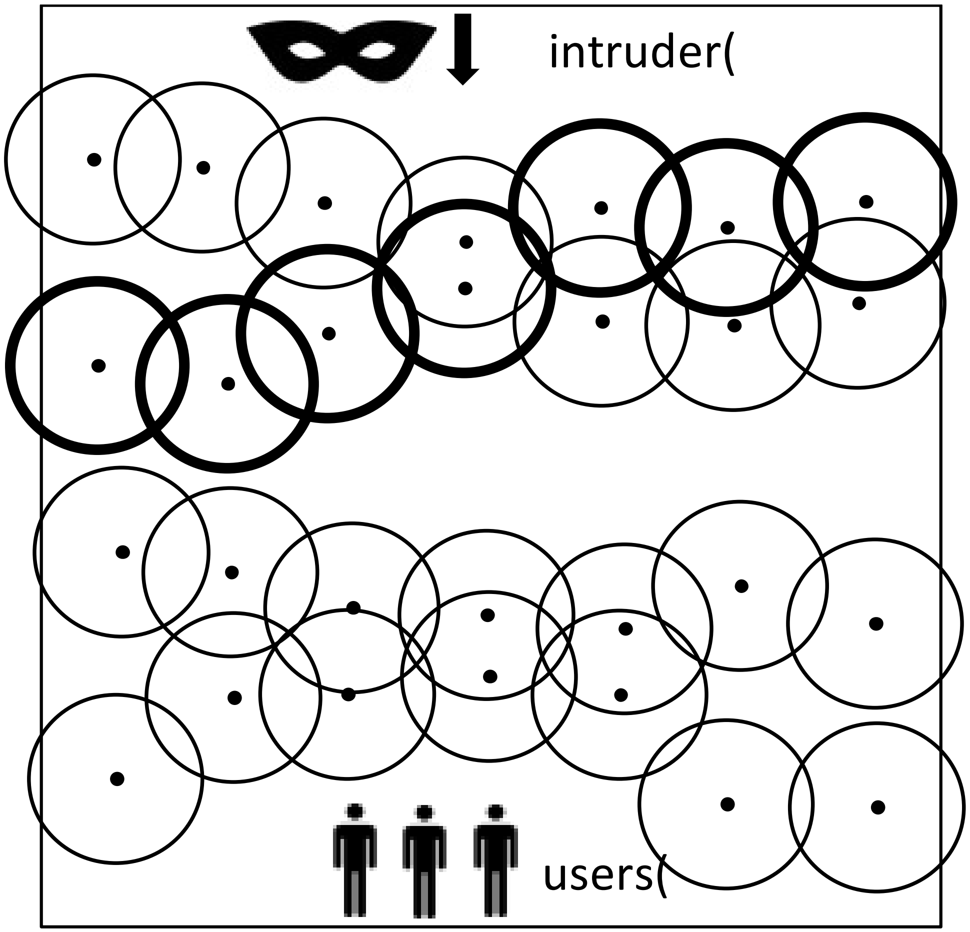
On Improving the Efficiency of Breach-Free Scheduling of Reinforced Sensor Barriers

Jorge A. Cobb

Outline

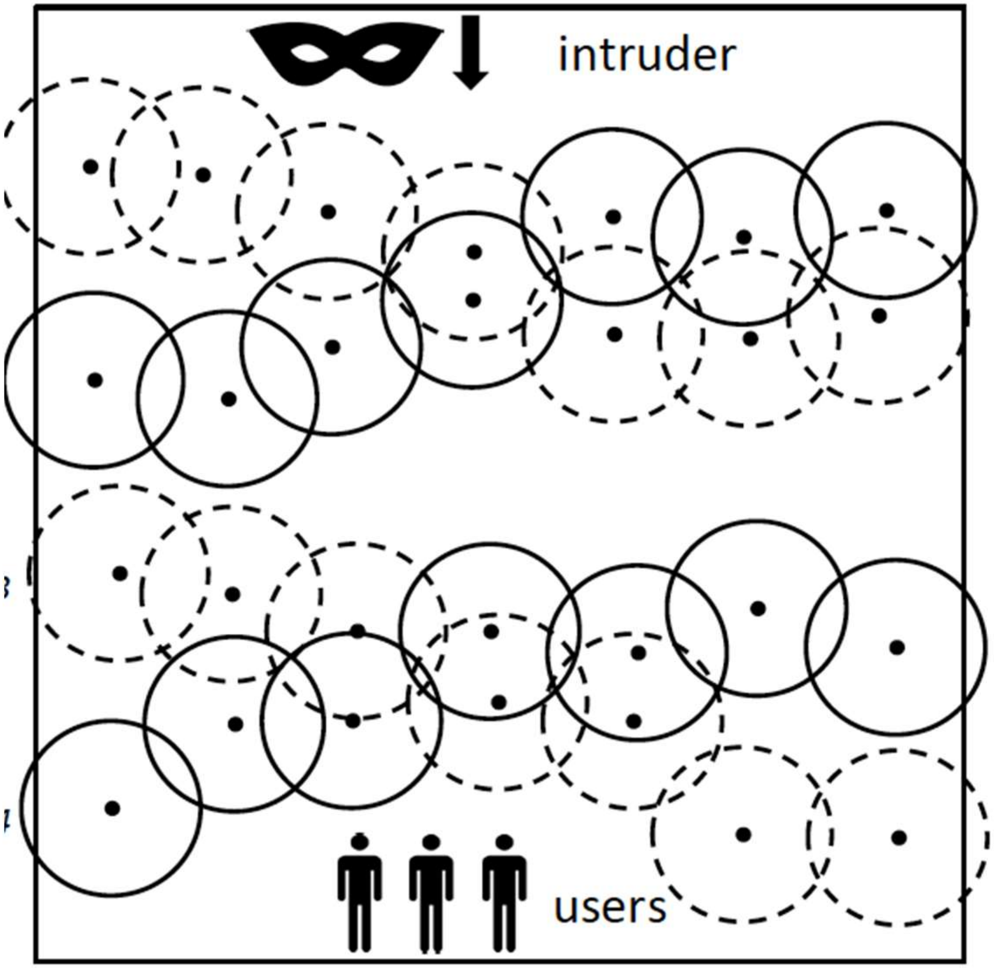
- Sensor barriers (horizontal)
  - Barrier breaches
  - Breach-free scheduling of barriers
- Reinforced barriers
  - Breaches in reinforced barriers
  - Breach-free scheduling of reinforced barriers
- Efficient heuristics
  - Random walk
  - Flooding

