On the design of wireless sensor networks for mobile target detection

Edoardo Amaldi   Antonio Capone   Matteo Cesana   Ilario Filippini

Dipartimento di Elettronica e Informazione, Politecnico di Milano
Piazza Leoardo da Vinci 32, 20133 Milano, Italy

Abstract

We consider surveillance applications through wireless sensor networks in which one has to select among a set of candidate sites where to locate sensors so as to detect mobile targets traversing a given area. By adopting the concept of exposure as a measure of coverage quality, we address two problem versions: the minimization of the sensors installation costs while guaranteeing that the exposure of any path is above a certain threshold, and the maximization of the exposure of the least exposed paths subject to a budget on the sensors installation costs. We give compact mixed integer linear programming formulations that also account for wireless connectivity constraints and can be solved to optimality for medium-size instances. Then we outline tabu search heuristics that provide close-to-optimum solutions of large-size instances in short computing time.

Keywords: *Sensor networks, location, connectivity, bilevel mixed integer programming.*

1   Introduction

Wireless Sensor Networks (WSNs) are composed of small sized battery-operated devices with embedded sensing, processing, and wireless communication capabilities. Due to their low cost and flexibility, WSNs have emerged as effective solutions in a variety of applications involving, among others, event detection, target tracking, environmental and industrial monitoring [1].

In this work, we focus on mobile target detection in accessible environments where sensors positions can be optimized. Roughly speaking, given an area and a set of candidate sites where sensors can be installed, the problem is to decide where to locate sensors so as to make it as likely as possible an intruder crossing the area is detected. Mobile target detection is of utmost importance in several practical scenarios, including border protection/surveillance and critical areas surveillance (museums, factories, departments, etc.).

Most of the previous work on WSN design deals with coverage models with on/off sensing range (for instance, where the sensors must satisfy some connectivity conditions and each point of the area must be within the range of k sensors) that are suited to the detection of localized events or static targets (see e.g. [2, 3]).

In this paper we make a step forward by considering, instead of the above 0-1 definition of coverage, the concept of *path exposure* introduced in [4]. Intuitively, the more exposed a path, the better the coverage provided by the WSN, and the higher the probability to detect the mobile object moving along that path.

Common sensing mechanisms are based on the energy of the signal received from the target (either generated or just reflected by it). If l sensors located in $s_1 \ldots s_l$ cooperate to detect the target in position p, the total energy detected by the cooperating set is: $I(p) = \sum_{i=1}^{l} I_{s_i}(p)$, where $I_{s_i}(p)$ denotes the

energy of the signal received by a sensor at position s_i from a target in p. Then the exposure along a path P can be defined as:

$$E(P) = \int_P I(p)dp = \sum_{i=1}^{l} \int_P I_{s_i}(p)dp, \tag{1}$$

where $\int_P$ represents the line integral along path P [4]. Noise in the detection process can be easily taken into account [8].

So far the concept of path exposure has only been used to assess the performance of randomly deployed WSNs [5, 8], identify the most/least exposed paths for given network topologies [6], and compute the minimum node density that guarantees a given quality [7, 9].

Since the exposure value of the least exposed path is a measure of the vulnerability level of the network, we address the two following WSN design problem versions. In the first one (P_1), sensors must be positioned in order to maximize the exposure of the least exposed path(s), subject to a budget B on the total installation costs. In the second one (P_2), sensors have to be positioned so as to minimize the installation costs, provided that the exposure of the least-exposed path(s) is above a given threshold $\tau > 0$.

2 Mixed integer linear programming models

Given the area of interest, let S denote the set of all candidate sites (CSs) where sensors can be installed. The area is approximated by a grid on which targets can move. The grid graph $G = (V, A)$ is defined as follows. The vertex set V includes one vertex for each grid intersection point, and two distinguished vertices o and t that represent the virtual origin and, respectively, destination of any path. The arc set A includes two arcs (i, j) and (j, i) for each grid edge $\{i, j\}$, the incoming/outgoing arcs connecting o to the leftmost column vertices of the grid, and the rightmost column vertices to t, respectively. Without loss of generality, we assume that G is a square grid of size $n \times n$.

