Exploiting Sensor Spatial Redundancy to Improve Network Lifetime

A. Alfieri[1], A. Bianco[2], P. Brandimarte[1], C.-F. Chiasserini[2]
[1] Dipartimento di Sistemi di Produzione ed Economia dell'Azienda, Politecnico di Torino
[2] Dipartimento di Elettronica, Politecnico di Torino
E-Mail: {arianna.alfieri,paolo.brandimarte,andrea.bianco,carla.chiasserini}@polito.it

***Abstract*— One of the most critical issues in wireless sensor networks is represented by the limited availability of energy within network nodes; thus, making good use of energy is a must to increase network lifetime. We define as network *lifetime* the period from the time instant when the network starts functioning till the network runs satisfying its quality requirements, i.e., a given level of coverage in the area of interest is guaranteed. To maximize system lifetime, we exploit sensor spatial redundancy by defining sub-sets of sensors active in different time period, to allow sensors to save energy when inactive. Two approaches are presented: the first one, based on mathematical programming techniques, must run in a centralized way, whereas the second one is based on a greedy algorithm aiming at a distributed implementation. To asses their performance and provide guidance to network design, the two approaches are compared by varying several network parameters.**

I. Introduction

Sensor networks are composed by small electronic devices, named sensors, which can perform remote monitoring and object-tracking in different environments and for a wide range of applications. Due to their low-cost and low-complexity nature, sensors are characterized by several constraints, such as a short transmission range, poor computation and processing capabilities, low reliability and data transmission rates, and a limited available energy. Thus, sensor networks should be designed with the aim to overcome these limitations, e.g., by exploiting the synergy between multiple nodes.

The limited availability of energy within sensor nodes is one of the most critical issues. Indeed, recharging or replacing the nodes' battery may be inconvenient, or even impossible in disadvantaged working environments. This implies that the time during which all sensors are able to sense, transmit, receive and process information is limited; and, the network *lifetime*, i.e., the interval during which the network functions properly, becomes an important performance metric. There are various possible definitions for network lifetime, depending on the network application. In this work, we define network lifetime as the time spanning from the instant when the network starts functioning till a given level of coverage of the area of interest can be guaranteed. Our objective is to devise solutions that maximize network lifetime.

We take as a case study a video-surveillance network for monitoring a given territorial area (for simplicity a rectangular area), named area of interest, with a desired level of coverage. While monitoring the area of interest, sensors gather information (i.e., images), and send it to some gateway node. Sensors that are unable to reach by direct transmission the gateway node, deliver the collected information by using intermediate sensors as relays. We assume that sensors can be switched off if needed to reduce power consumption. We also assume that the number of deployed sensors is large enough that sensors sub-sets can provide the desired level of coverage, if they are properly chosen. Given that sensor battery lifetime should be maximized to maximize network lifetime, a fairly intuitive approach is to switch on, at a given time, only the minimum number of sensors needed to guarantee the desired level of coverage in the area. Based on the above observation, we divide sensors into sub-sets, each sub-set being active in different period of time, and devise an optimal scheduling of sensors' activity, so that the sensor battery lifetime is maximized and the quality of service requirements (e.g., desired level of coverage) are met.

Our problem can be regarded as a generalization of the set partitioning approach proposed by [1]. Their approach entails finding the maximum number of disjoint subsets of sensors, which is NP-hard. In fact, if we assume that energy is consumed only for sensing and not for transmitting data, it is easy to see that their problem can be transformed into our problem, which is therefore NP-hard as well. Thus, we must rely on heuristic solution methods. In this paper we propose two approaches: the first one is based on a mathematical programming model and is defined in detail in Sec. IV. The second is a greedy approach that should be more easily implemented in a distributed way in a realistic scenario, although the definition of a proper protocol to support the proposed approach is beyond the scope of this paper. The second approach is described in Sec. V.

The main critical point of the first approach is that we assume a centralized management scheme, in contrast to the literature concerned with distributed protocols and decentralized management (see, e.g., [2]). While this is a debatable assumption, and there are application fields in which this approach would probably be too difficult to pursue, we think that the main contribution of the approach is in providing a flexible framework that can be easily extended to deal with more complex scenarios, as hinted later.

II. System Model Description and Assumptions

Different models have been proposed for sensor networks depending on the type of sensing involved and the specific application. The basic network setting that we assume in this paper rely on the work in [1], [3], [4].