Sensor Barriers



- Consider an area randomly covered with wireless, battery-powered, computing nodes with sensing capabilities.
- We assume the sensing range of each node is circular and of equal size.
- A **barrier** cover divides the original area in two horizontal sections (see highlighted sensors)
- It is a form of *partial* coverage of the area.
- Four barriers are possible in the figure.

Scheduling Barriers

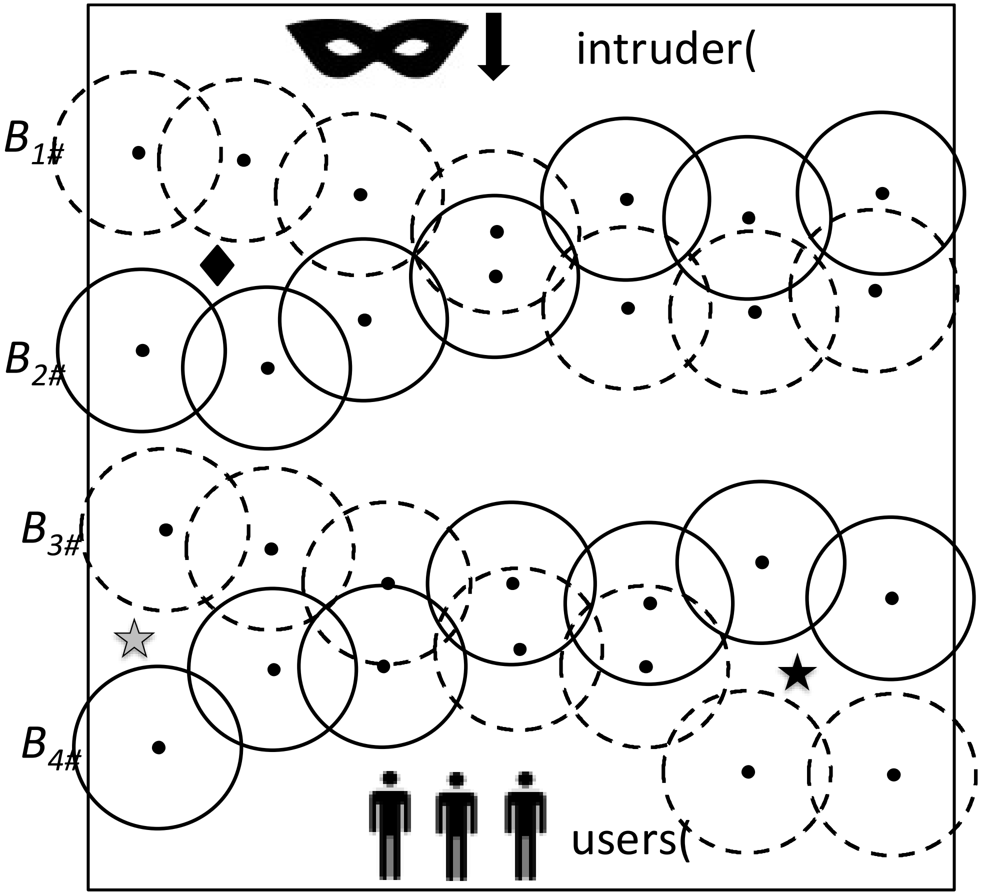


- If all sensors are active at the same time, then
  - network lifetime = lifetime of a sensor node.
- If we use the barriers in a sequential wakeup-sleep cycle (B_1, B_2, B_3, and finally B_4), the users are protected for a total of four sensor lifetimes.

Maximum Number of Barriers

- The maximum number of sensor barriers has been solved in polynomial time by Kumar et. al. (Stint algorithm)
- It builds a flow graph consisting of all sensor nodes 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.
- The graph 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.

