Breach-Free Scheduling of Reinforced Sensor Barriers

Jorge A. Cobb
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080-3021
U.S.A.
Email: cobb@utdallas.edu

***Abstract*—Intrusion detection is an important function of wireless sensor networks. Due to their limited lifetime, rather than covering the entire area of interest at all times, sensors can be divided into barriers, where each barrier is a subset of sensors that prevents the intruder from crossing the area. However, a security problem was discovered, known as a *barrier-breach*, where an intruder can find a location in between two consecutive barriers that allows the area to be crossed when one barrier is replaced by the next. Given a set of barriers, deciding if there is a breach-free schedule of these barriers is intractable. This has led to the development of several heuristics. In a recent 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 reinforced barriers. However, this work did not address obtaining a breach-free schedule for these barriers. In this paper, we present a heuristic to obtain a breach-free schedule of reinforced barriers from a random placement of sensors in the area of interest. We show via simulation that in practical scenarios the heuristic achieves a schedule that is close to optimal.**

***Index Terms*—sensor networks; barrier coverage; security breaches**

I. INTRODUCTION

A Wireless Sensor Network (WSN) consists of an area of interest in which sensor nodes have been randomly placed. Due to running on batteries, sensors have a limited lifetime [1]. One important use of a WSN is intrusion detection, in which sensors monitor the area of interest and report to a base station any anomalous presence. Typically, sensors have a sensing range that is significantly smaller that the area of interest, and thus, multiple sensors need to be operating simultaneously.

Due to their limited lifetime, it is common to have more sensors than necessary to cover the area. Sensors are divided into groups, where each group covers the entire area. A sleep-wakeup schedule is created, where one sensor group is active and the remaining are asleep. Once the first group's battery is close to exhaustion, the second group is activated, and so on.

The degree to which the area of interest is covered by active sensors falls in two categories: full coverage and partial coverage. In full coverage, the entire area is covered at all times by the active sensors [2]–[5]. In partial coverage, only certain regions are covered at a time by the active sensors. Thus, any event occurring outside of covered area is not detected [6]–[8].

A particular form of partial coverage is *barrier coverage*, where each group of sensors forms a barrier across the area such that intruders are prevented from crossing undetected. There have been extensive studies of sensor barriers due to their many applications [9]–[16]. Fig. 1(a) highlights a subset of sensors that provide barrier coverage to the area. The highlighted sensors will remain active and the rest asleep until they are close to exhausting their battery power. If n disjoint barriers are constructed, the protection lasts n times the lifetime of a sensor. Fig. 1(b) shows the sensors divided into four disjoint barriers.

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.

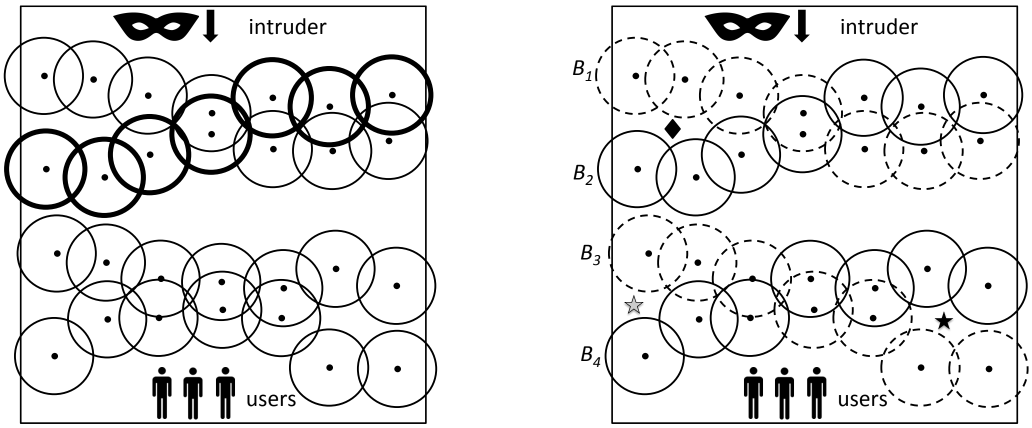
Subsequently, a vulnerability of sensor barriers, known as a *barrier breach*, was discovered [17], [18]. For some barriers, it is possible for an intruder to cross the area of interest after activating one barrier and deactivating the previous one.

