SensDep: A Design Tool for the 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

This paper is an extension to our recent work in which we presented a heterogeneous sensing devices deployment problem on environment with differential surveillance requirements. In this paper, we introduce *SensDep* as software design tool that incorporates several solution strategies to optimize sensor networks cost and coverage. The tool helps the designer answer many "what-if" questions that usually arise in the design of surveillance operations networks. It helps the designer set parameters, experiment with alternative designs and component properties, and see the relevant results. Also, it considers several operation capabilities for the sensing devices including reliability, mobility, transfer cost, sensors' cost, lifespan and power self-scheduling in addition to the environment parameters during the deployment process. Moreover, it provides the designer the optimal deployment scheme for small size design using mathematical programming. It also provides near optimal schemes for large scale designs using a set of heuristic solutions. A set of experiments is conducted to test the tool capabilities for different design settings. Several design scenarios are presented to illustrate how the tool can be utilized.

1. Introduction

Advances in wireless sensing technologies have significantly broadened their applications including defense, environment protection, homeland security, infrastructure management, and healthcare. In most of these applications, the goal is to maximize coverage in a given environment using limited surveillance devices. Given the wide range of existing sensing technologies, sensors used in one surveillance operation could vary in their cost, reliability, lifespan, energy consumption, mobility and power self-scheduling capabilities. They may also differ in their coverage and communication ranges. Integrating the capabilities of these heterogeneous sensors in one deployment scheme complicates the planning of most surveillance operations, especially if these sensors were to be deployed in dynamic environments with differential surveillance requirements.

Early contribution to the problem of surveillance deployment was reported in [6], as a solution to 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., [12]). In general, 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, 5,10]. 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 in [3] and [21]. Advanced device capabilities such as mobility and power self-scheduling are considered in [1,2,8]. In addition, Howard studied deploying surveillance devices that may operate cooperatively through sharing information and/or surveillance tasks [7].

In this paper, we present *SensDep*: a decision support tool for the design of large-scale automated surveillance operations. The tool considers the deployment of heterogeneous set of sensing devices with advanced capabilities such as mobility and power self-scheduling. Furthermore, monitored fields with dynamic differential surveillance requirements are considered. This tool is expected to help surveillance architects answer a variety of “what-if” questions that usually arise in the design of large-scale surveillance operations.

The tool incorporates multiple algorithms that are capable of generating near-optimal deployment schemes for problems with special structures in short running time. These algorithms are based on a modeling framework in which the monitoring filed is divided into a gird of cells (zones). Each zone is defined through its location along with a time-varying function representing surveillance requirements. A heterogeneous set of sensing devices is assumed. These sensors could vary in their operational characteristics including reliability, lifespan, mobility, movement cost, and power self-scheduling capabilities. The solution determines the optimal deployment pattern for each sensor. A deployment pattern for one sensor is described in terms of the zone to be covered in each time interval of the monitoring horizon.

The rest of this paper is organized as follows. The problem definition is given in section 2. The tool and its algorithms are described in section 3. Performance evaluation and experiment results are presented in sections 4 and 5. Finally, we give our conclusion in section 6.

2. Deployment Problem

Given is a field *A* to be monitored for a horizon of length *T*. The field is divided into small cells (zones) $i \in A$. The surveillance requirements for each zone are defined through a time-varying weight function w_i^t, where $t \in T$. This weight function defines the importance of the observations in this zone over the period T. Also, given is a set of sensors *S*. These sensors differ in their operational characteristics as well as in their cost C_s. Each sensor $s \in S$ is described by its reliability R_s^t, which might vary with time. In addition, a predefined lifespan L_s is attributed for each sensor. The lifespan of a sensor $s \in S$ is mainly based on its battery lifetime. Also, sensors could be stationary or mobile. Stationary sensors remain in the same zone from the time they are deployed to the end of the horizon *T*. On the contrary, mobile sensors are capable of moving among different zones to cover high weight observations in the different zones. Each move is associated with a transfer cost E_{sij}^t. All mobile sensors are assumed to have no restrictions on the start or the end locations of their deployment scheme.

Sensors are assumed to have self-scheduling (state-switching) capabilities. Sensors could be active "*on*" to monitor weight observation w_i^t in zone *i* in interval *t* or inactive "*off*" otherwise (e.g., no observations). Thus, it saves the sensor's lifespan to be used during other intervals with high observation weights. A maximum number of state-switching P_s is defined for each sensor. In this version of *SensDep*, all sensors are assumed to have unlimited communication range and to cover exactly one zone at any given time. Given a limited set of heterogeneous sensing devices which are described in terms of R_s^t, L_s, P_s, E_{sij}^t, and C_s, the problem is to determine an efficient deployment scheme that integrates the capabilities of these sensors. The best deployment scheme maximizes the field coverage and minimize the overall sensors cost.