We consider a set of sensors, indexed by $j = 1, \dots, N$, whose placement is known. Then, we consider a discrete set of points as a good sampling of the region, indexed by $i = 1, \dots, M$. The quality of coverage may be measured and a minimal quality is required; here, we require that at least a certain percentage of points are covered. Since we do not address the issues of information redundancy and reliability, we assume that a point is covered by at most one sensor (of course, a sensor may cover different points).

For each point i we know the set S_i of nodes which can cover i, and for each sensor j, we know the set of points C_j that it can cover and the set R_j of reachable nodes, as communication may be limited both by distance and natural obstacles.

All sensors are equal and have an initial energy endowment E. A special node, the gateway, is denoted by G; its energy is unbounded. The gateway is the node to which all the data must be routed. The other sensors may be used in different roles; indeed, in the following we will speak of *nodes* rather than sensors to point out the multiple roles they may take. Roles are not mutually exclusive. Examples of roles are: sensing (which we assume continuous rather than event-driven), compressing data, routing data (implies either sensing and transmitting or receiving and transmitting) to another node or the gateway. Data compression may be performed by a sensing node to minimize energy consumption.

For each point i, we define as d_i the corresponding data flow, that is given as an input in the node covering the point (e.g., number of images per unit time). In the following we assume that the data rate is the same for any point. We also assume as known:

- E^T_{jk}: the energy required to transmit data (e.g., an image) from node j to node k; it depends on the distance;
- E^R: the energy required by any node to receive data (e.g., an image);
- E^C: the total energy required to sense and compress data.

Furthermore, we have a lower bound L on coverage, i.e., the minimal percentage of points which must be covered. Hot spots and more refined measures of coverage may be easily dealt with by adapting the models below.

III. Problem Statement

As a first step, we leave aside the concept of sensor sub-nets and consider a unique set of sensors. We introduce the following decision variables.

- $x_{ij} \in \{0, 1\}$, set to 1 if point i is covered by node j, in which case point i contributes an input flow d_i into node j;
- $w_j \geq 0$: data flow rate from node j to the gateway (e.g., compressed images per unit time);
- $f_{jk} \geq 0$: data flow rate from node j to node k (e.g., compressed images per unit time).

We write the power required by a node as,

$$P_j = \sum_{i \in C_j} E^C d_i x_{ij} + \sum_{k \in R_j} E^T_{jk} f_{jk} + \sum_{k \in R_j} E^R f_{kj} + E^T_{jG} w_j.$$

Notice that when node j cannot reach the gateway, i.e., $j \notin R_G$, then $w_j \equiv 0$. By multiplying the power by the system lifetime T, we get the energy consumption which cannot exceed the energy available at any node. To get a linear model, the objective of maximizing system lifetime can be rephrased in terms of balancing the power requirement across nodes. In other words, by minimizing the maximum power consumed $P_{\max} = \max_j P_j$ across the nodes, subject to coverage constraints, we maximize system lifetime. Thus, in this case, we have the following MILP (Mixed Integer Linear Programming) model:

$$\text{(L1)} \quad \min \quad P_{\max}$$

$$\text{s.t.} \quad \sum_{i \in C_j} d_i x_{ij} + \sum_{k \in R_j} f_{kj} = \sum_{k \in R_j} f_{jk} + w_j \quad \forall j = 1, \dots N \tag{1}$$

$$\sum_{j \in S_i} x_{ij} \leq 1 \qquad \forall i \tag{2}$$

$$\sum_i \sum_{j \in S_i} x_{ij} \geq L \cdot M \tag{3}$$

$$P_{\max} \geq \sum_{i \in C_j} E^C d_i x_{ij} + \sum_{k \in R_j} E^T_{jk} f_{jk} + \sum_{k \in R_j} E^R f_{kj} + E^T_{jG} w_j \quad \forall j = 1, \dots N \tag{4}$$

$$x_{ij} \in \{0, 1\},\ w_j \geq 0,\ f_{jk} \geq 0. \tag{5}$$

Eq. (1) expresses conservation of flows. Constraint (2) states that each point must be assigned to at most one sensor, whereas (3) enforces the minimal required coverage. Finally, (4) sets the maximum power $P_{\max}$.