Barrier Breaches



- Care must be taken with respect to the order in which B_1 and B_2 are scheduled.
  - Enabling B_2 first, then B_1, allows an intruder to move to the solid diamond.
  - When B_1 is enabled, the intruder can reach the users.
  - Enabling B_1 first and then B_2 does not have this problem
- Only one of B_3 and B_4 is of use.
  - Schedule B_3;B_4 and B_4;B_3 have the same problem.

Definition: (barrier-breach).

- A point p in the plane is called a *barrier-breach* with respect to an ordered pair (B_1 , B_2) of sensor barriers which are observing an intruder trying to move from a position x_1 (e.g., on top border) to another position x_2 (e.g., bottom border) if
  - p is outside the sensing range of B_1 and B_2,
  - B_1 cannot detect an intruder moving from x_1 to p, and
  - B_2 cannot detect an intruder moving from p to x_2.

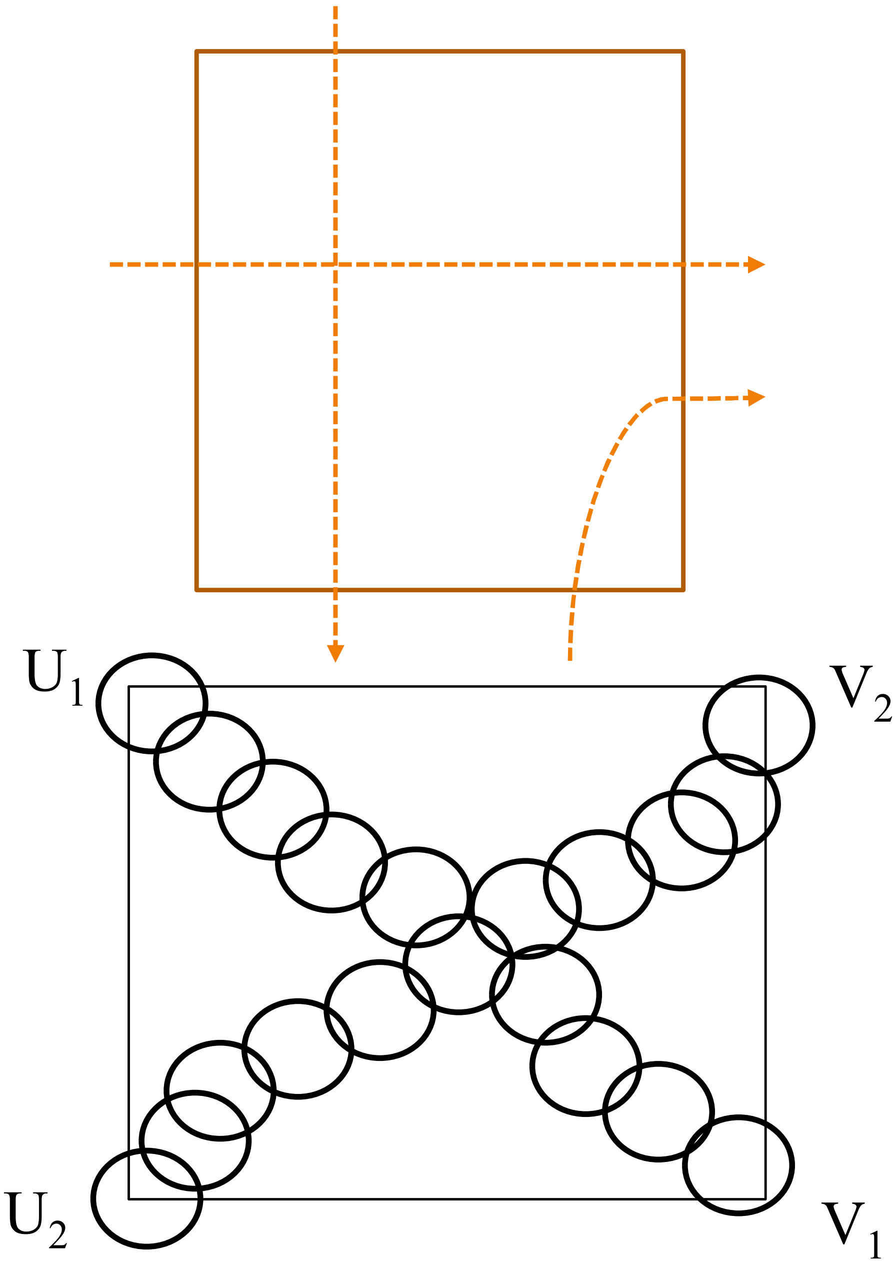
- We desire all barrier schedules to be breach-free, i.e., every pair (B_i, B_j) of consecutively scheduled barriers have no breach.

Maximum Breach-Free Schedule

- The complexity of finding the barriers and their associated breach-free schedule such that the schedule length is maximized remains an open question.
- However, **given** a set of disjoint barriers, finding the longest breach-free schedule is NP-Complete (Zhang, et al.)
- Various heuristics have been presented for this problem.