Given an instance defined by (S, V, A) with two special vertices o and t, and a budget value B, problem (P_1) consists in deciding where to install the sensors so as to maximize the exposure of the least exposed path from o to t while guaranteeing a total installation cost of at most B. This problem can be viewed as an extension of the network interdiction problem [11], restricted to two-dimensional grid graphs.

Consider the following decision variables: y_s is equal to 1 if a sensor is installed in CS s and 0 otherwise, x_{ij} is equal to 1 if arc (i, j) belongs to the least exposed path and 0 otherwise, and the non-negative continuous variable z corresponding to the exposure of the least exposed path. These variables lead to the following formulation:

$$\max \quad z \tag{2}$$

$$\text{s.t.} \quad \sum_{s \in S} c_s y_s \leq B \tag{3}$$

$$\min \sum_{(i,j) \in A} \left(\sum_{s \in S_{ij}} e_{ij}^s y_s \right) x_{ij} \geq z \tag{4}$$

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1 & i = o \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V \tag{5}$$

$$x_{ij} \geq 0 \quad \forall (i, i) \in A \tag{6}$$

$$y_s \in \{0, 1\} \quad \forall s \in S, \tag{7}$$

where c_s is the cost of installing a sensor in CS s, and the parameter e_{ij}^s is the exposure estimated according to (1). The set S_{ij}, which denotes the set of all CSs that cover/affect the arc (i, j), clearly depends on the devices' sensing range. A CS belongs to S_{ij} if its exposure over arc (i, j) is above a certain threshold. Constraint (3) enforces the upper bound on the installation budget. Since z is maximized,

constraint (4) ensures that z is equal to the exposure of the least exposed path. As expected, each installed sensor affects the exposure (weight) of all the arcs in S_{ij}, and the exposure (weight) of arc (i,j) is given by $\sum_{s \in S_{ij}} e_{ij}^s y_s$. Constraints (5) are the flow balance equations that define a path from the origin o to the destination t.

Although the mixed integer bilevel mathematical program (2)-(7) is nonlinear due to constraint (4), it can be linearized by applying linear programming strong duality to the shortest path minimization subproblem (4)-(6). This leads to the following equivalent mixed integer linear programming (MILP) formulation (F_1):

$$\max \quad z \tag{8}$$

$$\text{s.t.} \quad \sum_{s \in S} c_s y_s \leq B \tag{9}$$

$$\pi_t - \pi_o \geq z \qquad \pi_o = 0 \tag{10}$$

$$\pi_j - \pi_i \leq \sum_{s \in S_{ij}} e_{ij}^s y_s \qquad \forall (i,j) \in A. \tag{11}$$

Note that for each arc of G there is a single linear constraint (11) involving the corresponding dual π variables.

In problem (P_2), we minimize the total installation costs while guaranteeing that the exposure of any path traversing the detection area is above a certain threshold $\tau > 0$. Similarly, by strong duality, we have the MILP formulation (F_2):

$$\min \quad \sum_{s \in S} c_s y_s \tag{12}$$

$$\text{s.t.} \quad \pi_t - \pi_o \geq \tau \qquad \pi_o = 0 \tag{13}$$

$$\pi_j - \pi_i \leq \sum_{s \in S_{ij}} e_{ij}^s y_s \qquad \forall (i,j) \in A. \tag{14}$$

From the complexity point of view, it is easy to show that both basic problems (P_1) and (P_2) are NP-hard even when the grid is a single path and each CS can detect the target across a single arc. Indeed, there is a simple polynomial-time reduction from the recognition version of the knapsack problem.

We now extend the above basic MILP models to account for sufficient wireless *connectivity* among the sensors to allow for cooperative decisions. In particular, constraints on the wireless backbone include node capacity and multi-hop connectivity.

Let us define the communication range as the maximum distance at which a transmission from a sensor can be correctly decoded at a receiver. A sink node acts as a centralized server that collects events detected by sensors, and estimates whether an intruder is present.