3. SensDep Description

As shown in Figure 1, the input to *SensDep* is a script which is supplied through the script editor component. The script is translated by the script translator component into a set of commands that are sequentially executed. These commands include calling the selected algorithms, reading the suitable input data, setting the required parameters, generating and displaying the solution. The solution module implements several algorithms that produce optimal and near optimal solutions. The optimal solution is provided by an integer mathematical program which is implemented using CPLEX-80. For problems with special structures, two heuristic algorithms are provided: pattern-based algorithm and observation-based algorithm. These algorithms generate near optimal solution to large-size problems in very short running time compared to the time required to generate an exact optimal solution. In the following subsections, we describe the main properties of each of these sub-modules.

3.1 Optimal Solution

In this section, the optimal solution for the sensor deployment problem is presented. The problem is formulated using integer mathematical programming. A set of binary variables are defined as follows:

$x_{si}^t = 1$, if device s exists in active state on zone i in time interval t, and 0 otherwise.

$y_{si}^t = 1$, if device s exists in inactive state on zone i in time interval t, and 0 otherwise.

$m_{sij}^t = 1$, if device s is moved from zone i to zone j at time interval t, and 0 otherwise.

$on_{si}^t = 1$, if device s is turned to active state by the end of time interval t on zone i, and 0 otherwise.

$off_{si}^{t} = 1$, if device s is deactivated by the end of time interval t on zone i, and 0 otherwise.

$V_s = 1$, if sensor s is used on one of the zones and 0 otherwise.

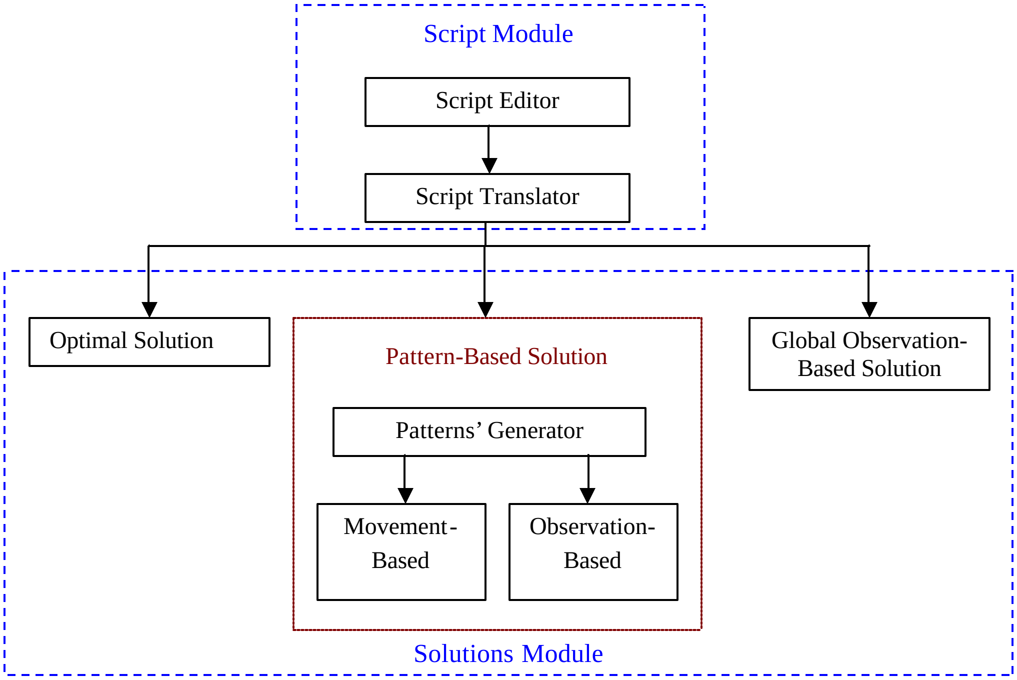

Figure 1: SensDep main modules

The solution framework allows solving the following problem settings:

A. Maximizing the field coverage consider the sensors' time-dependent reliability as given in *(1)*. In other words; the objective function is to cover zones with the highest observation weights with the most reliable sensors.

$$\textbf{Maximize}: \sum_t \sum_i \sum_s w_i^t . x_{si}^t . R_s^t \qquad (1)$$

$$\text{Where } x_{si}^t = \begin{cases} 1 & \text{If ssensing device s is deployed in active state} \\ & \text{on zone i during time interval t,} \\ 0 & \text{Otherwise} \end{cases}$$

B. Maximizing the coverage based on sensors' time-dependent reliability while limiting the overall sensors cost to a given budget *B*. The objective function will be the same as in *(1)*. However, constraint *(2)* needs to be added which limits the overall sensors cost to *B*.