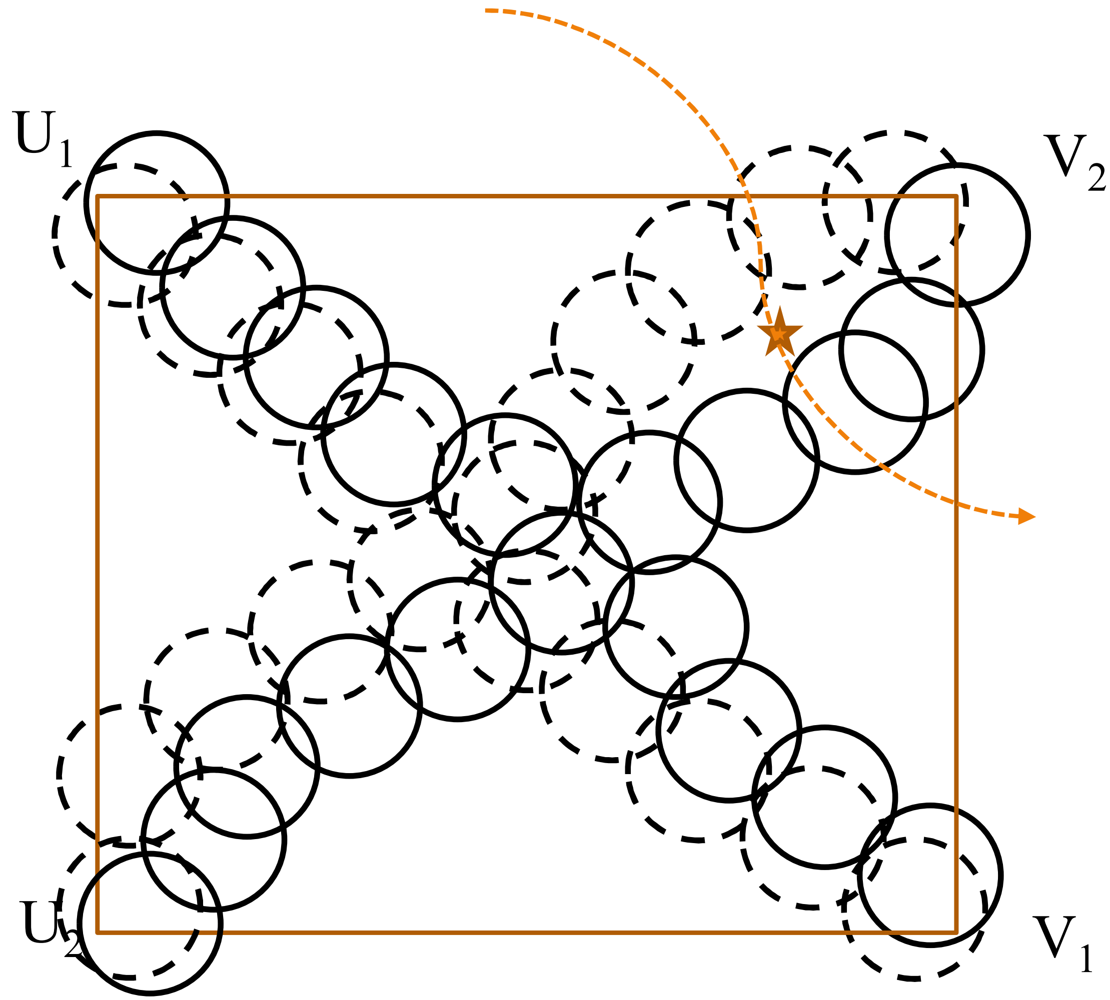
Reinforced Barriers

- In earlier work we presented **reinforced barriers.**
  - Must prevent movement from any side of the area to any other side
  - This requires a barrier to consist of two diagonal barriers
    - One from corner U_1 to corner V_1
    - Another from corner U_2 to corner V_2
- Several heuristics were presented for this problem.



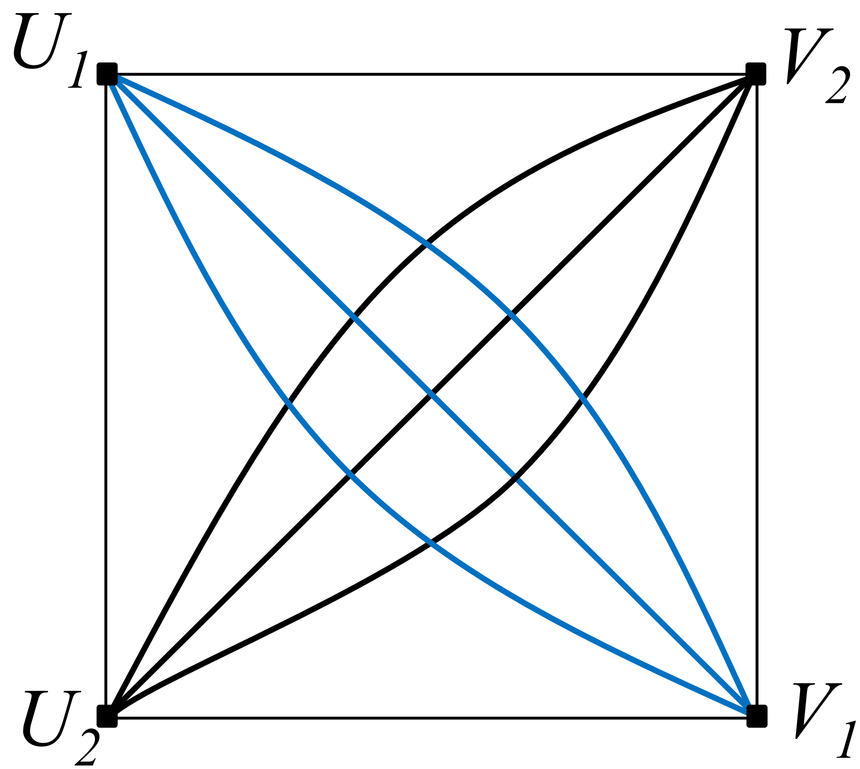
Breaches in Reinforced Barriers

- A pair of reinforced barriers can also have a breach.
- Assume the solid-line barrier is scheduled before the dashed-line barrier.
- Then, an intruder can cross from the top side to the right side via the point with the star.