Consider a new directed graph $G' = (S, A')$ with one vertex per CS and an arc set $A' \subseteq S \times S$ containing one arc (i,j) for each pair of CSs that are within communication range. A special node σ and its relative incoming connections are added to S and A' to represent the sink. To guarantee connectivity among the sensors and the sink, a unit of traffic is required from each installed sensor to the sink. If a continuous non-negative variable f_{ij} represents the traffic flow on arc $(i,j) \in A'$, the extension of (F_1)

accounting for connectivity, referred to as (F_{1C}), is:

$$\max \quad z \tag{15}$$

$$\text{s.t.} \quad \sum_{\substack{s \in S: \\ s \neq \sigma}} c_s y_s \leq B \tag{16}$$

$$\pi_t - \pi_o \geq z \quad \pi_o = 0 \tag{17}$$

$$\pi_j - \pi_i \leq \sum_{\substack{s \in S_{ij}: \\ s \neq \sigma}} e_{ij}^s y_s \qquad \forall (i,j) \in A \tag{18}$$

$$\sum_{(s,r) \in A'} f_{sr} - \sum_{(r,s) \in A'} f_{rs} = \begin{cases} -y_r & r \neq \sigma \\ \sum_{\substack{q \in S: \\ q \neq \sigma}} y_q & r = \sigma \end{cases}$$
$$\forall r \in S. \tag{19}$$

The objective function (15) and constraints (16)-(18) are as in (F_1). Constraints (19) are flow balance equations for the graph G', ensuring that one unit of flow is sent from each installed sensor to the sink σ.

Connectivity can thus be imposed by adding constraints that do not modify the shortest-path subproblem and only place additional conditions on sensor installation. Similarly, model (F_2) can be extended to account for connectivity by adding constraints (19).

Sensor's RF equipment has usually limited bandwidth and energy resources. In general, we want to avoid overloading a sensor with communication relay traffic, thus leading it to early depletion or load congestion. To this end, we introduce a node capacity u_s that limits the maximum sum of flows entering and exiting each node s, and we add the following constraints:

$$\sum_{(r,s) \in A'} f_{rs} + \sum_{(s,r) \in A'} f_{sr} \leq u_s y_s \qquad \forall s \in S : s \neq \sigma. \tag{20}$$

For failure tolerance purposes, one or more backup paths should be selected so that the sink is reached even in case sensors fail. This can be achieved by imposing k-connectivity on the subgraph of graph G' defined by the installed sensors.

3 Tabu search heuristics

To tackle large-size instances of both problem versions, we have developed Tabu Search (TS) algorithms. The rationale is to try to improve an initial solution by iteratively installing or removing sensors selected on the basis of their exposure contribution to a least exposed path of the current solution. We consider a typical WSN scenario where c_s are equal for all CSs.

First problem version. The objective is to maximize the minimum path exposure subject to a budget constraint on the installation costs. Let $\mathbf{y}^0$ be a random initial solution satisfying the budget constraint. We consider swap moves in which a sensor in the current feasible solution $\mathbf{y}^k$ is deleted while a new sensor is installed in an empty CS (a CS i with $y_i^k = 0$). At each iteration, given a least exposed (shortest) path P_k corresponding to $\mathbf{y}^k$, all the CSs are ordered according to their contribution $E_i(P_k)$ to the path exposure. Consider the l empty CSs with the largest such values and the l CSs where a sensor is located with the lowest such values. Then the next iterate $\mathbf{y}^{k+1}$ is obtained by performing, among all the above l^2 possible swaps, the swap leading to the best objective function value.

In order to prevent cycling (consider feasible solutions that have already been considered) and try to escape from local maxima, a list of "tabu moves" is maintained. The tabu list is implemented in a simple way: sensors that are installed (deleted) cannot be deleted (reinstalled) during L iterations. According to the "aspiration criteria", tabu moves that lead to a best found solution can be performed. To favor search diversification, if the best solution found in the neighborhood does not improve the current solution during R moves, a random swap is carried out: a randomly selected sensor is deleted from $\mathbf{y}^k$ while a

sensor is installed in an empty CS selected at random. The best solution found during the search process is stored and returned after a maximum number of iterations max_{it}.