Fig. 1(b) illustrates barrier breaches. Four different sensor barriers are displayed with different line types. If we use the barriers in a sequential sleep-wakeup cycle (B_1, B_2, B_3, and finally B_4), the users are protected for a total of four time units. However, the order in which the barriers are scheduled affects the effectiveness of the barriers. Instead, consider scheduling B_2 followed by B_1. In this case, an intruder could move to the point highlighted by a diamond, and after B_2 is turned off, the intruder is free to cross the area. Also, note that only one of B_3 or B_4 is of use. If we activate B_3 first, then the intruder can move to the location marked by the black star. When B_4 is activated and B_3 deactivated, the intruder can reach the users undetected. The situation is similar if B_4 is activated first, and the intruder moves to the location of the grey star.

There have been several heuristics that generate a set of sensors barriers and their breach-free schedule from randomly placed sensors [17]–[20]. In [21], it is shown that, *given* a set of disjoint barriers, obtaining the longest breach-free schedule of the given barriers is intractable, and a probabilistic algorithm is given for the problem. The complexity of finding the longest breach-free schedule of barriers from a from a random placement of nodes remains an open problem.

A stronger form of a barrier, called a *reinforced barrier*, was

 ISBN: 978-1-61208-895-2



(a) Sensor Barrier

(b) Barrier Breaches

Fig. 1. Sensor Barriers

introduced in our earlier work [22]. To illustrate this barrier, consider Fig. 2(a), in which the area of interest is a rectangle. The 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 a horizontal intrusion, while I_3 is a corner intrusion (by turning from vertical to horizontal).

To prevent these intrusions, consider Fig. 2(b), where there is a barrier of sensors from corner U_1 to corner V_1, and another barrier from corner U_2 to corner V_2. Notice that these two barriers do not need to be disjoint. By combining these two barriers, none of the above intrusions are possible.

Heuristics to obtain the maximum number of reinforced barriers were presented in [22]. However, the issue of barrier breaches was not addressed. Note that barrier breaches are still possible, as shown in Fig. 2(c). The figure consists of two reinforced barriers drawn with different line styles. If we schedule the solid line barrier first, then an intruder can arrive from the top side. Once we switch to the dashed-line barrier, the intruder is free to exit via the right side. The issue is similar if we schedule the dashed barrier first.

In this paper, we consider the problem of obtaining a maximum-length breach-free schedule of reinforced barriers starting from a random placement of sensor nodes. We present a parameterized algorithm based on the general approach presented in [21]. The algorithm is exponential in the number of barriers, which is expected to be small and is polynomial in the number of sensor nodes. Via simulation we demonstrate that the method produces schedules of near-optimal length.

The rest of this paper is organized as follows. Section II presents background and definitions. In Section III, we present our heuristic. Simulation results are presented in Section IV. Concluding remarks are 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. Definitions

We consider a rectangular area where a set S of n sensor nodes have been deployed randomly. A *barrier* consists of a set B, $B \subseteq S$, such that there is a sequence of sensors, $s_1, s_2, \ldots, s_k$, such that the sensor ranges of s_i and s_{i+1}, $1 \leq i < k$, 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_k overlaps the opposite side of the rectangle. Barrier B_1 in Fig. 1(b) is an example. A barrier is vertical if the sides being overlapped are the top and bottom, and is horizontal otherwise.

