On the Optimal Deployment of Heterogeneous Sensing Devices

Rabie Ramadan  
Department of Computer Science and Engineering  
Southern Methodist University  
Dallas, TX 75275-0122  
raramada@engr.smu.edu

Khaled Abdelghany  
Department of Environmental and Civil Engineering  
Southern Methodist University  
Dallas, TX 75205-0335  
khaled@engr.smu.edu

Hesham El-Rewini  
Department of Computer Science and Engineering  
Southern Methodist University  
Dallas, TX 75275-0122  
rewini@engr.smu.edu

Abstract

The problem of optimally deploying a heterogeneous set of sensing devices in environments with differential surveillance requirements is presented. The problem is formulated in the form of integer mathematical program with an objective function that maximizes overall system coverage. The formulation considers several operation capabilities for the sensing devices including reliability, mobility, transfer cost, lifespan and power self-scheduling. Due to the intractability of the problem, optimal solutions obtained using our mathematical programming formulation can be obtained for small problems. In addition to the optimal solution, two heuristic algorithms are presented to solve different versions of the problem. A set of experiments is conducted to test the algorithms performances for different problem settings. Results of these experiments illustrate that near-optimal coverage performance could be achieved in very less running time.

1. Introduction

Considerable progress in automated surveillance and sensing technologies has been observed over the last two decades. Advances in technology have resulted in sensing devices that are wireless, communicating, mobile, and power self-scheduling. These advances in the surveillance technologies have been widely adopted in many applications including defense, environment protection, homeland security, healthcare, etc. For instance, sensors have been used to gather intelligence in battlefields. They have been also used to secure critical infrastructures such as power plants, water treatment facilities, airport and transit terminals, etc. Moreover, the revolutionary use of real-time data to better manage and control many infrastructure systems (e.g., highway networks) has added more importance to automated surveillance technologies. Nevertheless, methodologies used to optimally deploy surveillance devices have not advanced at an equivalent pace. In most cases, surveillance architects depend on ad-hoc deployment rules that lack accurate representation of the capabilities of these sensors. In addition, most surveillance operations utilize heterogeneous set of sensors that might differ in their coverage and communication ranges, reliability, trust-worthiness, lifespan, energy consumption and mobility. Integrating the capabilities of these heterogeneous sensors in one efficient deployment scheme is beyond the capability of most existing deployment methodologies, which usually assume the deployment of homogenous set of sensors.

In this paper, we study the problem of deploying sensing devices in a field with differential surveillance requirements. The objective is to maximize the coverage of this field using a limited set of heterogeneous sensing devices. The field is divided into equally sized zones. Each zone is defined through its location along with a time-varying function representing surveillance requirements (observation weights) for this zone over the monitoring period. A heterogeneous set of sensing devices is assumed. These sensors vary in their operational characteristics including reliability, lifespan, mobility, movement cost, and power self-scheduling capabilities. An integer mathematical program is formulated for this problem. The objective function of this program is designed such that sensing devices with high reliability are assigned to zones with high surveillance requirements. In addition to our optimal formulation, two heuristic algorithms are proposed for this problem. The first algorithm, which explicitly considers the transfer cost of mobile sensors, decomposes the problem into two phases. In the first phase, a set of deployment patterns for the sensors is generated. These patterns combine high-weight observations that require least number of sensor transfers between zones. The generated patterns are then evaluated against the actual capabilities of the available sensors. The second algorithm is more suitable for cases where sensor transfer cost is
relatively fixed. It solves the problem following a greedy approach that considers potential degradation in the sensors reliability. Observations over the entire horizon are first ranked based on their importance. Sensors are also ranked based on their reliability at each observation interval. Given the timestamp of each observation, it is assigned to the sensor with the highest reliability at this time interval.

