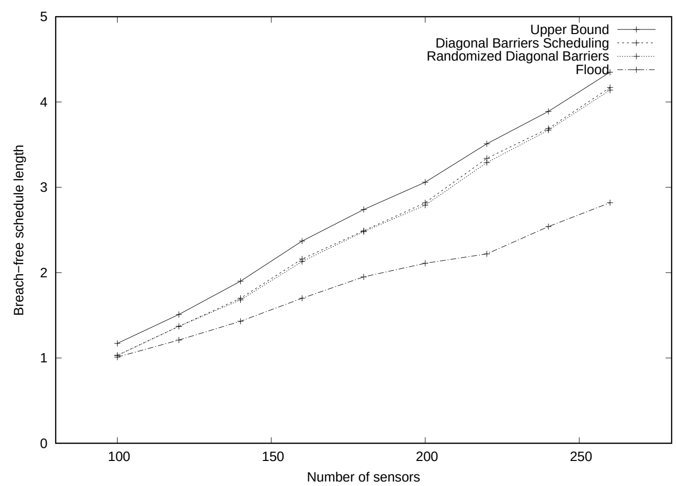


Figure 6. Number of sensors vs schedule length in rectangular area.

and RDB closely matches DBS. Also, FL is unable to obtain schedules of significant size.

The scenarios in Fig. 6 and 7 are similar to those in Fig. 4 and 5, except that the area is now a 400×600 meters rectangle. A similar behavior as before is observed in these scenarios.

V. Conclusion

We have presented two new heuristics to obtain the longest breach-free schedule of reinforced barriers in a sensor network. These were compared against the heuristic with high complexity, DBS, that we introduced in [23]. Of the two new heuristics, one clearly outperforms the other, and performs very closely to the DBS heuristic.

We have several directions for possible future work. We have used sensor ranges that are uniform in size. As mentioned earlier, using Stint as a foundation for our heuristics does not guarantee that the barriers obtained will include those of the optimal schedule. Having different sensor ranges may have an impact on this, and result on schedules of smaller size. Finally, we will also consider studying scenarios in which sensors are

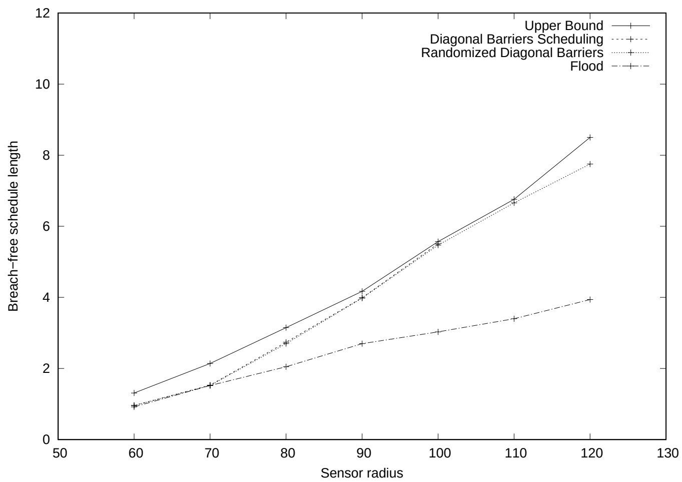


Figure 7. Radius vs. schedule length in rectangular area.

placed in more strategic locations rather than randomly. For example, a large number of sensor nodes could be located near the corners of the area and also in the center of the area, which presumably would increase significantly the number of reinforced barriers.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, vol. 38, no. 4, March 2002, pp. 393–422.

[2] C. Huang and Y. Tseng, "The coverage problem in a wireless sensor network," in ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA), 2003, pp. 115–121.

[3] H. Zhang and J. Hou, "On deriving the upper bound of α-lifetime for large sensor networks," in Proc. of The 5th ACM Int'l Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc), 2004, pp. 121–132.

[4] M. Cardei, M.T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in INFOCOM, 24th Annual Joint Conf. of the IEEE Computer and Comm. Societies, vol. 3, March 2005, pp. 976–1984.

[5] M. Thai, Y. Li, and F. Wang, "O(log n)-localized algorithms on the coverage problem in heterogeneous sensor networks," in IEEE Int'l Performance, Computing, and Comm. Conference, IPCCC, April 2007, pp. 85–92.

[6] S. Gao, X. Wang, and Y. Li, "p-percent coverage schedule in wireless sensor networks," in Proc. of 17th Int'l Conference on Computer Communications and Networks, ICCCN, August 2008, pp. 1–6.