Earlier Heuristic

- In ICSNC 21 we proposed using a variation of Stint to find the largest disjoint set of diagonal barriers D_1 from U_1 to V_1, and the largest disjoint set D_2 of diagonal barriers from U_2 to V_2.
- A conflict graph G = (V,E) is built with
  - **$V = D_1 \times D_2$**
  - (B_1,B_2) is in E if ordered pair (B_1,B_2) is beach-free.



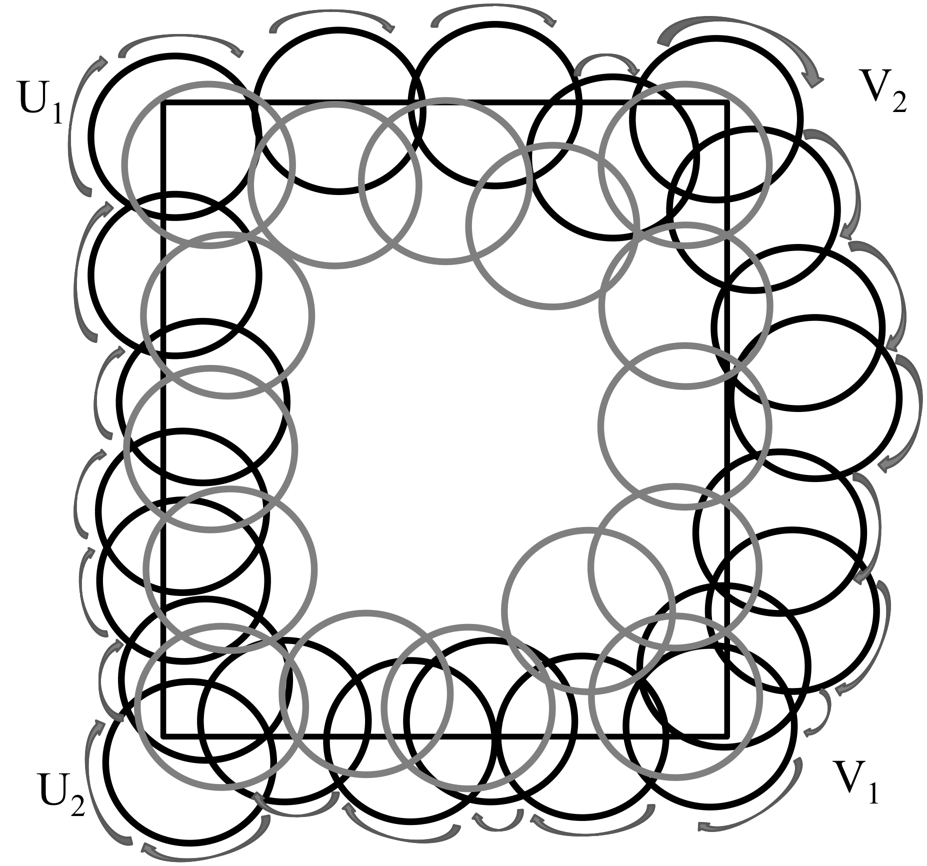
- Graph G is then explored to find its longest directed path.
- Any barrier in D_1 or in D_2 can occur only ONCE in the path

Efficient Heuristic

- Our earlier heuristic had high complexity due to exploring all paths in G (longest path in a directed graph is NP-complete)
- We thus propose two alternative heuristics
  - Flooding
  - Randomized walk in G

Flooding

- We begin with the outmost corner node at U_1, and follow the outer "rim" of the sensors until we return to U_1.
- The first barrier in our figure would be the dark circles.
- We repeat, with the second barrier in gray.



Randomized Diagonal Barriers

- Here, we also use graph G, where $V = D_1 \times D_2$.
- Instead of finding all paths, we randomly search for the longest path in G.
- Care must be taken to ensure each barrier in D_1 and D_2 is used only once.

Simulation Results

- We compare all three heuristics.
- We also compare the upper bound, which is $\max(|D_1|,|D_2|)$.
  - This bound arises because barriers cannot be reused.
- The area of interest is a square of size 500 × 500 meters. We also simulated a rectangular area of 400 × 600 meters with similar results.
- 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.