The above formulation does not exploit the possibility of switching sensors off. To improve system lifetime, provided enough redundancy in the sensors is available, we introduce the concept of sensor sub-nets. We extend the approach of [1] by building a partition of nodes in terms of disjoint sub-sets of nodes covering the points of interest. Given each sub-set, which basically amounts to fixing the assignment decision variables x_{ij}, we build the complete network by solving the above formulation with respect to the flow variables w_j and f_{jk}. However, the problem is too complex to be solved with a direct approach. Therefore, we resort to two heuristic solutions described in the following sections: one based on the decomposition method of column generation, the other based on a greedy approach.

IV. A COLUMN-GENERATION BASED DECOMPOSITION FRAMEWORK

We describe here the mathematical programming approach based on column generation. The overall problem we consider has two main components:

1) a routing component, which is linked to defining roles for each node and assigning points to nodes;
2) a scheduling component, since a node may play different roles in different time instants.

The idea is generating a set of network configurations (sub-nets), each of which is connected and meets the minimal coverage requirements. Then we should decide how much time each sub-net is used. By alternating the configurations, we exploit the available redundancy in sensors.

Column generation is a general purpose framework which has been often proposed either as a computationally efficient alternative to standard methods or as a modeling tool when a direct approach is infeasible. See, e.g., [5, chapter 11] for a tutorial treatment, or [6] for a recent survey.

In our case, columns correspond to sub-nets, i.e., network configurations. The aim of the master problem, described below, is to select the columns and to decide the length of the time interval a sub-net is used, subject to energy budget constraints for each node. From the dual variables of the energy budget constraints we derive costs which are used in the column (sub-net) generation sub-problem. The sub-net generation sub-problem aims at finding a feasible sub-net ensuring the minimal required coverage.

A. The sub-net combination (master) problem

Let s be an index referring to a sub-net generated by the sub-net generation sub-problem. Each sub-net is characterized by the role of each node and by the power required for that role. Since we do not assume mutually exclusive roles (each node may both sense and route data from other sensors), the role is basically characterized by the input and the output flows through each node.

Let P_j^s be the power required for node j in sub-net s (it may be 0 if the node is not activated) and $t^s \geq 0$ a decision variable corresponding to the time sub-net s is used. Then, to maximize the network lifetime we solve the master problem (MP):

$$\begin{aligned} \text{(MP)} \quad \max \quad & \sum_s t_s \\ \text{s.t.} \quad & \sum_s P_j^s t_s \leq E \qquad \forall j = 1, \ldots N \qquad (6) \\ & t_s \geq 0. \end{aligned}$$

Note that further constraints could be easily dealt with, such as the maximum number of sub-nets we want to use and a minimal time an activated sub-net must be used.

This problem is a classical LP problem, solved with standard simplex algorithm. As stated before, columns P^s represent network configurations, that is sub-sets of nodes playing defined roles to which a given power requirement is associated. They are generated on a as-needed basis by the sub-net generation sub-problem described in the next section. Let π_j be the dual variable (shadow price) associated to the energy budget constraint (6) for node j. This is used to define the cost objective for sub-net generation. Intuitively, a large shadow price for a node implies that using the corresponding node is costly.

B. The sub-net generation sub-problem

In this sub-problem, denoted by GEN, we do not consider energy limitations directly; the energy of each node is priced by the dual variables from the master problem (MP). The objective here is to cover the set of points with minimum cost, subject to quality constraints. Thus the problem is a modification of model (L1) of Sec. III.

$$\begin{aligned} \text{(GEN)} \quad \min \quad & \sum_j \pi_j \left[\sum_{i \in C_j} E^C d_i x_{ij} + \sum_{k \in R_j} E_{jk}^T f_{jk} \right. \\ & \left. + \sum_{k \in R_j} E^R f_{kj} \right] + \sum_{j \in R_G} \pi_j E_{jG}^T w_j \qquad (7) \\ \text{s.t.} \quad & (1), (2), (3), (5). \end{aligned}$$

Given an optimal solution to this sub-problem, i.e., x_{ij}^*, f_{kj}^*, w_j^*, we compute the power requirement for each node as,

$$P_j^s = \sum_{i \in C_j} E^C d_i x_{ij}^* + \sum_{k \in R_j} E_{jk}^T f_{jk}^* + \sum_{k \in R_j} E^R f_{kj}^* + E_{jG}^T w_j^*$$

which is the information needed by the master problem (MP).