$$\sum_{s} V_s.C_s \leq B \qquad (2)$$

Where $V_s = \begin{cases} 1 & \text{If sensing device s is used,} \\ 0 & \text{Otherwise} \end{cases}$ and

$$V_s \leq \sum_{i}\sum_{t} x_{si}^{t} \qquad \forall s \qquad (3)$$

C. Maximizing the coverage considering the sensors' time-dependent reliability and minimizing the sensors total cost as given in *(4)*. Using the weight parameters ?$_1$ and ?$_2$, the system cost is subtracted from the sum of all collected observations as illustrated in the objective function given in *(4)*.

$$\textbf{Maximize}: \ l_1.\sum_{t}\sum_{i}\sum_{s} w_i^t.x_{si}^t.R_s^t - l_2.\sum_{s} V_s.C_s \qquad (4)$$

The common set of constraints for these different problem settings could be described as follows (see our recent work in [13] for more details).

- Deployment constraints to relate x_{si}^t and y_{si}^t

$$x_{si}^t + y_{si}^t \leq 1 \qquad \forall t, i, s \qquad (5)$$

$$y_{si}^{t+1} \geq x_{si}^t - \sum_{j} x_{sj}^{t+1} \qquad \forall t, i, s \qquad (6)$$

$$y_{si}^{t-1} \geq x_{si}^t - \sum_{j} x_{js}^{t-1} \qquad \forall t, i, s \qquad (7)$$

$$y_{si}^{t+1} \geq y_{si}^t - \sum_{j} x_{js}^{t+1} \qquad \forall t, i, s \qquad (8)$$

$$y_{si}^{t-1} \geq y_{si}^t - \sum_{j} x_{js}^{t-1} \qquad \forall t, i, s \qquad (9)$$

Where $y_{si}^t = \begin{cases} 1 & \text{If sensing device s is deployed in inactive state} \\ & \text{on zone i during time interval t,} \\ 0 & \text{Otherwise} \end{cases}$

- Assignment constraints to ensure that each zone is covered by at most one sensing device in any time interval. Also, at each time interval, a sensing device is covering at most one zone.

$$\sum_{i}\left(x_{si}^t + y_{si}^t\right) \leq 1 \qquad \forall t, s \qquad (10)$$

$$\sum_{s} x_{si}^t \leq 1 \qquad \forall t, i \qquad (11)$$

- Mobility constraints which record sensors movement and limit number of moves to M_s.

$$m_{sij}^{t} \geq \left(\left(x_{sj}^{t+1} + y_{sj}^{t+1}\right) + \left(x_{si}^{t} + y_{si}^{t}\right)\right) - 1 \qquad \forall t, i, j, i \neq j, s \tag{12}$$

$$m_{sij}^{t} \leq x_{sj}^{t+1} + y_{sj}^{t+1} \qquad \forall t, i, j, s \tag{13}$$

$$m_{sij}^{t} \leq x_{si}^{t} + y_{si}^{t} \qquad \forall t, i, j, s \tag{14}$$

$$\sum_{i}\sum_{j}\sum_{t} m_{sij}^{t} \leq M_s \qquad \forall s \tag{15}$$

- State switching constraints which keep track of sensors switching and limit the number of switching to P_s.

$$on_{si}^{t} \geq \left(x_{si}^{t+1} + y_{si}^{t}\right) - 1 \qquad \forall t, i, s \tag{16}$$

$$on_{si}^{t} \leq x_{si}^{t+1} \qquad \forall t, i, s \tag{17}$$

$$on_{si}^{t} \leq y_{si}^{t} \qquad \forall t, i, s \tag{18}$$

$$off_{si}^{t} \geq \left(y_{si}^{t+1} + x_{si}^{t}\right) - 1 \qquad \forall t, i, s \tag{19}$$

$$off_{si}^{t} \leq y_{si}^{t+1} \qquad \forall t, i, s \tag{20}$$

$$off_{si}^{t} \leq x_{si}^{t} \qquad \forall t, i, s \tag{21}$$

$$\sum_{t}\sum_{i} \left(On_{si}^{t} + Off_{si}^{t}\right) \leq P_s \qquad \forall s \tag{22}$$

- Lifespan constraints to limit the number of times that the sensor is active to sensor's lifespan L_s. This used lifespan is computed using x_{si}^{t} while it is one and the mobility cost E_{sij}^{t} if the sensor is moved from zone i to zone j at time *t*.

$$\sum_{t}\sum_{i} x_{si}^{t} + \sum_{i}\sum_{j}\sum_{t} E_{sij}^{t}\, m_{sij}^{t} \leq L_s \qquad \forall s \tag{23}$$

- Binary constraints

$$\{x_{si}^{t}, y_{si}^{t}, m_{sij}^{t}, on_{si}^{t}, off_{si}^{t}\} = 1 \text{ or } 0 \qquad \forall t, i, s \tag{24}$$

Due to the intractability of the problem, the optimal solution based on the formulation is obtained only for small-size problems. For large-scale problems, the following heuristics are developed.