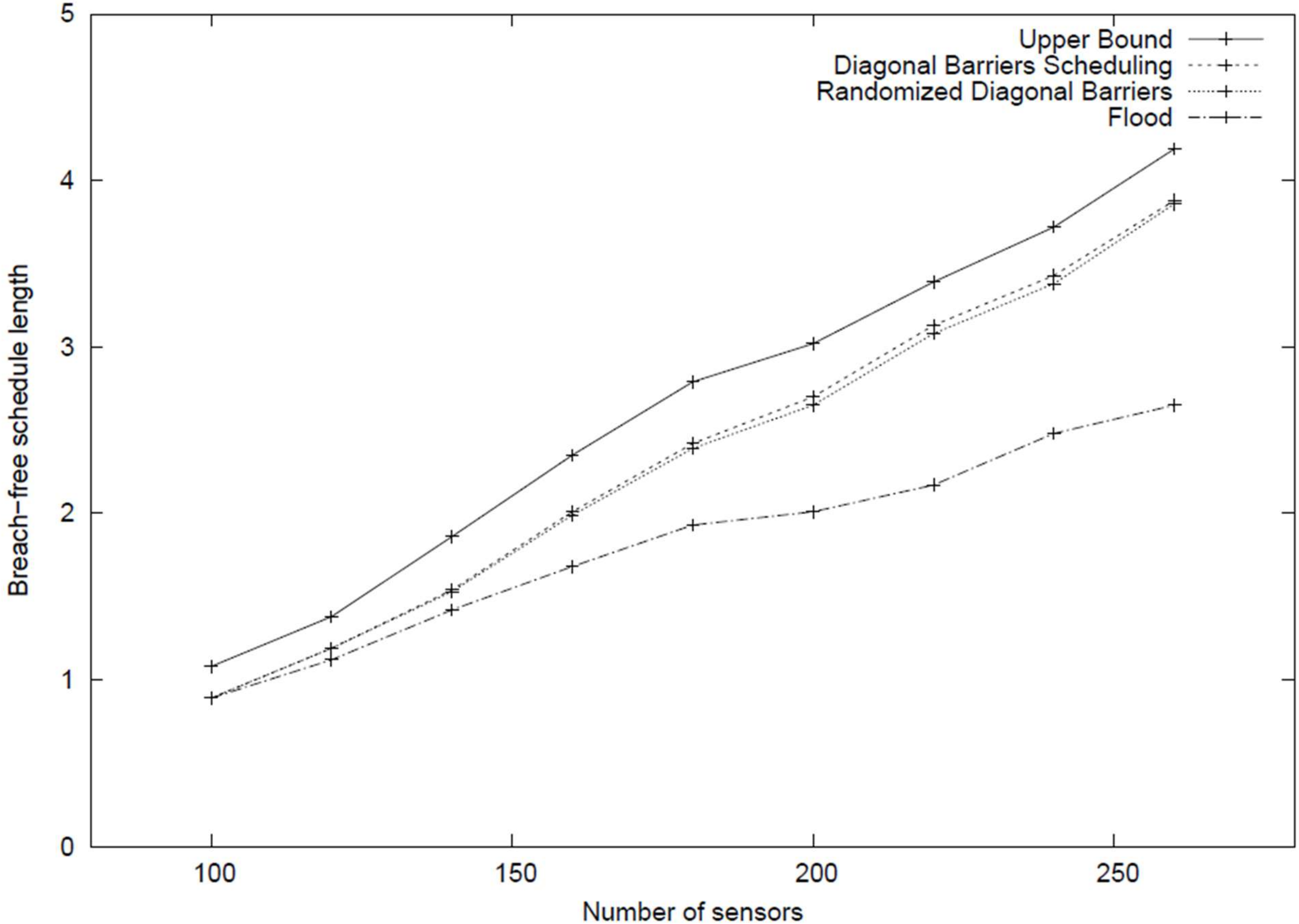


Figure 4. Number of sensors vs schedule length in square area.