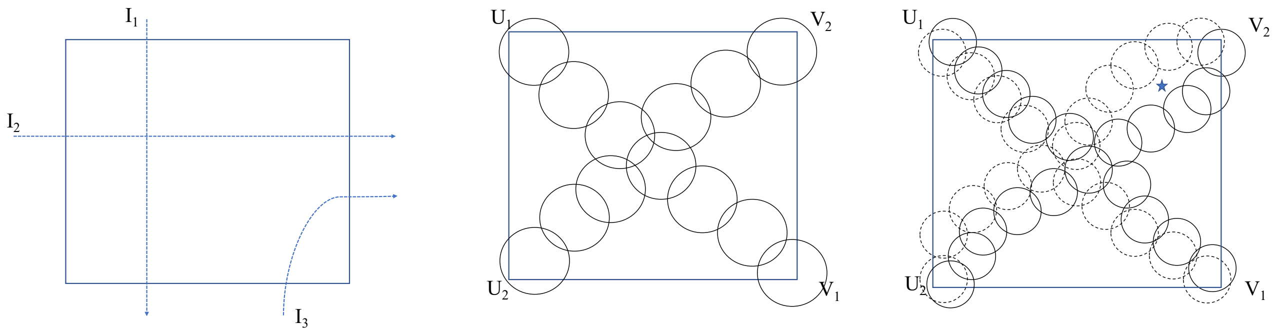
A *reinforced barrier* R is a set of sensors such that a line cannot be drawn starting from a side of the rectangle and ending at a different side without crossing the sensing area of any one of the sensors. Note that this requires the corners to be covered, and it also implies that there is a subset 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.

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 from p to the bottom of the area without overlapping the sensing area of B_2. A sequence (or schedule) of barriers $B_1, B_2, \ldots, B_k$ is *breach-free* if every pair of consecutive barriers in the sequence does not form a breach.

Similarly, an unordered 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.

B. Longest Barrier Schedule

Finding the largest number of horizontal disjoint barriers has been solved in polynomial time by Kumar et al. [11] with their algorithm known as Stint. The method builds a flow graph F where the maximum flow corresponds to the number of



(a) Area Intrusions (b) Reinforced Barrier (c) Reinforced Barrier Breach

Fig. 2. Reinforced Sensor Barriers

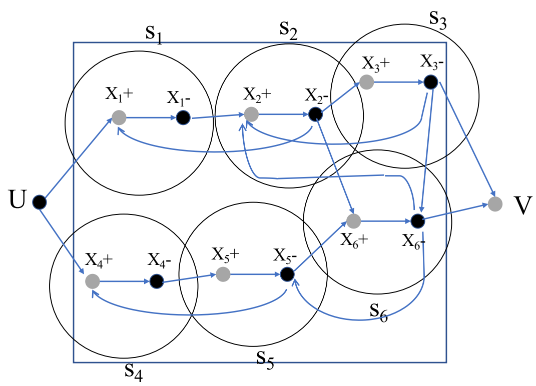


Fig. 3. Stint Maximum Flow Method

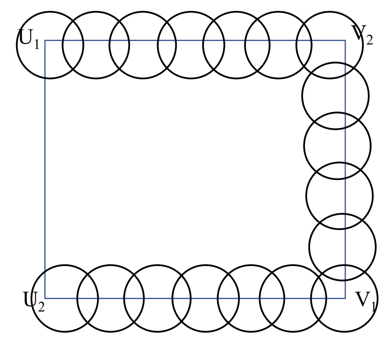


Fig. 4. Reinforced Barrier Extreme Case

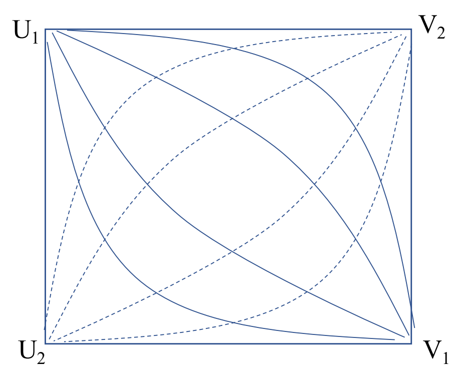


Fig. 5. Diagonal Barriers

sensor barriers, and a path with non-zero flow corresponds to a barrier. A brief outline of the method is as follows, and a sample graph F is illustrated in Fig. 3.

Create two nodes U and V representing the left and the right borders, respectively. Then, for each sensor node s_i, create two nodes, x_i+ and x_i-, with a directed edge (x_i+, x_i-) of capacity one. This edge corresponds to the life of the sensor. All other edges have a capacity of infinity. For every sensor s_i overlapping the left border, add the directed edge (U, x_i+), and for every sensor s_j overlapping the right border, add the directed edge (x_j-, V). Finally, for every pair of sensors s_i and s_j whose sensing area overlaps, add an edge (x_i-, x_j+) and (x_j-, x_i+).