Early contribution to the problem of surveillance deployment is due to Chvatal [5] who introduced the Art Gallery problem. In this problem, the goal is to determine the minimum number of observers required to secure an art gallery with a non-uniform geometry. Different versions of this problem have been studied to include mobile guard and guards with limited visibility (e.g., O’Rourke [10]). Research in the area of surveillance devices deployment has rapidly advanced with the emergence of wireless sensors networks. Most of research work in this area has concentrated on studying the optimal formation of a wireless sensing network that can be used to collect data from a given field and to transmit this data to one or more sink points [4,6,9]. The problem is studied considering different assumptions regarding configuration of the monitored field and characteristics of the used sensors. For instance, the problem of providing differentiated surveillance service in a field using homogenous device set is studied by Bhatnagar et al. [3] and Yan et al. [12]. Advanced device capabilities such as mobility and power self-scheduling are considered in al. [1,2,8]. In addition, Howard in [7] studied deploying surveillance devices that may operate cooperatively through sharing information and/or surveillance tasks.

Our modeling framework for the deployment problem presented in this paper extends existing models in three main aspects: a) it considers the deployment of heterogenous set of sensing devices; b) advanced device capabilities such as mobility and power self-scheduling are explicitly considered; and finally c) surveillance requirements in the monitoring field are assumed to vary over time and space. The rest of this paper is organized as follows. In Section 2, we introduce the problem definition. Section 3 presents our mathematical programming formulation, which produces the optimal solution. Two heuristic algorithms are presented in Section 4. We describe our experiments and display the results in Section 5. Finally we provide our concluding remarks in Section 6.

2. Problem Definition

Given is a field $F(A)$ which is divided into a set of zones $A$. This field is monitored for a horizon of length $T$ using a set of sensing devices $S$. Each zone $i \in A$ is associated with a time-varying weight function $w_{it}$, where $t \in T$. This weight function defines the importance of the observations (surveillance requirement) in this zone over the period $T$. Each sensor $s \in S$ is characterized by a predefined reliability $R_{it}$ that typically changes over time. In addition, a sensing device $s$ is also assumed to have a predefined lifespan (battery life) $L_s$. Also, each sensing device is assumed to have limited self-scheduling (state-switching) capability $P_s$. Thus, during the monitoring period $T$, a sensing device could be active (on) or inactive (off). A sensing device is expected to be active on link $i$ in time interval $t$ if the observation on this zone during this time interval is of high value ($w_{it}$ is relatively high). If $w_{it}$ is relatively low, this sensing device could be turned off to save its power. Thus, it could be used in other time intervals with high observation weights or in other more important zones. The maximum number of allowed state switching $P_s$ is known for each device $s$. We define the variable $On^t_{si} = 1$ if device $s$ is turned to active state by the end of time interval $t$ on zone $i$, and $On^t_{si} = 0$ otherwise. Similarly, we define the variable $Off^t_{si} = 1$ if device $s$ is turned to inactive state by the end of time interval $t$ on zone $i$, and $Off^t_{si} = 0$ otherwise.

In addition, a sensing device could be stationary or mobile. If a stationary device is deployed on a zone, this device is assumed to remain in this zone for the entire lifespan of the device. On the contrary, a mobile device can cover multiple zones over the period $T$. All mobile devices are assumed to have no restriction on the start or the end locations of their deployment. We define the binary variable $m^t_{sij} = 1$, if device $s$ is transferred from zone $i$ to zone $j$ by the end of time interval $t$, and $m^t_{sij} = 0$ otherwise. Each mobile device is assumed to have a maximum number of allowed moves $M_s$ ($M_s = 0$ for static devices) during the period $T$. A device transfer between two zones is assumed to be associated with a cost. This cost is expressed in terms of the loss in the device’s lifespan $E^t_{sij}$.

Given is a limited set of heterogeneous sensing devices in terms of $R_{it}, L_s, P_s$, and $E^t_{sij}$, the objective is to determine their optimal deployment schemes such that the field coverage is maximized. Coverage is maximized when observations with the highest importance are collected. Also, most reliable sensing devices are assigned to observations with the highest weights. For this purpose,
we define the decision variable $x_{si}^t$, which is equal to one if sensing device $s$ is deployed in active state in zone $i$ during time interval $t$, and zero otherwise. We also define the binary variable $y_{si}^t$, to trace the location of device $s$ while being inactive. If device $s$ is inactive in zone $i$ in time interval $t$, the variable $y_{si}^t$ is set to one, and zero otherwise.