The generation sub-problem is a possibly hard MILP model. We solve the sub-net generation problem at optimality by standard branch-and-bound; however, from a practical point of view, we should note that it can be solved sub-optimally by introducing some sub-optimality tolerance in a

 0-7803-8794-5/04/$20.00

standard branch and bound algorithm. Actually, this could be useful during the first iterations, to enrich the initial set of sub-nets as quickly as possible to get good shadow prices. It is important to note that many variations may be accommodated within this column generation framework: in some applications, even using constraint-based search within column generation has been proposed [7].

C. Initializing the column set and stopping criteria

To start-up the column generation process, an initial set of columns is required, to be able to solve the master problem (MP) once and obtain the first set of dual variables. It is also important to start with a good set of columns. Solving the basic problem (L1) is a way to get one initial column. Partitioning the set of sensors according to [1] and solving the basic problem with the covering variables setting accordingly provides another set of columns. We have used this second approach as it proved to be computationally faster and it allows us to get a set of initial columns, and not only one. Considering the dual of the master problem, it can be demonstrated [5] that we should go on generating sub-nets until we get an objective value less than one from the generation sub-problem. In this case, it is impossible to find a new column that, added to the master problem, increases the network lifetime. We have observed that sometimes the last columns which are generated do not contribute significantly to the increase of network lifetime. In such cases, a possible alternative (that we did not use in this work) is to stop the master/sub-problem iterations when the maximum lifetime is not increased significantly.

V. A GREEDY APPROACH

The general idea of the distributed approach is always based on the fact that we can use the high spatial redundancy in sensor nodes by making active a small sub-set of nodes in a given sub-area, thus exploiting sensor activities in different period of time. In other words, only a sub-set of sensors is active for a given period of time, named scheduling period, whereas all other sensors are in inactive state, saving energy for future scheduling period.

The proposed approach consists of the following steps: 1) form a proper (i.e., covering) sub-set of sensors that will be switched on (active sensors); if the sub-set does not guarantee the required level of coverage, discard this instance and repeat step 1; 2) put all other sensors in off state (inactive sensors); 3) determine a suitable minimum cost routing to transfer the sensed information from all active sensors to the gateway node; if it is not possible with the selected sub-set of sensors to guarantee full connectivity, i.e., all active sensor are not able to communicate, possibly via multi-hop transmission, with the gateway node, the sub-set is discarded and the process is restarted from step 1; 4) determine the scheduling period duration, i.e., the amount of time for which this sensor sub-net lasts, while guaranteeing the target level of coverage; 5) compute sensors power consumption, subtract it from each sensor power budget, and eventually consider some of the sensors as unavailable in the future due to energy depletion; 6) iterate through this process until no other sub-set of covering and fully connected sensors can be found.

Let us now describe the above steps with more details. For what concerns the selection of the sensor sub-set (step 1), we simply allow each sensor to decide independently with a given probability (equal for all sensor nodes) whether to be in the off (inactive) or on (active) state in the current scheduling period. We check whether the selected sub-set is feasible, i.e., able to guarantee the required coverage. If infeasible, this sub-set is discarded at no cost and the algorithm restarts. For each point to be covered in the area of interest, only one randomly chosen sensor among the currently active sensors is really activated, whereas all other sensors are put in inactive state (step 2).

If the sub-set is feasible, we solve the routing problem (step 3) among sensors building a tree routed in the gateway, obviously taking into account sensors transmission capabilities. Standard techniques to determine a tree on an unknown topology (e.g., spanning tree) could be used to solve this problem. However, we implemented a simpler sub-optimal algorithm to determine paths among sensor nodes. Each node is assumed to know the reachable and active sensors closer to the gateway; this could be simply implemented by periodically broadcasting node identity and distance from the gateway. Each node randomly selects one among the available sensors closer to the gateway, to send the sensed information in multi-hop fashion to the gateway node. All sensors not involved in either sensing or routing operations are put in inactive state. If some sensor is not able to find a neighbor, the sub-set is not connected and the sub-set is discarded.

Finally, the duration of the scheduling period is determined by the sensor that exhausts first its residual power budget (steps 4 and 5).

This process is iterated until no more feasible sub-sets can be found after a fixed, large, maximum number of iterations was run (step 6).