It is easy to show that a barrier-cover corresponds to a path from U to V in F. Since the capacities are integers, the maximum flow f in F is an integer, which corresponds to f edge-disjoint paths, and thus f node-disjoint barriers.

Most heuristics, such as [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. Another approach [21] is to simply try all possible schedules obtained from the Stint barriers. If the longest schedule is of length l, then the approach is exponential in l, but polynomial in the number of sensor nodes. Due to this exponential growth, a probabilistic algorithm was presented in [21] that finds the longest schedule with high probability.

Note that the barriers chosen for the above methods are obtained from the output of Stint. There is, however, no guarantee that the set of barriers that can generate the longest possible schedule are the barriers obtained from Stint.

III. Breach-Free Reinforced Stint Barriers

In this section, we present our heuristic for obtaining the longest breach-free schedule of reinforced barriers. As shown in Fig. 2(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. Consider for example Fig. 4 where there is a sequence of sensor nodes covering all three sides $\overline{U_1V_2}$, $\overline{V_2V_1}$, and $\overline{V_1U_2}$. Given that the diagonal barriers do not need to be disjoint (they will be activated concurrently), then this sequence of nodes can be thought of as two diagonal barriers, the first from U_1 to V_1 along sides $\overline{U_1V_2}$ and $\overline{V_2V_1}$, and the second from U_2 to V_2 along sides $\overline{U_2V_1}$ and $\overline{V_1V_2}$.

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. 5. Let $\mathcal{D}_1 = \{B_{1,1}, B_{1,2}, B_{1,m_1}\}$, where $|\mathcal{D}_1| = m_1$, be a maximal set of disjoint barriers from U_1 to V_1. Similarly, let $\mathcal{D}_2 = \{B_{2,1}, B_{2,2}, B_{2,m_2}\}$, $|\mathcal{D}_2| = m_2$, be a maximal set of disjoint barriers from U_2 to V_2. Then, the union of any two barriers $B_{1,i}$ and $B_{2,j}$, where $B_{1,i} \in \mathcal{D}_1$ and $B_{2,j} \in \mathcal{D}_2$, form a reinforced barrier, $R_{i,j}$. Note in particular that $B_{1,i}$ and $B_{2,j}$ do not need to be disjoint since they will be activated simultaneously. We denote the set of all reinforced barriers with $\mathcal{R}$, i.e., $\mathcal{R} = \bigcup_{i,j} R_{i,j}$.

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 $\mathcal{D}_2$. To obtain $\mathcal{D}_1$, the flow graph F_1 is built with an arc from U_1 to each node x_i+, where s_i is a sensor whose sensing are overlaps U_1's corner. Also, an arc is made from each x_j- to V_1, where the sensor area of x_j overlaps V_1's corner. The arcs between sensor nodes are the same as before. A similar approach using U_2 and V_2 will yield the set $\mathcal{D}_2$.

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 $\mathcal{R}$. An edge exists from an element $R_{i,j}$ to an element $R_{k,l}$ if the pair $(R_{i,j}, R_{k,l})$ does not constitute a breach. Obtaining the longest breach-free schedule is equivalent to the problem of finding the longest path in G starting at an arbitrary node and without repeating nodes in the path.

The above approach is similar to the one used in [21], except that the problem considered is obtaining a maximum breach-free schedule of horizontal barriers. The barriers are obtained from Stint, and a graph is built such that an arc corresponds to a pair of horizontal barriers that do not form a breach.

There is a significant difference in the case of reinforced barriers that does not occur in horizontal barriers. That is, reinforced barriers are not independent of each other. If a barrier $R_{i,j}$ is used somewhere in the schedule, then for any i, $R_{i,l}$ cannot appear in the same schedule. This is because the diagonal barrier $B_i \in \mathcal{D}_i$ takes part in both reinforced barriers $R_{i,j}$ and $R_{i,l}$, and barrier B_i can only appear once in a schedule. We refer to the pair $R_{i,j}$ and $R_{i,l}$ as being *incompatible* barriers. Similarly, barrier $R_{i,j}$ is incompatible with barriers $R_{k,j}$ for all k.