### 3. Mathematical Formulation

The problem is formulated in the form of an integer mathematical program. The objective function and list of constraints developed for this program are as follows.

#### Program A:

Maximize:  
\[ \sum_t \sum_i \sum_s w_{si} x_{si}^t R_{si} \]  

Subject to:

**Deployment Constraints**

\[ x_{si}^t + y_{si}^t \leq 1 \quad \forall t, i, s \]  
\[ y_{si}^{t+1} \geq x_{si}^t - \sum_j x_{sj}^{t+1} \quad \forall t, i, s \]  
\[ y_{si}^{t-1} \geq x_{si}^t - \sum_j x_{sj}^{t-1} \quad \forall t, i, s \]  
\[ y_{si}^{t+1} \geq y_{si}^t - \sum_j x_{sj}^{t+1} \quad \forall t, i, s \]  
\[ y_{si}^{t-1} \geq y_{si}^t - \sum_j x_{sj}^{t-1} \quad \forall t, i, s \]

**Assignment Constraints**

\[ \sum_i (x_{si}^t + y_{si}^t) \leq 1 \quad \forall t, s \]  
\[ \sum_s x_{si}^t \leq 1 \quad \forall t, i \]

**Mobility Constraints**

\[ m_{sij}^t \geq \left( x_{sj}^{t+1} + y_{sj}^{t+1} \right) + \left( x_{si}^t + y_{si}^t \right) - 1 \quad \forall t, i, j, i \neq j, s \]  
\[ m_{sij}^t \leq x_{sj}^{t+1} + y_{sj}^{t+1} \quad \forall t, i, j, s \]  
\[ m_{sij}^t \leq x_{si}^t + y_{si}^t \quad \forall t, i, j, s \]  
\[ \sum_t \sum_i \sum_s m_{sij}^t \leq M_s \quad \forall s \]

**State Switching Constraints**

\[ On_{si}^t \geq (x_{si}^{t+1} + y_{si}^t) - 1 \quad \forall t, i, s \]  
\[ On_{si}^t \leq x_{si}^{t+1} \quad \forall t, i, s \]  
\[ Off_{si}^t \leq y_{si}^t \quad \forall t, i, s \]  
\[ Off_{si}^t \geq (y_{si}^{t+1} + x_{si}^t) - 1 \quad \forall t, i, s \]  
\[ Off_{si}^t \leq y_{si}^{t+1} \quad \forall t, i, s \]  
\[ Off_{si}^t \leq x_{si}^t \quad \forall t, i, s \]  
\[ \sum_t \sum_i \left( On_{si}^t + Off_{si}^t \right) \leq P_s \quad \forall s \]

**Lifespan Constraints**

\[ \sum_t \sum_i \sum_j \sum_s E_{sij}^t m_{sij}^t \leq L_s \quad \forall s \]

**Binary Constraints**

\[ \{ x_{si}^t, y_{si}^t, m_{sij}^t, On_{si}^t, Off_{si}^t \} = 1 \text{ or } 0 \quad \forall t, i, s \]

As shown in (1), the objective function maximizes the field coverage which is described as the sum over all time intervals, zones and sensing devices of the products of the observation weight $w_{si}$, the decision variable $x_{si}^t$ ($x_{si}^t = 1$, if device $s$ exists in active state on zone $i$ in time interval $t$) and the reliability of the used device $R_{si}$.

Constraints in (2) ensure that a sensing device is either active or inactive during any time interval. Constraints in (3) to (6) determine the value of the binary variable $y_{si}^t$ based on the value of $x_{si}^t$. If a sensing device is set to be active in zone $i$ during time interval $t$, and this device is not used in any zone during the next (previous) time interval, this device is assumed to be inactive and to remain in this zone during the next (previous) time interval. Similarly, if a device is set to be inactive in zone $i$ during time interval $t$, and this device is not used in any zone during the next (previous) time interval, this device is assumed to remain inactive in the same zone during the next (previous) time interval. Constraints in (7) ensure that each zone is covered by at most one device in any time interval. Also, at each time interval, a sensing device is covering at most one zone, which is guaranteed in constraints (8).