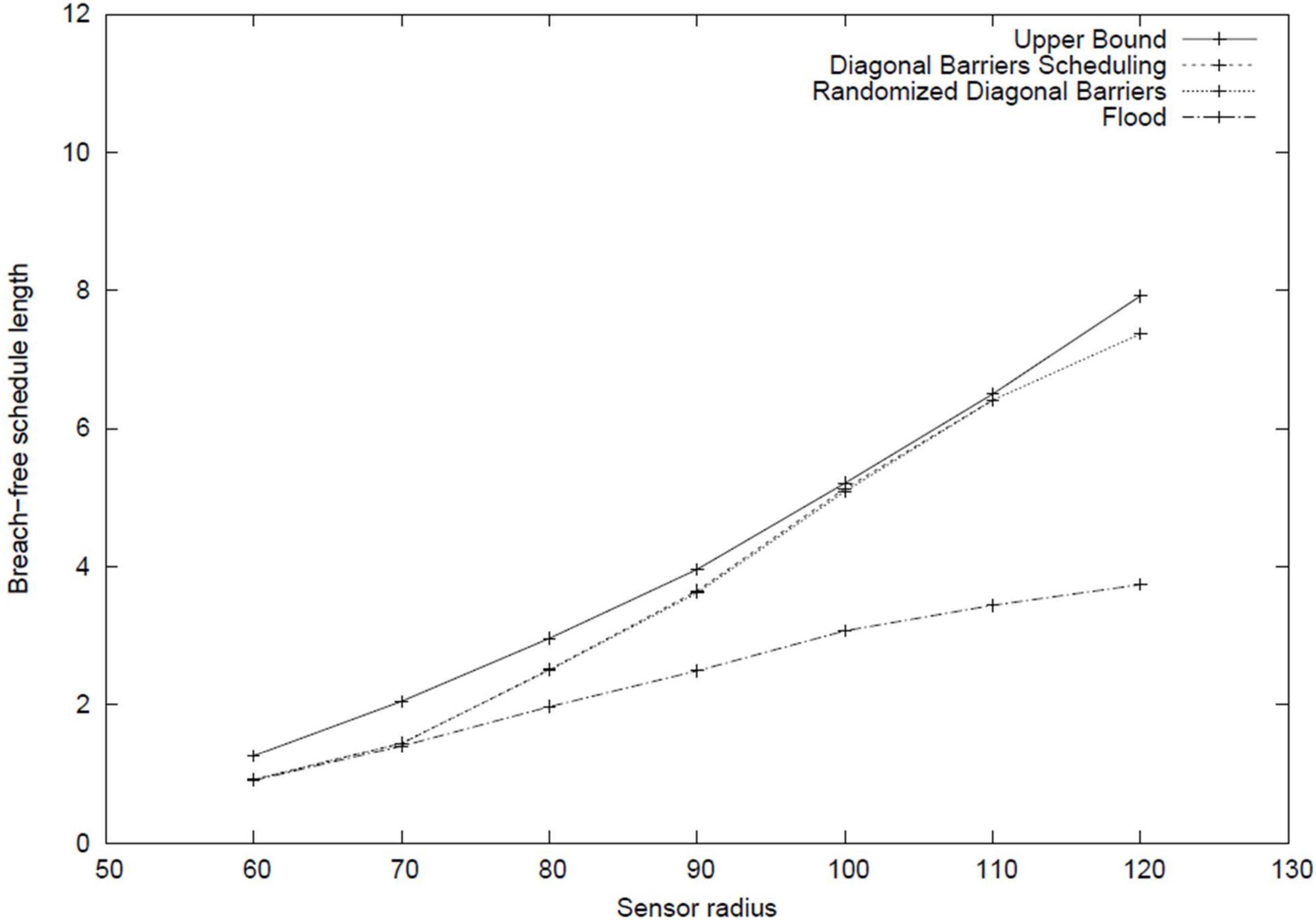


Figure 5. Radius vs. schedule length in square area.