Note that, because incompatible barriers cannot appear in a schedule, then the length of the schedule is upper bounded by $\min(|\mathcal{D}_1|, |\mathcal{D}_2|)$, i.e., $\min(m_1, m_2)$. Without this restriction, the length of the schedule could be as large as $m_1 \cdot m_2$.

This restriction on the length of the schedule is of significant consequence, because finding the longest path in a graph is an NP-Complete problem. A parameterized algorithm on the length l of the schedule can be obtained using dynamic programming combined with exploring all possible subsets of the set of barriers [21], and hence, it is exponential in the number of barriers. Since l is bounded by the number of barriers, the running time is significant for nontrivial problems. This motivated the authors of [21] to present a more efficient but probabilistic algorithm.

On the other hand, with reinforced barriers, the longest path of the graph is bounded by $\min(m_1, m_2)$, which yields a significantly smaller number. In addition, a diagonal barrier must begin with a sensor whose area overlaps a corner, while for horizontal barriers any sensor overlapping a side border can be used as a starting point. Therefore, we leave the possibility of a probabilistic algorithm for future work, and we consider all possible paths of length at most $\min(m_1, m_2)$.

As a final remark, the barriers obtained from Stint are not guaranteed to be the set of barriers from which an optimal schedule is obtained. Nonetheless, $\min(m_1, m_2)$ is an upper bound on the length of a breach-free schedule of reinforced barriers. The complete method is shown in Algorithm 1.

IV. Simulation Results

In this section, we compare the performance of the Breach-Free Reinforced Barriers (BFRB) algorithm against two upper bounds. Our objective is to determine how close our algorithm is in obtaining an optimal solution.

Our algorithm is compared against the Minimum Intervention Paths (MIP) heuristic [22]. This is a heuristic we presented for the problem of finding the maximum number of reinforced barriers. Since it does not take barrier breaches into account, we expect it to yield longer schedules than those of our BFRB algorithm. MIP is based on a greedy strategy that chooses a pair of diagonal barriers from $\mathcal{D}_1$ and $\mathcal{D}_2$ that overlap the least number of other barriers. In this way, a greater number of barriers are available for the subsequent round of the algorithm. Our algorithm is also compared against the upper bound m, where $m = \min(|\mathcal{D}_1|, |\mathcal{D}_2|)$. It is impossible to obtain a schedule longer than m, regarldess of whether breaches are present or not.

The area of interest is a square of size 500×500 meters. We also simulated a rectangular area of dimension 400×600 meters. Sensor nodes are randomly deployed in each area, ranging from 100 to 260. In addition, the radius of the sensing area for sensors ranges from 60 to 130. Every point in our plots corresponds to the average of 100 simulations.

Fig. 6 plots 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

Algorithm 1 ***Breach-Free Reinforced Barriers***

Inputs: sensor set S and rectangular area A.

Output: breach-free schedule of reinforced barriers.

$(U_1, V_2, V_1, U_2) \leftarrow$ the four clockwise corners of A.
$F \leftarrow (V_F, E_F)$; // (flow graph)
$V_F \leftarrow \{U_1, V_1\}$; $E_F \leftarrow \emptyset$;
for each s, $s \in S$, **do**
  $V_F \leftarrow V_F \cup \{x_s+, x_s-\}$; $E_F \leftarrow E_F \cup \{(x_s+, x_s-, 1)\}$;
  $E_F \leftarrow E_F \cup \{(U_1, x_s+, \infty)\}$ if $U_1 \in$ *sensing_area*(s);
  $E_F \leftarrow E_F \cup \{(x_s-, V_1, \infty)\}$ if $V_1 \in$ *sensing_area*(s);
  for each s', $s' \in S \wedge s \neq s' \wedge$ *overlapping_sensing_area*(s, s'), **do**
    $E_F \leftarrow E_F \cup \{(x_s+, x'_s-, \infty)\} \cup \{(x'_s+, x_s-, \infty)\}$;
  end for
end for
$F' \leftarrow$ *Ford-Fulkerson-Max-Flow*(F);
$\mathcal{D}_1 \leftarrow \langle\rangle$; // empty sequence of diagonal barriers)
for each path P in F' with non-zero capacity **do**
  $\mathcal{D}_1 \leftarrow \mathcal{D}_1 : barrier(P)$; // add the barrier
    // corresponding to path P
end for
$m_1 \leftarrow |\mathcal{D}_1|$;
Obtain similarly $\mathcal{D}_2$ from U_2 and V_2;
$m_2 \leftarrow |\mathcal{D}_2|$;
$m \leftarrow min(m_1, m_2)$;
for each i and j, $1 \leq i \leq m_1 \wedge 1 \leq j \leq m_2$, **do**
  $R_{i,j} \leftarrow \mathcal{D}_1(i) \cup \mathcal{D}_2(j)$;
end for
$G \leftarrow (V_G, E_G)$; // (breach graph)
for each i and j, $1 \leq i \leq m_1 \wedge 1 \leq j \leq m_2$, **do**
  $V_G \leftarrow V_G \cup \{R_{i,j}\}$;
end for
for each i, j, k, l, $1 \leq i, k \leq m_1 \wedge 1 \leq j, l \leq m_2$, **do**
  $E_G \leftarrow E_G \cup \{(R_{i,j}, R_{k,l})\}$
    if $(R_{i,j}, R_{k,l})$ is not breached;
end for
$Q \leftarrow$ longest path (length at most m) in G.
return Q