Constraints in (9) to (11) determine if sensing device $s$ is moved from zone $i$ to zone $j$ at the end of interval $t$. They compare the zones where sensing device $s$ is deployed during time intervals $t$ and $t+1$. The binary variable $m_{sij}^t$ is set to one if they are different. Constraints in (12) ensures that number of moves made by a device is less than or equal to maximum number of moves allowed for this device.

The state-switching of a sensing device from active state to inactive state and vice versa are determined in
constraints \((13)\) to \((18)\). The binary variables \(x^t_{si}\) and \(y^t_{si}\) are examined for each sensing device while being deployed in every zone. If both variables are equal to one in two successive time intervals, it indicates that the device’s state is altered. The total number of state switching for each sensing device is computed and compared to the maximum number of switching allowed for each device as given in constraints \((19)\). Constraints in \((20)\) ensure that each sensing device is not utilized beyond its lifespan. The consumption of a device’s lifespan is computed as the sum over all intervals in which the device is active plus the loss in the device lifespan associated with its moves along the different zones. Finally, the integrality of all binary variables is preserved in constraints \((21)\).

4. Heuristic-Based Algorithms

4.1 Pattern-Based Algorithm

The pattern-based approach consists of two main steps. In the first step, a set of deployment patterns is generated. The number of these patterns is equal to the number of available sensors. These patterns include zones with the highest surveillance requirements (observations with the highest weights) over the monitored horizon. These zones are clustered in the different patterns such that the sensors’ total transfer cost is minimized. The generated patterns are not overlapping. In other words, two patterns cannot include the same observation. While generating these patterns, sensors are assumed to have unlimited lifespan and unlimited number of moves. Figure 1 illustrates a greedy algorithm that is used to generate these patterns. The highest \(k\) observations at each time interval are picked and sorted in a decreasing order, where \(k\) is the number of patterns to be generated. The movement cost associated with adding each of these \(k\) observations in each pattern is determined. The movement cost is the cost associated with moving the device from the zone where the device is deployed in the previous time interval to the zone where the device is deployed in the current time interval. The \(k\) observations are then added to the \(k\) patterns such that total transfer cost over all patterns is minimized. The process continues until all patterns are filled with \(T\) observation, where \(T\) is the monitored horizon. Sorting the observations takes \(O(T |A| \log |A|)\) operations, where \(|A|\) is the number of zones. Appending the observations in the patterns while minimizing the movement cost requires \(O(Tk^2)\) operations. Thus, the overall complexity of step 1 is \(O(Tk^2 + |A| \log |A|)\).

### Algorithm 1: Patterns-Generation

\[
\text{for } t=1 \text{ to } T \text{ do} \\
\text{List } = \text{Sort (observations at } t) \\
\text{for } l=1 \text{ to } k \\
\quad \text{observation} = \text{List}[l] \\
\quad \text{pattern} = \text{getMinimumCostPattern (observation, patterns)} \\
\quad \text{AppendObservationToPattern (observation, pattern)} \\
\text{end do} \\
\text{end for}
\]

### Figure 1: Algorithm for patterns generation

Once these patterns are generated, the next step is to determine the maximum possible coverage performance that can be achieved by each sensor. Sensors have limited lifespan, limited number of moves and limited number of state switches. Thus, patterns generated in the previous step have to be modified to consider the limited capability of the assigned sensors. Some of the observations might be removed from a pattern to ensure feasibility. Although this problem is NP-hard in its general form, a pseudo-polynomial algorithm can be used to solve versions of this problem with limited horizon. Figure 2 illustrates the algorithm used to determine a near-optimal pattern for each sensor. Given a pattern \(p\), observations in this pattern are sorted based on the product of their weights and the sensor reliability in the corresponding time intervals. Observations in this sorted list are sequentially appended to the pattern while ensuring that each added observation is not violating the capability of the sensor. In other words, an observation is added to the pattern only if the sensor’s lifespan, maximum allowed number of moves, and maximum allowed number of state switches are not violated. Sorting the pattern observations requires \(O(T|log|T)\) operations, where \(T\) is monitored horizon. The feasibility checking requires \(O(T)\). Therefore, the worst case complexity of this algorithm is \(O(T|log|T)\) operations per pattern.