3.2 Pattern-Based Solution

The idea of the pattern-based algorithm is to decompose the problem in terms of the available set of sensing devices. Observations are clustered in patterns. Each pattern is described in terms of its zones and time intervals in which these zones are visited. These patterns are generated for sensing devices with unlimited lifespan, unconstrained mobility, unlimited state switching capabilities, and sensor's cost is assumed to be zero. Available sensing devices are then assigned to these patterns. An optimal matching problem is then solved to ensure that available devices are optimally utilized and the sensors' cost constraint is satisfied. Using this decomposition approach, a solution algorithm is developed. The algorithm consists of three main steps as follows.

Solution Algorithm

Step 1: Generation of deployment patterns

Step 2: Determining device-pattern performances

Step 3: Device-Pattern Matching

Step 1: Generation of deployment patterns

The first step generates the deployment patterns for a set of hypothetical devices S' with unlimited capabilities. These deployment patterns are generated such that they include observations with the highest weights in the entire horizon. Generated patterns are not overlapping in the sense that two patterns cannot include the same observation. In this step, k patterns are generated. The value of k is equal to the number of given sensors $|S|$ if $|S| \leq |A|$, where $|A|$ is the number of zones, otherwise $k=|A|$. Figure 2 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. At each time, the highest observation is assigned to a specific pattern. The process continues until all patterns are filled with T observations, where T is the monitored horizon. Sorting the observations takes $O(T\ |A|\ log\ |A|)$ operations. Appending the observations to the patterns requires $O(Tk)$ operations. Since k can not be more than the |A|, the overall complexity of this step is $O(T(|A|\ log\ |A|))$.

Algorithm 1: Patterns-Generation

```
for t=1 to T do
        List = Sort($W_i^t$)
          for l=1 to k  do
            H_Patterns [t][l] = List [l]
                    l++
         end  for
end for
```

Figure 2: Patterns Generation Algorithm

Step 2: Determining sensor-pattern performances.

In step 1, different patterns are generated for a set of unrestricted sensors. In other words, these sensors are assumed to have no restriction on their operation characteristics (R_s^t, L_s, M_s, P_s and E_{sij}^t). Once the actual sensors are applied to these patterns, they are expected to result in less performance. Determining the performance of each sensor when assigned to any of the generated patterns is done through two different greedy algorithms: (1) Movement-based and (2) Observation-based. Movement-based algorithm is suitable for problem structures where variations in the sensors transfer cost are large. On the other hand, the observation-based algorithm is suitable for problems where sensor's transfer costs are relatively fixed.

Movement-Based Greedy Algorithm:

In any of the generated patterns A', assume two observations $i \in A'$ and zone $j \in A'$ with weights $w_i^{t'}$ and $w_j^{t''}$, respectively, where $t', t'' \in T$ and $t'' > t'$. Also, assume that device $s \in S$ loses $e_{sij}^{t'}$ and $e_{sij}^{t''-1}$ lifespan time units to travel from link i to link j at intervals t' and t'', respectively. Furthermore, reliability levels of $R^{t'}$ and $R^{t''}$ are estimated for this device at times t' and t''. The gain $g_{sij}^{t'}$ associated with covering these two observations consecutively is computed as follows.

$$g_{sij}^{t'} = \frac{R^{t'} w_i^{t'} + R^{t''} w_i^{t''}}{2 + e_{sij}^{t'}} \qquad \textit{(25)}$$

As shown in equation *(25)*, the numerator gives the gained observation weights considering the device's variable reliability, while the denominator describes the device's lifespan consumed in performing this task. This lifespan includes two time intervals plus the time lost in the move assuming that this move is performed at the end of interval t'. These gain values are computed for each feasible move and then sorted in a descending order. Starting from the top of the list, a move is directly appended to the pattern under construction. The resulting pattern is then checked for feasibility against the sensor capabilities (lifespan, number of moves and number of switches). If feasible, the new pattern is accepted. Otherwise, the algorithm proceeds to the next element in the list of gain values until all elements are scanned or until one of the device's limitations is reached. As shown in Figure 3, the sorting of the gain values list could be conducted in $O(T^2 \log T^2)$. This list is scanned T times to check the feasibility of each observation addition to the pattern. Thus, Step 2 has a worst case complexity of $O(T^3 \log T^2)$.

```
Algorithm 2: Movement-Based Greedy
          Table = GeneratMovementTable(K_List)
          S_table = Sort (table)
           for l=1 to S_Table.size
            Feasible = CheckFeasibility(S_Table[l])
              if  (Feasible)
              Current_Pattern+= S_Sorted[l]
             end if
             if (sensor is saturated)
              Stop
            end if
         end for
```