[7] C. Vu, G. Chen, Y. Zhao, and Y. Li, "A universal framework for partial coverage in wireless sensor networks," in IEEE 28th Int'l Perf. Computing and Comm. Conf. (IPCCC), December 2009, pp. 1–8.

[8] Y. Li, C. Vu, C. Ai, G. Chen, and Y. Zhao, "Transforming complete coverage algorithms to partial coverage algorithms for wireless sensor networks," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 4, April 2011, pp. 695,703.

[9] S. Kumar, T. Lai, and A. Arora, "Barrier coverage with wireless sensors," in Proc. of the 11th Annual Int'l Conference on Mobile Computing and Networking (MobiCom), 2005, pp. 284–298.

[10] A. Saipulla, C. Westphal, B. Liu, and J. Wang, "Barrier coverage of line-based deployed wireless sensor networks," in IEEE INFOCOM, April 2009, pp. 127–135.

[11] S. Kumar, T. Lai, M. Posner, and P. Sinha, "Maximizing the lifetime of a barrier of wireless sensors," IEEE Transactions on Mobile Computing, vol. 9, no. 8, August 2010, pp. 161–1172.

[12] H. Yang, D. Li, Q. Zhu, W. Chen, and Y. Hong, "Minimum energy cost k-barrier coverage in wireless sensor networks," in Proc. of the 5th Int'l Conf. on Wireless Alg., Systems, and Applications (WASA), 2010, pp. 80–89.

[13] H. Luo et al., "Imperfection better than perfection: Beyond optimal lifetime barrier coverage in wireless sensor networks," in Proc. of The IEEE 10th Int'l Conference on Mobile Ad-hoc and Sensor Networks (MSN), December 2014, pp. 24–29.

[14] D. Li, B. Xu, Y. Zhu, D. Kim, and W. Wu, "Minimum (k,w)-angle barrier coverage in wireless camera sensor networks," Int'l Journal of Sensor Networks (IJSNET), 2014, pp. 179–188.

[15] L. Guo, D. Kim, D. Li, W. Chen, and A. Tokuta, "Constructing belt-barrier providing quality of monitoring with minimum camera sensors," in 23rd Int'l Conf. on Computer Communication and Networks (IC-CCN), August 2014, pp. 1–8.

[16] B. Xu et al., "Fortifying barrier-coverage of wireless sensor network with mobile sensor nodes," in Proc. of the 9th Int'l Conference on Wireless Algorithms, Systems, and Applications (WASA), June 2014, pp. 368–377.

[17] D. Kim, J. Kim, D. Li, S. S. Kwon, and A. Tokuta, "On sleep-wakeup scheduling of non-penetrable barrier-coverage of wireless sensors," in Proc. of the IEEE Global Communications Conference (GLOBECOM), December 2012, pp. 321–327.

[18] D. Kim et al., "Maximum lifetime dependable barrier-coverage in wireless sensor networks," Ad Hoc Networks, vol. 36, no. 1, Jan 2016, pp. 296–307.

[19] Z. Zhang, W. Wu, J. Yuan, and D.-Z. Du, "Breach-free sleep-wakeup scheduling for barrier coverage with heterogeneous wireless sensors," IEEE/ACM Trans. on Networking, vol. 26, no. 5, 2018, pp. 2404–2413.

[20] J. A. Cobb, "Improving the lifetime of non-penetrable barrier coverage in sensor networks," in IEEE 14^{th} International Workshop on Assurance in Distributed Systems and Networks (ADSN), June 2015, pp. pp. 1–10.

[21] J. Cobb, "In defense of stint for dense breach-free sensor barriers," in Int'l Conf. on Systems and Networks Communications (ICSNC), 2017, pp. 14–19.

[22] H. Kim and J. A. Cobb, "Maximizing the lifetime of reinforced barriers in wireless sensor networks," Concurrency and Computation: Practice and Experience, vol. 29, no. 23, 2017.

[23] J. Cobb, "Breach-free scheduling of reinforced sensor barriers," in Int'l Conf. on Systems and Networks Communications (ICSNC), 2021, pp. 24–29.

[24] R. Y. N. Alon and U. Zwick, "Color-coding," Journal of the ACM, vol. 26, no. 5, 1995, pp. 844–856.

 ISBN: 978-1-61208-998-0