Note that, if we neglect any cost in creating and discarding a sub-set, it is always convenient to have all sensors potentially active in a scheduling period, thus setting the probability of being active equal to 1. However, this parameter is important in realistic implementations to reduce the cost of exchanging information among nodes to determine the feasibility of a selected sub-set.

This algorithm is largely sub-optimal; however, it may be reasonably implemented in a distributed fashion and shows good performance in the considered scenarios.

VI. Testing scenario and numerical results

The column generation scheme was implemented using the OPL STUDIO/CPLEX optimization modeling system, whereas the greedy approach was implemented in MATLAB. To assess the merits of the proposed algorithms, various instance classes, were considered, whose parameters are reported in Table I. Each class is characterized by a number of sensors, a number of points to be monitored and a maximum sensing range that is assumed to be equal to the maximum transmission range of sensor nodes.

TABLE I

Instance classes

Class	Sensors	Points	Range
1	25	5	2.5
2	25	5	3
3	25	10	2.5
4	25	10	3
5	50	5	2.5
6	50	5	3
7	50	10	2.5
8	50	10	3
9	100	5	2.5
10	100	5	3
11	100	10	2.5
12	100	10	3

For each class, the same 10 problem instances were generated for the two approaches according to the following specifications:

- the area to be covered is a square with side $Q = 10$;
- the gateway node is located at the center (coordinates (5,5)) of the covered area;
- sensors are uniformly distributed on the area;
- to position the points of interest in such a way to cover the area as uniformly as possible we have used Halton low-discrepancy sequences; low-discrepancy sequences are the basis of quasi-Monte Carlo integration methods; basically, they are just a way to cover unit hypercubes (in our bi-dimensional case, a square) as uniformly as possible by a deterministic sequence, rather than by random sampling (see, e.g., [8, chapter 4]);
- minimal required coverage: 100%;
- initial energy endowment E: 0.75Ah*3.3V=8910 J
- energy required to transmit a compressed image [mJ], as a function of distance d: $5.0 + 0.01 * d^2$
- energy required to receive a compressed image: 5.0 mJ;
- energy required to compress an image: 3.6 mJ;
- sampling rate and transmission interval: one image every 15 s.

Note that we use the same radio model used in [9]: the radio dissipates 50 nJ/bit in the transmitter circuitry, 50 nJ/bit in the receiver circuitry, and 100 pJ/bit/m^2 in the transmitter amplifier. The compressed image size is of 12672 bytes; the compression energy cost is derived by assuming that a JPEG-based scheme with compression ratio 2:1 is executed on an Intel StrongARM 1110 @ 59 MHz [10].

Using these instances, we tested both the column generation scheme and the greedy algorithm; the results were averaged over 10 instances for each class.

TABLE II

Column generation approach solution

Class	Generated nets	Tot. LT [days]	Used nets
1	12.1	147.48	4.2
2	15.2	160.71	4.4
3	13.3	67.82	4.3
4	15.0	89.55	5.5
5	58.7	318.18	13.4
6	85.6	470.27	14.8
7	64.3	143.09	13.9
8	125.7	236.43	17.2
9	190.4	681.09	30.0
10	282.0	974.84	32.0
11	248.6	331.72	39.2
12	314.9	485.11	34.3

TABLE III

Greedy solution

Class	Tot. LT [days]	Used nets
1	124.10	2.4
2	141.90	2.5
3	67.10	2.5
4	81.90	2.6
5	237.10	7.2
6	412.90	13.7
7	122.00	6.7
8	212.80	12.1
9	637.10	24.3
10	946.10	34.3
11	330.70	24.2
12	482.30	33.9

Tables II and III report, for each class of instances, the average total lifetime (LT) of the system achieved by sequentially using different networks and the average number of networks employed to monitor the area under control. The results in Table II refer to the optimal solution, while the results in Table III have been obtained through the greedy approach. Table II also presents the average number of networks generated to reach the optimal solution; this metric is not reported when the greedy approach is employed since, in this case, all generated networks that meet the coverage and connectivity constraints are used to build the final solution. Tables IV and V present the characteristics of the networks used in the optimal and greedy solution, respectively. The second column contains the average lifetime of each used network, while the third indicates how many sensors are active, on average, in each of these networks. Tables VI and VII report the performance of the optimal and greedy solution, respectively, in terms of maximum and mean value of power consumed (PW) by a single sensor.