Figure 3: Movement-based greedy algorithm

Observation-Based Greedy Algorithm:

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. Figure 4 illustrates the observation-based algorithm used to determine near optimal pattern for each sensor. Sorting the pattern observations requires *O(TlogT)* operations, where *T* is monitored horizon. The feasibility checking requires *O(T)*. Therefore, the worst case complexity of this algorithm is *O(TlogT)* operations per pattern.

```
Algorithm 3: Observation-Based Greedy
      List = Sort ( $\overline{w}_i^t$ , $R^t$ )
              for l=1 to T
      observation= List[l]
      checkFeasibility(observation)
               if (feasible) addToPattern(observation)
      end for
```

Figure 4: Observation-Based Greedy Algorithm

Step 3: Device-Pattern Matching

Step 2 gives the total observation weights O_{sp} that could be collected by assigning sensing device $s \in S$ to deployment pattern $p \in K$ considering the limited capability of this device. Now, assume that this value is determined for all device-pattern combinations. The problem is to find the optimal match between available devices and the deployment patterns such that: a) available devices are optimally utilized, b) the total cost is minimized and c) the given budget B is not violated. A similarity between this optimal matching problem and 0/1 knapsack problem [14] can be made. 0/1 knapsack problem is informally defined as; we are given a set of n items from which we are to select some number of items to be carried in a knapsack. Each item has both a *weight* and a *profit*. The objective is to choose the set of items that minimize the weights, maximize the profit and fits the knapsack limit. This problem is extensively investigated and already has many solutions; see [9,15,20]. Our solution to the optimal matching problems is inspired by the 0/1 knapsack solutions.

Given a list of four tuples *<k, s, o, c>;* where *k* is a pattern, *s* is a sensor, and *o* is the objective resulted from using sensor *s* with pattern *k*, and *c* is the sensor's cost. Objective cost ratio R_{oc} is computed by dividing the resulted objective o and the sensors cost c. This list is sorted in decreasing order based on the R_{oc}. The top element in the list is selected and added to the final assignment list named *F_list*. Then, every entry contains the selected pattern $k \in K$ and sensor $s \in S$ is deleted from the sorted list. This is repeated until there is no element in the list or the given budget *B* is reached. As shown in Figure 5, sorting the list requires $O(K|S| \ log \ K|S|)$ and scanning it requires $O(K|S|)$. Thus, the overall complexity is $O(K|S|)$.

```
Algorithm 4: Device-Pattern Matching
        List = Sort (SPR_oc _List)
        while (List is not empty or the budget B is not violated)
          F_List = Select(List)
           DeleteEntries (selected sensor and/or pattern)
        end while
```

Figure 5: Device-pattern matching

3.3 Global Observation-Based

Global observation-based algorithm is more suitable for cases where sensors transfer cost among the different zones is relatively fixed. Following this algorithm, observations are sorted in descending order based on their weights. In addition, sensors are sorted in descending order based on their capabilities. Equation (26) illustrates an example of how each sensor is evaluated in terms of the sensor's lifespan l_s, number of switching p_s, and number of moves m_s, respectively. We use g_1, g_2, and g_3 to weigh the importance of the different operational parameters for this sensor. For instance, TRAMA [17], T-MAC [16], and S-MAC [18] routing protocols require more switching capabilities to reduce sensors' power consumption. Thus, it gives higher value to g_2. Also, SPIN [11] might need sensors to be *"on"* all the time. Therefore, g_1 will be the factor the designer needs to emphasize more.

$$P(s) = \frac{(g_1.ls + g_2.p_s + g_3.m_s).((\sum_t R_s)/T)}{C_s} \qquad (26)$$

An observation is selected from the top of the list and its feasibility is checked against the top sensor. If feasible, the observation is added to the sensor's final pattern. If not, the observation feasibility is checked against the next sensor in the list. If the observation turned to be infeasible for all sensors; the observation is discarded. This is repeated until no more observations available or sensors are saturated.

4. Performance Evaluation

In this section, a set of experiments are conducted to measure the performance of the algorithms compared to the optimal solution. 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. Three different sets of experiments are conducted. 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. Sensors' operational characteristics L_s, M_s, E^t_{sij} and P_s are generated randomly as function of the length of monitoring period using the uniform distribution *U(1, T)*. Sensors reliability is assumed to be fixed all the time.