### Algorithm 2: Sensor Maximum Performance

\[
\text{List } = \text{Sort (} \bar{W}^t_i, R^t_i \text{)} \\
\text{for } k=1 \text{ to } T \\
\text{observation} = \text{List}[l] \\
\text{checkFeasibility(} \text{observation} \text{)} \\
\quad \text{if (feasible) addToPattern(} \text{observation} \text{)} \\
\text{end for}
\]

### Figure 2: Sensor maximum performance algorithm
4.2 Observation-Based Algorithm

The observation-based algorithm is more suitable for cases where the sensors transfer cost among the different zones is relatively fixed. Figure 3 illustrates the steps of this algorithm. Observations are sorted in descending order based on their weights. Given the timestamp associated with each observation, all sensors are scanned to find the sensor with the highest reliability in this time interval. The feasibility of this observation to join the pattern of this sensor is then checked. If the observation is not feasible, the next reliable sensor is selected and so on until a sensor is found for this observation. If no sensor is found, the observation is discarded and the next highest observation in the list is selected. The process continues until no more observations can be added to any of the available sensors.

As described earlier, an observation is feasible only if it does not violate the sensor’s lifespan, maximum number of moves and maximum number of state switches. Sorting the sensors reliabilities over the horizon $T$ requires $O(T|S| \log |S|)$ operations, where $|S|$ is the number of sensors. Also, sorting the observations over $T$ requires $O(T|A| \log T|A|)$ operations. In addition, the feasibility checking takes $O(T|A||S|)$ operations. Thus, the worst case complexity for this heuristic is $O(T|A||S|)$.

**Algorithm 3: Observation-Based Algorithm**

```plaintext
List = Sort (W')
for l=1 to List.size
    observation= List[l]
    timestamp = observation.timestamp
    while(observation is not assigned) do
        sensor=findMostReliableSensor(timestamp)
        counter++
        checkFeasibility(observation, sensor)
        if(feasible)
            assignObservationToSensor(observation, sensor)
            assigned = true
        if(not feasible && counter == |S|)
            assigned = true
    end do
end for
```

**Figure 3: Observation based algorithm**

5. Experiments and Results

5.1 Experiment I: Optimal Solution Limits

In this experiment, the global optimal solution is obtained using CPLEX-8. We specifically study the effect of changing the problem size (number of zones, number of sensors, and the size of the horizon) on the time it takes to obtain an optimal solution. It is important to find out the size of the largest problem we can solve optimally in a reasonable amount of time. This information can be used as a base line to evaluate the performance of the proposed heuristics, as we will see in the following experiments.

Three different sets of experiments are conducted as shown in Table 1. In all experiments, the time-varying observations on the different zones were generated randomly following a uniform distribution $U(0,200)$. In addition, a heterogeneous set of sensors is assumed. The sensors’ operational characteristics $L_s$, $M_s$, and $P_s$ are generated randomly as function of the length of monitoring period. For example, if the monitoring horizon is $T$ intervals, the sensor lifespan, maximum allowed number of moves, and maximum allowed number of state switching are generated randomly using the uniform distribution $U(1, T)$. Sensors reliability is assumed to be fixed with time and no lifespan loss associated with their moves.

As expected and as shown in Table 1, the algorithm running time increases with the increase in number of zones, number of sensing sensors and length of the monitoring period. For example, in the first set of experiments, increasing the number of zones from 10 to 30 results in a jump in the solution running time from 1542.36 seconds to 59031.1 seconds. The optimal solution fails to solve a field with 35 zones in reasonable time. Similarly, when the number of sensing sensors is increased from 3 to 10, an increase in the running time from 1500 to 1628723 seconds is recorded. The optimal solution could not be obtained in reasonable time when 15 sensors were used on the same field. Also, increasing the monitoring horizon from 3 to 12 intervals results in an increase in the running time from 100 to 18012.4 seconds. In the following sections, these sets of experiments are used as a benchmark to evaluate the performance of the proposed heuristics.