The computational results reported in the next section have been obtained with the following parameter settings: $l = 15$, $L = 4$, $R = 2$, $max_{it} = 250$.

Second problem version. The objective is to minimize the total installation costs while guaranteeing a minimum path exposure τ. We start from the initial solution $\mathbf{y}^0$ where all sensors are installed. Each iteration consists of two steps: 1) m sensors are deleted from the current solution $\mathbf{y}^k$ to obtain $\tilde{\mathbf{y}}^k$, 2) the "neighborhood" $N(\tilde{\mathbf{y}}^k)$ is explored as in the previous algorithm looking for a better solution.

In the first step, given a least exposed path P_k corresponding to $\mathbf{y}^k$, all the CSs are ordered according to their contribution $E_i(P_k)$ to the path exposure. By randomly selecting m of the first h CSs and removing them from the solution (setting $y_i^k = 0$ for the corresponding i), we obtain $\tilde{\mathbf{y}}^k$. The second step consists of swap moves, similar to those in the algorithm for the first problem version. At each second-step iteration j, given a least exposed path P_{k_j} corresponding to $\tilde{\mathbf{y}}^{k_j}$, all the CSs are ordered according to their contribution $E_i(P_{k_j})$ to the path exposure. Consider the l empty CSs with the largest such values and the l CSs where a sensor is located with the lowest such values. Then the next iterate $\mathbf{y}^{k_{j+1}}$ is obtained by performing, among all the above l^2 possible swaps, the swap leading to the best objective function value. The second step ends after max_j iterations giving the next $\mathbf{y}^{k+1}$.

The tabu list is as above. Sensors that are installed (deleted) in the first or the second step cannot be deleted (reinstalled) during the next L iterations of the same type.

The overall algorithm is characterized by two phases. The first one aims at getting close to a good solution and the second one at refining and widening the search for the best solution. In the first phase, named Steepest Descent, $m = 5$ among h CSs are removed in the first step and $l = 5$ CSs are considered in the second step, at each iteration k. The Steepest Descent phase continues until it reaches iteration k' where the best solution found during its second step has a path with a smaller exposure than the desired τ. Sensors removed during the first step of iteration k' are reinstalled and the second phase, named Slow Descent, begins at iteration $k' + 1$. During the Slow Descent phase $m = 1$ among the h CSs is removed in the first step and $l = 10$ CSs are considered in the second step of every iteration k. The Slow Descent phase stops when the second step of the current iteration ends providing a solution which does not guarantee the desired minimum exposure τ. The algorithm stops when the Slow Descent phase ends, the returned solution is the solution $\mathbf{y}^{k^*}$ of the next to last iteration.

Finally, we have included a multi-start strategy starting from the end of the Steepest Descent phase. At each multi-start run, the first set of h CSs candidate to be removed is randomly generated. After n_r scheduled runs, the solution of the run which ended with the best results (smallest number of installed sensors) is returned as final solution.

Computational results reported in the next section have been obtained with the following parameter settings: $L = 4$, $h = 20$, $max_e = 10$, the above mentioned m and l values, $n_r = 3$.

4 Some computational results

To evaluate the performance of our compact MILP formulations and TS heuristics, we report some computational results obtained for realistic randomly generated instances. We consider a 80m × 80m square area with 80 CSs. The sensing range is set to 10m. In the variants with connectivity constraints, the communication range is 15m and a random sink node with a communication range of 20m is included. In the variants with sensor capacity, the maximum amount of traffic per node is equal to the total communication load generated by 15 sensors. Computational tests are run on a 3.0GHz PC with 1GB of RAM under Linux. The MILP formulations are solved with CPLEX 10.0 and our TS heuristics are implemented in C++ using Boost libraries. Results are averages on 20 randomly generated instances.