As shown in Table 1, the two heuristics are generally able to achieve reasonable average coverage performance. The coverage performance of movement-based greedy algorithm ranges form 82% (experiment 1) to 98% (experiment 5). The coverage performance of observation-based greedy Algorithm ranges form 81% (experiment 1) to 98% (experiment 9). Global observation-based coverage performance ranges from 91% (experiment 1) to 100% (Experiment 5). In addition, all of the heuristics are running in much less time compared to the running time required to obtain optimal solution. For example, the required time to run the global observation-based algorithm on a field with 20 zones and 10 sensors for 12 units of time is .0000014 % from the required time of the optimal solution.

Table 1: Optimal and near optimal solutions comparison

Exp. No.	No. of Zones	No. of Sensors	Horizon	Movement-Based Greedy Algorithm		Observation-Based Greedy Algorithm		Global Observation-Based	
				Objective (%)	Running Time (%)	Objective (%)	Running Time (%)	Objective (%)	Running Time (%)
1	10	5	12	82	.1	81	.005	91	.002
2	20	5	12	82	.03	94	.002	94	.0003
3	25	5	12	80	.03	94	.003	94.5	.0004
4	30	5	12	87	.02	89	.001	94.6	.0002
5	20	3	12	98	.12	95	.3	100	.002
6	20	5	12	82	.03	94	.002	94	.0003
7	20	10	12	82	.01	86	.0001	98	.0000014
8	20	5	3	89	.09	92	.03	99	.006
9	20	5	6	87	.04	98	.006	99	.001
10	20	5	12	82	.03	94	.002	94	.0003

5. Experimental Design Scenarios

In this section, we illustrate part of *SensDep* capabilities by showing how the tool can be used in answering questions that frequently face architects while planning surveillance operations. Three scenarios are presented. In the first scenario, the tool provides surveillance architects with the relationship between the maximum achievable coverage performance and the corresponding cost of the surveillance operation. A homogenous set of sensors is assumed in this scenario.

In the second scenario, the assumption of having homogenous set of sensors is relaxed. The coverage performance and corresponding surveillance cost are given for different sets of sensors. Each set consists of a combination of three different sensor types. The last scenario illustrates how the architects can evaluate the effect of the sensors attributes on overall coverage performance. It also enables measuring the trade-off among the different attributes.

Scenario 1

In this design scenario, the tool provides the overall coverage performance for different surveillance budgets. Given is a field of 20 zones, which is monitored for twelve time intervals. The field is covered by a homogenous set of sensors. A hypothetical cost value of $100 per sensor is assumed. Figure 6 illustrates the maximum achieved coverage performance and the corresponding cost resulting from using different number of sensors. As expected, increasing number of sensors results in better coverage performance. For instance, if only five sensors are available, an objective function of about 2000 units is achieved. This value jumped to more than 6000 units when 30 sensors are used.

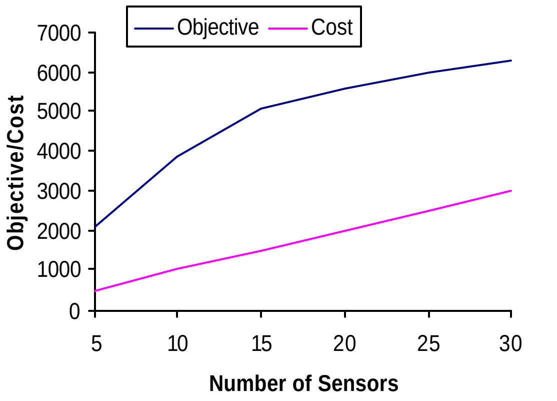

Figure 6: Objective -Cost and number of sensors

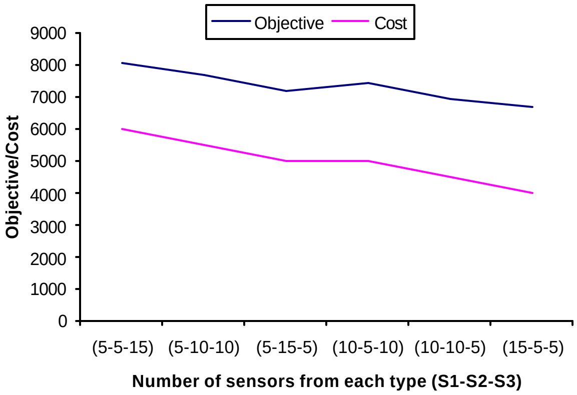

Figure 7: Multi-type sensors

Scenario 2

Now assume the availability of three types of sensors. These sensors differ in their capabilities as well as in their cost per unit. Without loss of generality, sensors with longer lifespan, more reliable and higher self-scheduling and mobile capabilities are assumed to be more expensive. We also assume that no more than fifteen sensors can be used from any of the three available senor types. A field of 20 zones that is monitored for 15 time intervals is used in this experiment. Figure 7 illustrates the coverage performance and the corresponding surveillance cost when different set of sensors are used. These sets consist of different combinations of the three available sensor types. The most expensive combination contains 5 sensors from S1, 5 sensors from S2, and 15 sensors from S3; (5-

5-15). On the contrary, the set (15-5-5) is the least expensive, which costs $4000, results in a coverage performance of about 7000 units.