TABLE IV

NETWORKS USED IN THE OPTIMAL SOLUTION

Class	Ave. LT [days]	Ave. nodes' no.
1	44.11	4.55
2	47.89	4.31
3	16.44	7.96
4	20.66	7.13
5	28.33	5.52
6	32.67	4.17
7	11.56	8.46
8	14.61	7.59
9	24.14	4.83
10	32.31	4.27
11	8.54	8.60
12	14.74	7.73

TABLE V

NETWORKS USED IN THE GREEDY SOLUTION

Class	Ave. LT [days]	Ave. nodes' no.
1	66.40	8.05
2	70.64	7.90
3	28.64	14.06
4	37.52	11.71
5	31.39	10.60
6	38.69	9.51
7	18.03	17.33
8	23.47	15.07
9	26.88	11.36
10	28.12	9.63
11	13.91	19.10
12	14.35	17.33

First, let us consider system performance when we fix both the number of sensors and the number of points of interest, and vary the sensor sensing/transmission range (i.e., compare classes 1 and 2, 3 and 4, etc.). From Tables II and III, we observe that the total system lifetime, as well as the number of used networks, increase significantly. Indeed, the number of sensors that can ensure the required coverage or successfully route data toward the gateway node grows, thus increasing the number of networks that can be employed. This also implies that the average number of sensors included in a single network decreases for larger values of the sensing/transmission range (see Tables IV and V). As for the sensor power consumption reported in Tables VI and VII, we observe a reduction in both the average maximum and the mean value as a larger sensing/transmission range is considered. This behavior can be explained as follows.

TABLE VI

SENSORS' POWER CONSUMPTION IN THE OPTIMAL SOLUTION

Class	Ave. max PW [W]	Ave. mean PW [W]
1	1.380	0.865
2	1.204	0.681
3	2.655	1.207
4	2.213	1.097
5	1.528	0.850
6	1.150	0.754
7	2.628	1.025
8	1.914	0.895
9	1.216	0.745
10	1.083	0.744
11	2.645	0.963
12	2.191	0.960

TABLE VII

SENSORS' POWER CONSUMPTION IN THE GREEDY SOLUTION

Class	Ave. max PW [W]	Ave. mean PW [W]
1	1.305	0.774
2	1.196	0.737
3	2.648	1.011
4	2.131	0.948
5	1.489	0.770
6	1.423	0.732
7	2.739	1.069
8	2.665	1.009
9	1.418	0.767
10	1.230	0.723
11	2.825	0.970
12	2.424	0.913

Recall that the contribution to power consumption due to the output transmit power is negligible, while the most relevant contribution is due to the transceiver, in transmission mode as well as in receive mode. By increasing the sensor range, the route length from the point of interest to the gateway node becomes shorter; this implies that less relay nodes will be involved, i.e., less nodes would experience both the transmission and reception cost.

Next, assume the sensor range and the number of sensors to be fixed. Comparing classes 1 and 3, 2 and 4, 5 and 7, and so on, in Tables II–VII, we can analyze system performance for two different values of the number of points of interest (namely, 5 and 10). As expected, the number of points to be monitored has a great impact on the system lifetime and power consumption. In particular, a large number of points leads to an increase in the number of sensors needed in each network and, hence, in the mean power consumption; as a consequence, the total system lifetime decreases. Moreover, in each single network, it is more likely that a node has to gather data from more than one point, or route a larger amount of data to the gateway node. It follows that the maximum value of power consumption experienced by the sensors grows and, thus, the average network lifetime decreases. Note that the number of used networks, as well as the number of generated networks, always increase when the optimal solution is applied; while, no significant variation is observed in the case of the greedy solution. Indeed, as the number of points to be controlled grows, it becomes harder to find an optimal solution, thus more networks are generated and tried out. Instead, in the case of the greedy solution, this phenomenon does not arise since the generated networks are used, if feasible, until the first network node exhausts its energy resources.