In Table 1, we compare the optimal solutions of the compact MILP formulation (F_1) with the solutions provided by the TS algorithm. Different classes of instances with two values of the grid density (25×25 and 50×50) and three different values of the budget B on the number of installed sensors ($B = 30, 40, 50$ and $c_s = 1$ for all s) are considered. The three pairs of columns from 3rd to 8th refer to the optimal

layouts, and indicate the average and standard deviation values of the computing time, the minimum exposure, and the actual number on installed sensors, respectively. The two last pairs of columns report the same information for the TS algorithm (the number of sensors is omitted since it is always equal to the budget).

Grid	B	Optimum						Heuristics			
		Time [s]		Exposure		# Installed Sensors		Time [s]		Exposure	
		average	std dev	average	std dev	average	std dev	average	std dev	average	std dev
25x25	30	11716.08	4854.78	0.47	0.07	30.00	0.00	38.23	0.48	0.45	0.07
25x25	40	4326.49	6601.95	0.52	0.10	40.00	0.00	38.47	0.39	0.51	0.09
25x25	50	2245.67	4489.82	0.59	0.17	48.60	3.56	30.85	0.16	0.58	0.17
50x50	30	12997.45	4332.03	0.40	0.07	30.00	0.00	181.25	2.24	0.39	0.07
50x50	40	10231.60	6433.64	0.50	0.12	39.30	2.10	186.14	1.57	0.49	0.11
50x50	50	5790.37	7081.13	0.55	0.16	48.30	5.10	150.29	1.22	0.54	0.15

Table 1: *Comparison between exact and heuristic solutions for the first problem version.*

First, note that CPLEX allows to solve to optimality medium-to-large size instances within 4 hours of CPU time. The TS algorithm performs very well on all instances: it provides solutions that are within 5% of the optimum in at most 3 minutes. Some aspects of the solutions are worth pointing out. For a given budget, the minimum exposure slightly decreases as the grid density increases, because higher-density grids allow for the representation of more accurate least exposed paths. Note that the number of deployed sensors is sometimes lower than the budget because it may happen that, due to the limited sensing range, none of the empty CSs can improve the exposure of all the least exposed paths.

Our TS heuristic can tackle much larger instances in reasonable time. Figure 1 depicts an example of network topology (obtained in 24 minutes of CPU time) spanning an area of 400m×400m, with 200×200 grid, 1500 CSs and a budget $B = 1000$.
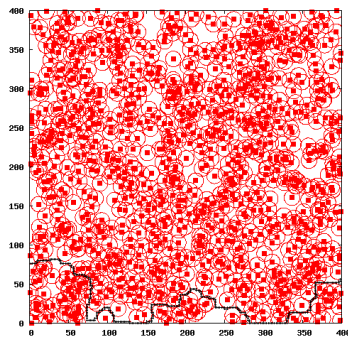

Figure 1: *Example of a large size network planned using the TS algorithm.*

When wireless connectivity and node capacity constraints are included, the first problem version is more challenging. Table 2 shows the results obtained for problem formulation (F_1) with 2-connectivity of the communication graph and node capacity. The exposure values are clearly smaller than the corresponding ones in Table 1. The price to pay in terms of budget in order to provide connectivity and resilience to the wireless backbone and to limit per-node traffic, is not negligible. For example, to achieve almost the same exposure level corresponding to $B = 30$ in Table 1, the number of sensors to be installed is now $B = 50$.

Grid	B	Time [s]		Exposure		# Installed Sensors	
		ave.	std dev	ave.	std dev	ave.	std dev
50x50	60	7401.66	8378.80	0.46	0.09	45.80	3.79
50x50	50	8414.59	9054.83	0.41	0.20	43.50	9.59
50x50	40	13686.04	9773.73	0.36	0.11	37.80	3.09
50x50	30	15116.85	8214.14	0.32	0.14	28.70	3.29

Table 2: *Exact solutions for the first problem version with 2-connectivity and capacity constraints.*