Scenario 3

In this scenario, the tradeoff between the sensors' different attributes is examined. Figures 8 to 11 illustrate the impact of the sensors' reliability, lifespan, mobility and state switching capabilities on the coverage performance. In these experiments, a homogenous set of sensors are used to cover a filed of 20 zones for 12 time intervals.

Figure 8 illustrates the relationship between the number of sensors and overall coverage performance for different levels of the sensors reliability. As illustrated in the figure, achieving the same coverage performance using less reliable sensors can only be achieved through using more of these sensors. For instance, five sensors with 100% reliability give the same coverage performance of 30 sensors with 25% reliability. The same pattern could also be observed when sensors with different lifespan are examined. As the sensors' lifespan decreases, more sensors will be needed to achieve a certain required coverage performance. As illustrated in Figure 9, to achieve a coverage performance of 3100 units, 15 sensors are needed if the sensors lifespan is equal to the monitoring horizon. This number jumps to about 30 sensors if the sensors lifespan is only 25% of the monitoring horizon.

In Figure 10, the coverage performance for sensors with different mobile capability is examined. As the sensor's maximum number of allowed moves decrease, more sensors are needed to achieve the same coverage performance. Similarly, in Figure 11, using sensors with high state-switching capabilities reduces the number of required sensors to achieve the same coverage performance.