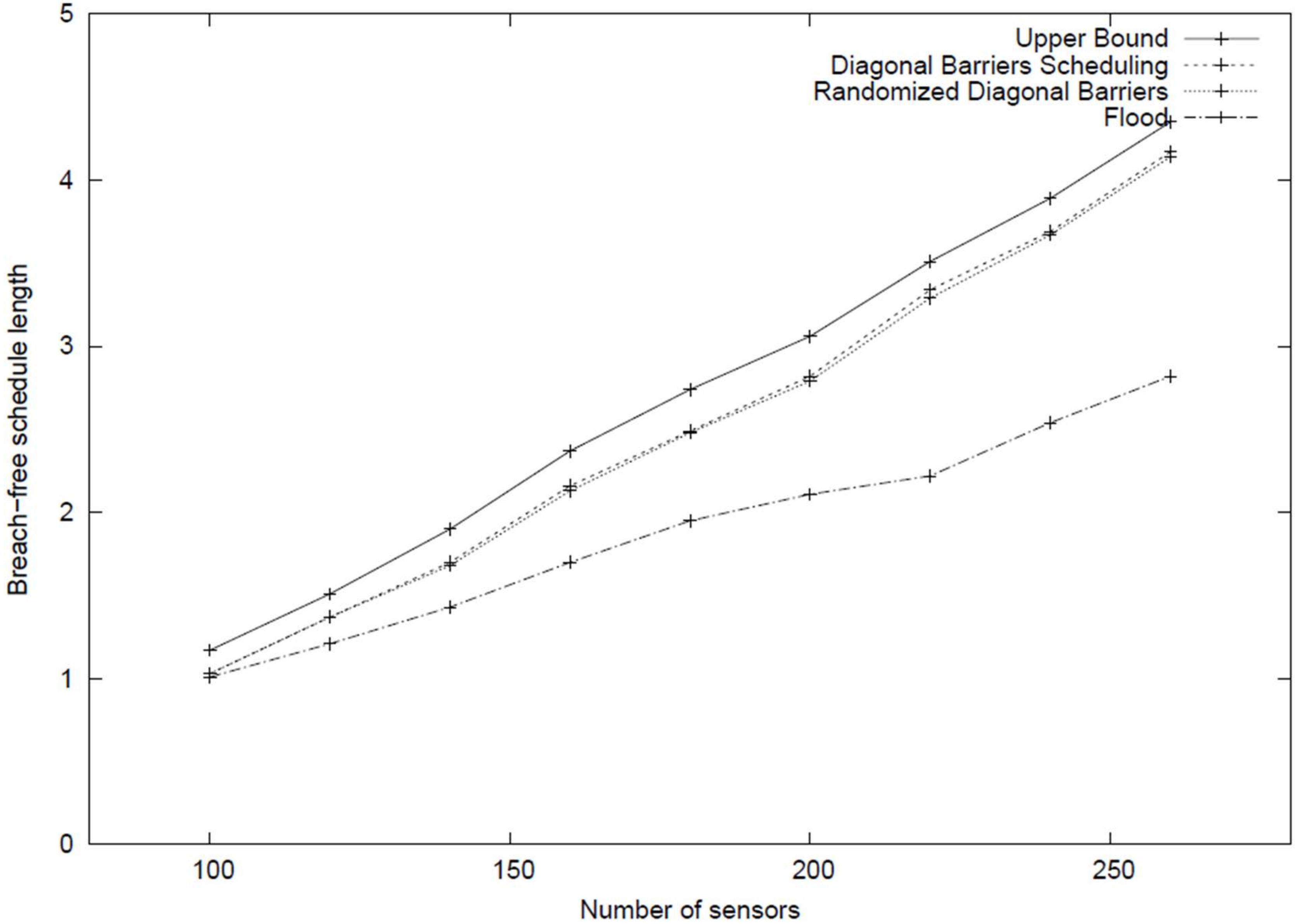


Figure 6. Number of sensors vs schedule length in rectangular area.

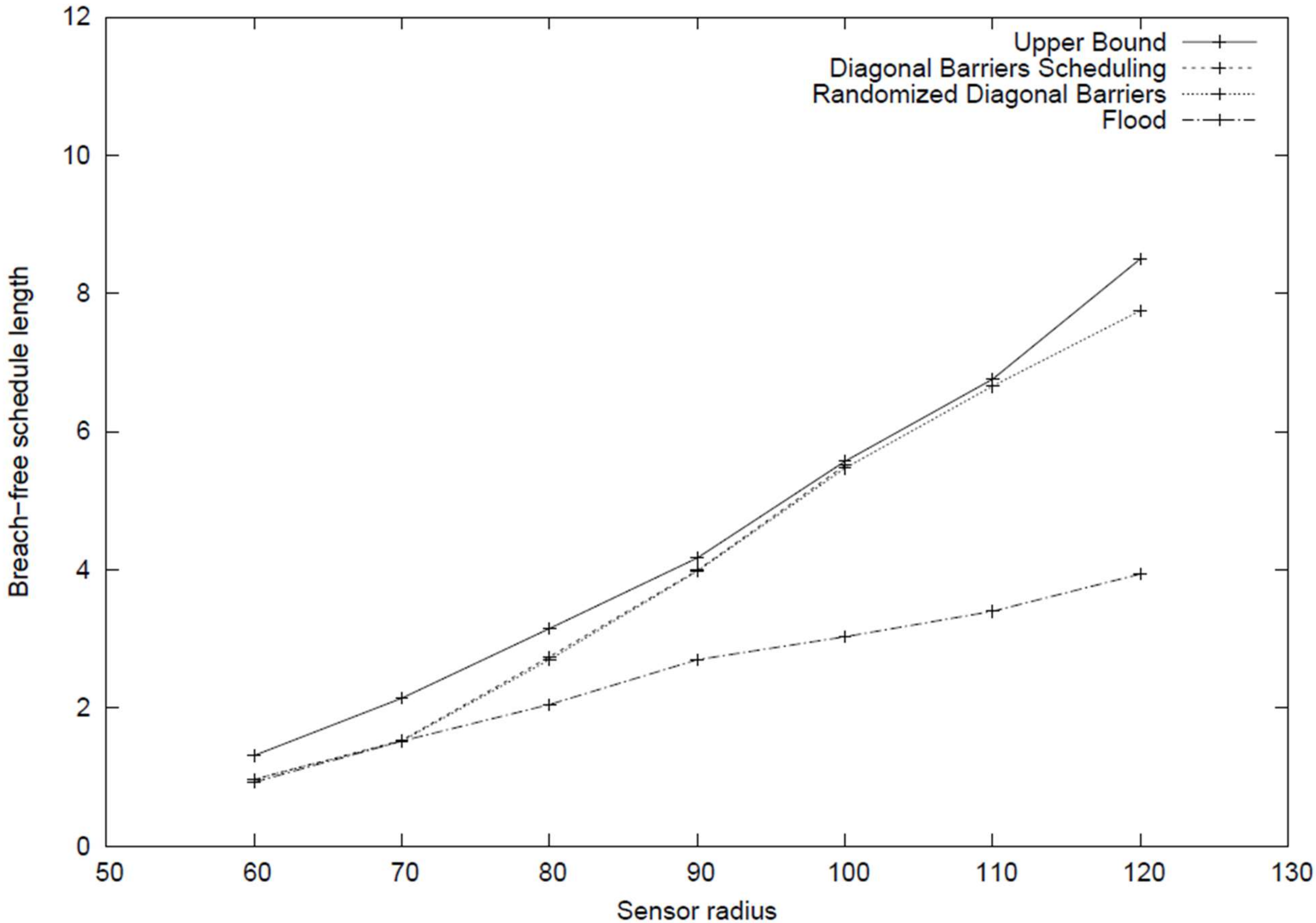


Figure 7. Radius vs. schedule length in rectangular area.

Concluding Remarks

- We have presented two new heuristics and compared them against the heuristic with high complexity.
- Clearly one new heuristic outperforms the other, and performs very closely to the original heuristic.
- We have several directions for possible future work.
  - Having nonuniform sensor ranges
  - Placing sensors in more strategic locations rather than randomly.