5.2 Experiment II: Pattern-Based Performance

5.2.1 Effect of the problem Size. In this section, the performance of the pattern-based (PB) algorithm is compared with the global optimal solution. The coverage performance and running time obtained using the PB algorithm for different problem settings are presented as percentage of the corresponding values obtained by the global optimal solution. In general, the pattern-based algorithm is able to achieve a reasonable coverage performance. As shown in Table 2, the coverage performance ranges form 68% (experiment 4) to 86% (experiment 8). In addition, in all cases, the PB algorithm produces the solution in very less running time compared to the global optimal solution. One can also notice that the difference in the running time between the pattern-base
based algorithm and the global optimal tends to increase in a non-linear order with the increase in the problem size. In experiment 6 where 3 sensors are used, the running time of the PB algorithm is recorded to be 20% of the running time of global optimal solution. When the number of sensors is increased to 15, the algorithm’s running time is recorded as 0.006% of the time required to find the global optimal solution.

5.2.2 Effect of the movement cost. In this section, a set of experiments is conducted to measure the sensitivity of the PB algorithm to the variance in the devices movement cost. The movement cost is randomly generated within different ranges that follow a uniform distribution \( U(1, T) \) where \( T \) is equal to 25%, 50%, 75%, or 100% of the given horizon \( T \). The results show that as the variance in the movement cost increases, the heuristic performance improves. For instance, at 1-25 range, the PB algorithm produces 80% of the global optimal. This percentage jumps to 80% when in the range of the movement cost is increased to 1-100. Thus, providing a set of sensing devices that vary in their transfer cost improves overall coverage performance.

\[ \text{Performance} \quad \begin{array}{cccc}
1-25\% & 1-50\% & 1-75\% & 1-100\%
\end{array} \]

\[ \begin{array}{c}
60\% \\
72\% \\
74\% \\
80\%
\end{array} \]

Figure 4: Effect of movement cost variance on coverage performance.

5.3 Experiment III: Observation-Based Performance

5.3.1 Effect of the problem size. Table 3 presents a comparison between the global optimal solution and the observation-based (OB) heuristic with different number of zones, number of sensors, and horizon lengths. A main observation is that the OB algorithm is able to produce much better solutions compared to the PB algorithm. In one case (experiment 6), the algorithm was able to find the global optimal deployment pattern. Similar to the PB algorithm case, the gap between the running time to determine the global optimal solution and the running time of the heuristic is increasing with the increase in the problem size. For instance, when 10 zones are used, the heuristic elapsed time is 0.002% of the corresponding global optimal running time. As the number of zones increased to 20, the elapsed time was 0.0003% of the optimal. Similarly, when the number of sensors is increased to 10, the heuristic requires only 0.000014 % of the global optimal running time.

5.3.2 Effect of Observations Variance on the Performance. In this section, another set of experiments is conducted to show the effect of variance of the observation weight on the performance. Figure 5 presents the coverage performance obtained using the OB algorithm as a percentage of the optimal solution for a field with different observation weight variances. A filed of 6 zones, 3 sensing devices that is monitored for 12 time intervals is used. The observation weights along the different number of zones are generated randomly using uniform distributions with different ranges. Increasing the distributions range produces more variance in the observation weights along the different zones and over the monitoring horizon. In this set of experiments, uniform distribution with a range of 200 represents the case with the highest variance. As the variance in the observation weights across the zones increases, the algorithm tends to produce solutions that are closer to the optimal ones. For example, a coverage performance of 93% is obtained when the zones’ observation weights are set according to uniform distribution with a range of 150. This performance is improved to 94% when the range is increased to 175.

\[ \text{Performance} \quad \begin{array}{cc}
0-200 & 25-175 \\
95\% & 94\%
\end{array} \]

\[ \begin{array}{c}
90\% \\
87\%
\end{array} \]

Figure 5: Effect of Observations Variance on Coverage Performance.