can be obtained. Note that BRFB remains significantly close the the upper bound of m, and therefore, the breach-free schedule lengths found are close to the maximum possible schedule (breach-free or not).

We observe that our MIP heuristic, which is oblivious to breaches, produces the longest possible schedules. We have no formal results on the optimality of MIP, and it is unlikely optimal, but from this performance it deserves further study.

Figure 7 plots the number of sensors vs. the resulting reinforced breach-free schedule length. The sensor radius is maintained at 90. The results are similar to those of the previous figure. BRFB obtains schedules that are close in length to those of the strict upper bound.

It is worth noticing that the number of schedules obtained is relatively small. This is related to the fact that for a diagonal barrier to exist, there has to be a sensor covering each of the corners of the area. Given the random placement of sensors, the number of sensors in these positions are few. In addition, due to the upper bound m, which is the minimum of the two sets of diagonal barriers, we expect the total number of reinforced barriers (breach-free or not) to be small.

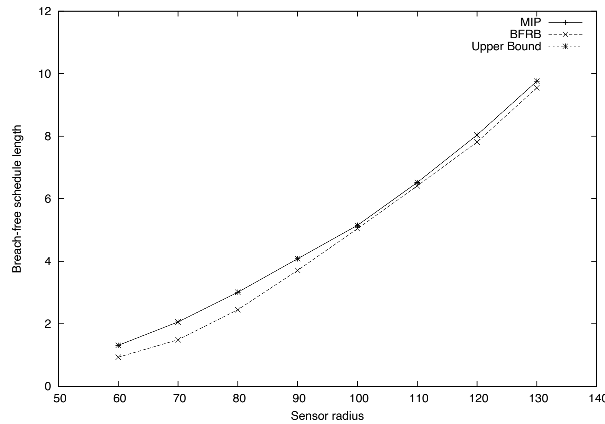


Fig. 6. Radius vs. schedule length in square area.

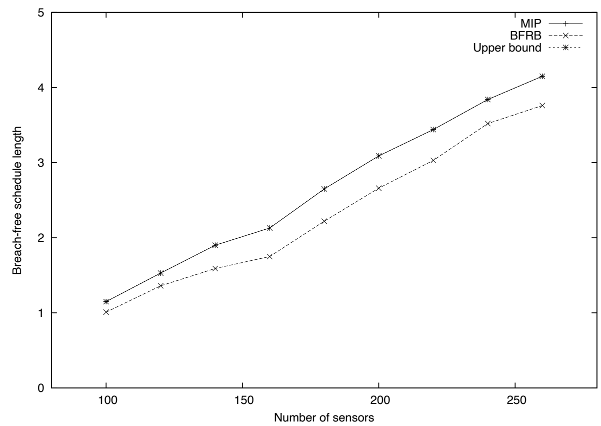


Fig. 7. Number of sensors vs schedule length in square area.

Fig. 8 and Fig. 9 are similar to Fig. 6 and Fig. 7, except that the area is now a 400×600 rectangle. As before, BRFB obtains schedules that are close in length to those of the strict upper bound.