Given the sensor range and the number of points of interest, consider the system performance as the number of sensors changes from 25 to 100 (i.e., in Tables II–VII compare classes 1,5,9, or 2,6,10, and so on). As expected, the total lifetime and the number of used networks increase as the number of sensors grows (see Tables II and III). In fact, when a larger number of sensors are available, a greater number of feasible networks can be found. Moreover, as the nodes exhaust their energy resources, further configurations that meet the constraints on connectivity and coverage are formed by using more nodes per single network. This is confirmed by the values of average number of active sensors presented in Tables IV and V. The fact that the number of used networks significantly increases with the increase in the number of available nodes justifies the reduction in the average network lifetime. Indeed, most of the networks that are created as the nodes start exhausting their energy have a short lifetime, which clearly impacts the average lifetime. As for the average power consumption, we can see from Tables VI and VII that increasing the number of sensors implies a lower mean power consumption per sensor. Indeed, a higher redundancy in the available nodes allows for more power-efficient networks. Also, note that in Table II the difference between the number of generated networks and the number of used networks increases while increasing the number of available sensors. For instance classes from 1 to 4, the number of networks used in the optimal solution is about $1/3$ of the total number of generated networks, while, for instance classes from 9 to 12, the ratio decreases down to about $1/7$. This is due to the fact that, for a large value of the number of nodes, the number of feasible networks increases significantly; however the number of "good" network configurations is limited by the given placement of the points under control.

Finally, with the column generation approach longer network lifetimes are experienced. This is not a surprise, given the larger complexity of the approach. The greedy approach provides network lifetime values closer to those obtained with the column generation approach when the network size is larger in terms of sensor nodes. This is rather encouraging, since the column generation approach can be barely used with networks with more than hundredth of sensors due to its computational complexity. The differences among the two approaches tend also to vanish when the number of points to be covered is too close to the number of sensors (classes 3 and 4), since in this scenario the redundancy is so small that it becomes difficult to optimize performance.

VII. Conclusions and Possible Extensions

Two approaches to extend system lifetime in sensor networks were presented, the more complex one being based on mathematical programming techniques, the simpler one on a greedy algorithm. Both approaches exploit the high spatial redundancy in sensor nodes: only a proper sub-set of sensors is active for a given period of time, whereas all other sensors save energy being in inactive state. Performance analysis allowed us to obtain important insight on sensor network design, as well as to determine the properties of the two algorithms. Although the proposed mathematical programming approach has the drawback of being centralized and fairly complex to solve, it is very flexible and can be easily generalized to deal with sensor' failures and different quality of service requirements. Moreover, it provides a useful term of comparison for other distributed, heuristic design schemes such as the greedy algorithm presented in this paper.

Acknowledgments

We acknowledge the support of the CNR Grant CNRC00BE35, Coordinated Project *Stochastic programming models for design and configuration of telecommunication networks* and of the MIUR (Italian Ministry of the University) project PRIMO, a project on reconfigurable platforms for wideband wireless communications.

References

[1] S. Slijepcevic, M. Poktonjak. 2001. Power Efficient Organization of Wireless Sensor Networks. *IEEE ICC 2001*, Vol. 2, pp. 472–476.
[2] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie. 2000. Protocols for the Self-Organization of a Wireless Sensor Network. *IEEE Personal Communications*, Vol. 7, pp. 16–27.
[3] M. Bhardwaj, A.P. Chandrakasan. 2002. Bounding the Lifetime of Sensor Networks via Optimal Role Assignments. *IEEE INFOCOM 2002*, Vol. 3, pp. 1587–1596.
[4] J.-H. Chang, L. Tassiulas. 2000. Energy Conserving Routing in Wireless ad-hoc Networks. *IEEE INFOCOM 2000*, Vol. 1, pp. 22–31.
[5] L.F. Wolsey. 1998. *Integer Programming*. Wiley, Chichester.
[6] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance. 1998. Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research*, Vol. 46, pp. 316–329.
[7] I.J. Lustig, J.-F. Puget. 2001. Program Does not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming. *Interfaces*, **31**, pp. 29–53.
[8] P. Brandimarte. 2001. *Numerical Methods in Finance: a MATLAB-Based Introduction*. Wiley, New York.
[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. 2000. Energy-efficient Communication Protocol for Wireless Microsensor Networks, *Hawaii Conf. on System Science*.
[10] C.-F. Chiasserini and E. Magli. 2004. Energy-efficient Coding and Error Control for Wireless Video-surveillance Networks, *Telecommunications Systems*, Special Issue on Wireless Sensor Networks, Vol. 26, No. 2-4, pp. 369-87.