The results obtained for the minimum cost problem version are reported in Table 3. With a CPU time limit of 4 hours, we are able to solve to optimality only instances with a minimum exposure of up

to 0.35, with a 25×25 grid, and up to 0.20, with a 50×50 grid. The TS algorithm performs very well also for this problem version: it provides solutions that are within 5% of the optimum in no more than 5 minutes. As in Table 1, for higher-density grids a larger number of sensors must be installed to guarantee the same minimum exposure.

| | | Optimum | | | | Heuristic | | | |
| | | Time [s] | | # Installed sensors | | Time [s] | | # Installed sensors | |
Grid	Exposure	average	std dev	average	std dev	average	std dev	average	std dev
50x50	0.20	9110.96	5983.07	15.40	1.51	272.96	41.89	16.10	2.42
50x50	0.15	8496.64	6383.17	12.60	0.84	225.81	51.84	12.70	0.82
50x50	0.10	7298.60	5950.95	9.90	0.57	149.94	28.63	10.40	0.52
50x50	0.05	9279.45	4891.95	7.30	0.48	96.69	21.16	7.40	0.52
25x25	0.35	6221.35	6047.92	23.00	3.09	66.98	9.70	23.60	3.24
25x25	0.30	3597.90	2619.76	19.90	2.18	72.24	13.14	20.50	2.80
25x25	0.25	3474.35	4680.67	17.00	2.26	64.96	13.66	17.20	2.10
25x25	0.20	4360.92	6427.70	14.50	2.01	52.70	12.55	14.80	1.69
25x25	0.15	2746.77	4434.32	12.10	0.74	44.27	9.75	12.10	0.74
25x25	0.10	1433.76	1558.14	9.80	0.42	35.18	6.14	9.90	0.57
25x25	0.05	965.86	601.28	7.20	0.42	18.06	4.34	7.40	0.52

Table 3: *Comparison between exact and heuristic solutions for the second (minimum cost) problem version.*

To conclude, our compact MILP formulations yield optimal solutions for medium-size instances, and our TS heuristics provide near-optimal solutions in short computing time and can be used to tackle large-size instances.

References

[1] I. F. Akyildiz et al., *A Survey on Sensor Networks*, IEEE Comm. Mag., **40** (8) (2002) pp. 102–116.

[2] X. Bai, S. Kumar, D.Xuan, Z.Yiun, T.H.Lai, *Deploying Wireless Sensors to Achieve Both Coverage and Connectivity*, in ACM MobiHoc 2006, pp. 131–142.

[3] M. Cardei and D.-Z. Du, *Improving Wireless Sensor Network Lifetime through Power Aware Organization*, ACM Wireless Networks, **11**(3) (2005) pp. 333–340.

[4] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, *Exposure in Wireless Ad Hoc Sensor Networks*, in ACM MobiCom 2001.

[5] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, *Worst and Best-Case Coverage in Sensor Networks*, IEEE Trans. on Mobile Computing, **4**(1) (2005) pp. 84–92.

[6] G. Veltri, Q. Huang, G. Qu and M. Potkonjak, *Minimal and maximal exposure path algorithms for wireless embedded sensor networks*, in ACM SenSys 2003, pp. 40–50.

[7] S.S. Ram, D. Manjunath, S.K. Iyer, and D. Yogeshwaran, *On the Path Coverage Properties of Random Sensor Networks*, IEEE Trans. on Mobile Computing, **6**(5) (2007) pp. 446–458.

[8] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and K. K. Saluja, *Sensor depolyment strategy for detection of targets traversing a region*, ACM Mobile Netw. and Appl., **8**(4) (2003) pp. 453–461.

[9] S. Adlakha and M. B. Srivastava, *Critical Density Thresholds in Wireless Sensor Networks*, IEEE Wireless Comm. and Networking **3** (2003) pp. 16–20.

[10] F. Glover and M. Laguna, *Tabu Search* , Kluwer Publishers, 1997.

[11] E. Israeli and R. K. Wood, *Shortest-path network interdiction*, Networks, **40**(2) (2002) pp. 97–111.