V. Concluding Remarks and Future Work

We presented a heuristic for the reinforced breach-free barriers problem, and we have shown that it performs well, achieving schedule lengths close to the upper bound. The heuristic suffers from the drawback that all schedules of length m obtained from $\mathcal{R}$ have to be examined. Note that $|\mathcal{R}| = m_1 \cdot m_2$, so a relatively small number of barriers, say, $\mathcal{D}_1 = \mathcal{D}_2 = 10$, yields a significantly number of possible reinforced barriers, $|R| = 100$.

We considered two approaches to examine all schedules. The first is to use a dynamic programming technique similar to that in [21], where, for every possible subset of $\mathcal{R}$, a value

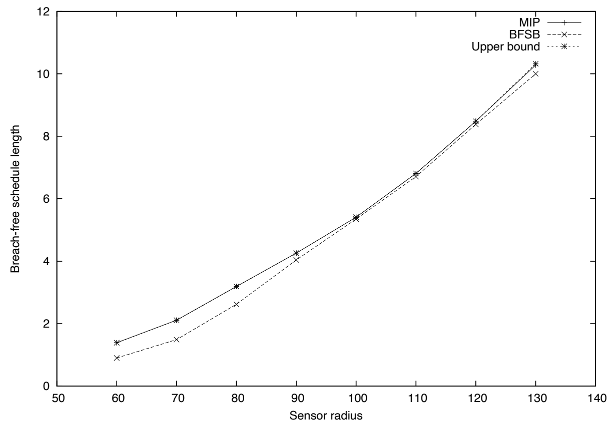


Fig. 8. Radius vs. schedule length in rectangular area.

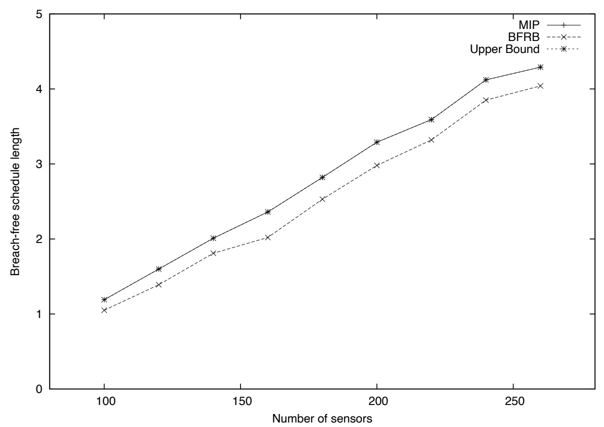


Fig. 9. Number of sensors vs schedule length in rectangular area.

(the length of the longest schedule using the subset) needs to be maintained. Above, this would require 2^{100} values.

The second approach, which we adopted, is to take advantage of the small value of m, and build all possible schedules of length m, ensuring that each addition to the schedule is in agreement with the breach-free graph G, and furthermore, that any new barrier added to the schedule is not incompatible with earlier barriers in the schedule. Nonetheless, in the worst case, the number of steps required is $|\mathcal{R}|^m$, which becomes infeasible as m grows. Given the above difficulties, we will explore probabilistic algorithms in the future and compare their performance against the known upper bound.

Finally, in earlier work on breach-free horizontal barriers, we developed algorithms that do not use the Stint method as a source of barriers, but rather developed barriers in a top-down approach ensuring that each new barrier does not create a breach with earlier barriers [19] [20]. A similar approach might be possible for the case of reinforced barriers, but barriers would have to be constructed in parallel over all sides. We will also leave this approach for future work.

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.

[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.

[4] Cardei, M., Thai, M.T., 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.

[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.

[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.

[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.

[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.

[13] H. Luo, H. Du, D. Kim, Q. Ye, R. Zhu, and J. Zhang, "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.

[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.

[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.

[16] B. Xu, D. Kim, D. Li, J. Lee, H. Jiang, and A. Tokuta, "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.

[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, H. Kim, D. Li, S. S. Kwon, A. O. Tokuta, and J. A. Cobb, "Maximum lifetime dependable barrier-coverage in wireless sensor networks," Ad Hoc Networks, vol. 36, no. 1, Jan 2016.

[19] 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.

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

[21] 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.

[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.