6. Conclusions

In this paper, the problem of deploying a heterogeneous set of sensing devices to maximize coverage of critical environments with differential surveillance requirements is presented. We implemented a formulation of the problem using mathematical programming. This implementation was used to find optimal solutions for small size problems. Two heuristics were presented to provide near optimal solutions to the problem. A number of experiments were conducted to evaluate the solutions obtained using heuristic
algorithms compared to the optimal solution. The results show that the pattern-based heuristic’s average performance is better than 70% of the global optimal performance. This performance improves noticeably as the number of links, number of sensors, and the horizon increases. Also, as the range of movement cost (variance) increases, the results show that the heuristic performance improves. The observation-based heuristic was tested through number of experiments. These experiments show that the performance is no less than 90% with different problem settings. Additionally, a number of experiments were conducted to show how these heuristics can be used to help system designers answer a variety of what-if questions and to capture the trade-offs between the sensors’ attributes.

### Table 1: Global optimal solution for different problem settings

<table>
<thead>
<tr>
<th>Exp No.</th>
<th>Number of Zones</th>
<th>Number of sensors</th>
<th>Horizon</th>
<th>Objective</th>
<th>Running Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>5</td>
<td>12</td>
<td>950</td>
<td>1542.4</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>5</td>
<td>12</td>
<td>991</td>
<td>18012.4</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>5</td>
<td>12</td>
<td>969</td>
<td>29057.9</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>5</td>
<td>12</td>
<td>990</td>
<td>59031.1</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>5</td>
<td>12</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>3</td>
<td>12</td>
<td>482</td>
<td>1500.1</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>5</td>
<td>12</td>
<td>991</td>
<td>18012.5</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>10</td>
<td>12</td>
<td>2100</td>
<td>1628723</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>15</td>
<td>12</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>680</td>
<td>100.2</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>5</td>
<td>6</td>
<td>887</td>
<td>1010.3</td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>15</td>
<td>12</td>
<td>991</td>
<td>18012.4</td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>5</td>
<td>24</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

### Table 2: Performance of the pattern-based algorithm

<table>
<thead>
<tr>
<th>Exp No.</th>
<th>Number of Zones</th>
<th>Number of sensors</th>
<th>Horizon</th>
<th>Objective (%)</th>
<th>Running Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>5</td>
<td>12</td>
<td>69</td>
<td>0.10</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>5</td>
<td>12</td>
<td>69</td>
<td>0.003</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>5</td>
<td>12</td>
<td>68</td>
<td>0.004</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>5</td>
<td>12</td>
<td>70</td>
<td>0.004</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>5</td>
<td>12</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>3</td>
<td>12</td>
<td>76</td>
<td>0.20</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>5</td>
<td>12</td>
<td>69</td>
<td>0.003</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>10</td>
<td>12</td>
<td>86</td>
<td>0.0006</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>15</td>
<td>12</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>70</td>
<td>0.01</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>5</td>
<td>6</td>
<td>70</td>
<td>0.003</td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>5</td>
<td>12</td>
<td>69</td>
<td>0.003</td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>5</td>
<td>24</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
Table 3: Performance of the observation-based algorithm

<table>
<thead>
<tr>
<th>Exp. No.</th>
<th>Number of Zones</th>
<th>Number of sensors</th>
<th>Horizon</th>
<th>Objective (%)</th>
<th>Running Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>5</td>
<td>12</td>
<td>91</td>
<td>.002</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>5</td>
<td>12</td>
<td>94</td>
<td>.0003</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>5</td>
<td>12</td>
<td>94.5</td>
<td>.0004</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>5</td>
<td>12</td>
<td>94.6</td>
<td>.0002</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>5</td>
<td>12</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>3</td>
<td>12</td>
<td>100</td>
<td>.002</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>5</td>
<td>12</td>
<td>94</td>
<td>.0003</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>10</td>
<td>12</td>
<td>98</td>
<td>.0000014</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>15</td>
<td>12</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>99</td>
<td>.006</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>5</td>
<td>6</td>
<td>99</td>
<td>.001</td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>5</td>
<td>12</td>
<td>94</td>
<td>.0003</td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>5</td>
<td>24</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

7. References