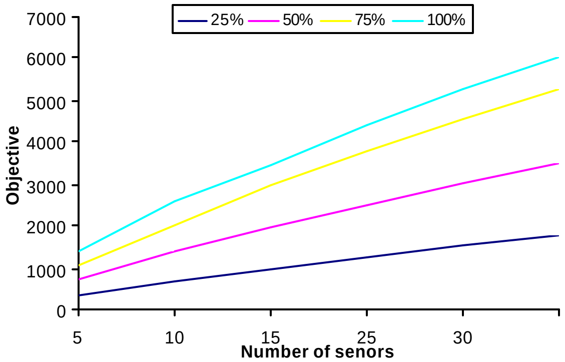

Figure 8: Sensors Reliability

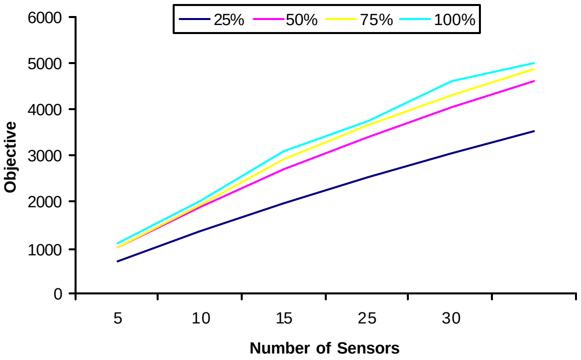

Figure 9: Sensors Lifespan

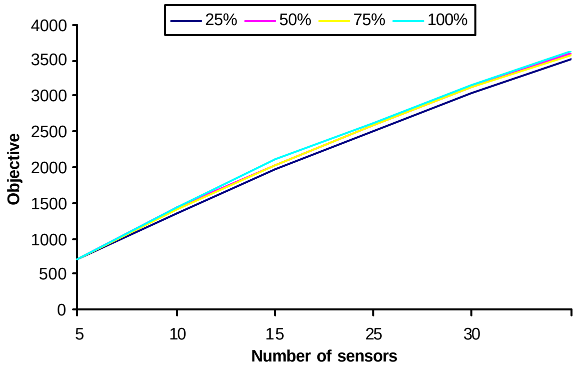

Figure 10: Sensors Mobility

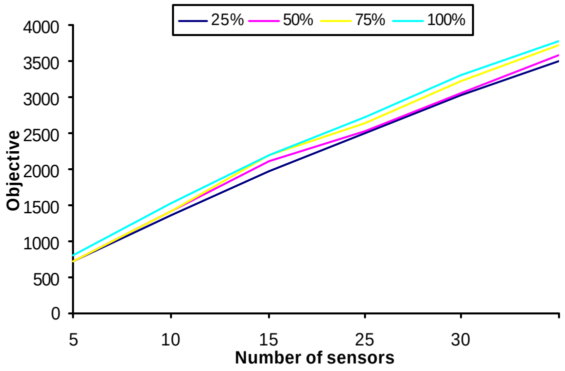

Figure 11: Sensors State Switching

6. Conclusions

In this paper, we presented *SensDep* which is a decision support tool for the design of large-scale automated surveillance operations. The new tool considers the deployment of heterogeneous set of sensing devices in environments with differential surveillance requirements. The tool works in interactive mode to help surveillance architects answer wide variety of “what-if” questions that usually arise in the design of large-scale surveillance operations. The tool consists of two main modules: script and solution. In script module, the designer requirements are edited and translated. In the solution module, the solution for the deployment problem could be obtained optimally using integer mathematical program that is formulated and implemented using CPLEX-80, or through using two heuristic algorithms which provide near-optimal solutions for problems with special structures. To illustrate the different capabilities of the tool, several experimental design scenarios are presented. In these scenarios, the tool provides surveillance architects with the relationship between the maximum achievable coverage performance and the corresponding surveillance cost assuming the use of homogenous and heterogeneous sets of sensors. In addition, the experiments illustrate how the architects can evaluate the effect of the sensors attributes on overall coverage performance and also to measure the trade-off among the different attributes.

7. References

[1] Batalina, M.A. and Sukhatme, G.S., "Sensor Coverage using Mobile Robots and Stationary Nodes ", In Procedings of the SPIE, volume 4868 (SPIE2002), Boston, MA, August 2002, pp. 269-276.

[2] Berman, P., Calinescu, G., Shah C. and Zelikovsky, A., "Power Efficient Monitoring Management in Sensor Networks," Proceedings of IEEE Wireless Communication and Networking Conference (WCNC'04), Atlanta, March 2004, pp. 2329-2334.

[3] Bhatnagar, S., Deb B., and Nath, B., "Service Differentiation in Sensor Networks," Presented at the Fourth Symposium on Wireless Personal Multimedia Communications, September 2001.

[4] Cardei, M. and Wu, J., "Coverage in Wireless Sensor Networks," in Handbook of Sensor Networks," M. Ilyas and I. Mahgoub (eds.), CRC Press, 2005, pp. 361-372.

[5] Chakrabarty, K., S. S. Iyengar, H. Qi and E. Cho, "Grid coverage for surveillance and target Location in distributed sensor networks," IEEE Transactions on Computers, vol. 51, December 2002, pp. 1448–1453.

[6] Chvatal, V., "A Combinatorial Theorem in Plane Geometry", Journal of Combinatorial Theory (B), vol. 18, 1975, pp 39-41.

[7] Howard, A., Mataric, M.J., and Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks ", G.S. Autonomous Robots, Special Issue on Intelligent Embedded Systems, vol. 13 , no. 2, September 2002, pp. 113-126.

[8] Hu, W., Chou C-T, Jha, S. and Bulusu, N., "Deploying Long-lived and Cost-effective Hybrid Sensor Networks," In Proceedings of the First Workshop on Broadband Advanced Sensor Networks (BaseNets 2004), San Diego, 2004.

[9] Ibarra O.H. and Kim C.E. "Fast approximation algorithms for the knapsack and sum of subset problems." JACM 22 (1975), 463-468

[10] Iyegnar, S., Parsad, L. and Min, H., "Advances in Distributed Sensor Technology," Englewood Cliffs, N.J.: Prentice Hall, 1995.

[11] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," Wireless Networks, vol. 8, no. 2/3, pp. 169–185, 2002.

[12] O'Rourke, J., "Galleries Need Fewer Mobile Guards: A Variation on Chvatal's Theorem", Geometriae Dedicata vol. 141983, pp 273-283.

[13] R. Ramadan, K. Abdelghany, and H. El-Rewini, "On the Optimal Deployment of Heterogeneous Sensing Devices," accepted in AICCSA-06, 2006.

[14] S. Sahni, “Approximate algorithms for 0/1 knapsack problem”, Journal of the ACM, Vol. 22 , pp. 115-124 , 1975.][Bruno R. Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++” , ISBN 0471-24134-2, 1997

[15] Sahni S. "Approximation algorithms for the 0/1 knapsack problem." JACM 22 (1975), 115-124.

[16] T. van Dam and K. Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks,” in Proceedings of the first international conference on Embedded networked sensor systems, 2003, pp. 171–180.

[17] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energyefficient collision-free medium access control for wireless sensor networks,” in Proceedings of the first international conference on Embedded networked sensor systems, 2003, pp. 181–192.

[18] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” 2002.

[19] Wang, G. Cao, La Porta, T.F., “A Bidding Protocol for Deploying Mobile Sensors,” in Proceedings of IEEE ICNP, November 2003.

[20] Xavier Gandibleux, Arnaud Freville: Tabu Search Based Procedure for Solving the 0-1 MultiObjective Knapsack Problem: The Two Objectives Case. J. Heuristics 6(3): 361-383 (2000)

[21] Yan, T., He, T. and Stankovic, J., “Differentiated Surveillance for Sensor Networks,” *Presented at SenSys’03*, Los Angeles, California, USA, November 2003.