The final authenticated version is available online at https://dx.doi.org/10.1007/978-3-642-21527-8_68

Exact and Metaheuristic Approaches to Extend Lifetime and Maintain Connectivity in Wireless Sensors Networks

Andrea Raiconi and Monica Gentili

Abstract Wireless sensor networks involve a large area of real-world contexts, such as national security, military and environmental control applications, traffic monitoring, among others. These applications generally consider the use of a large number of low-cost sensing devices to monitor the activities occurring in a certain set of target locations. One of the most important issue that is considered in this context is maximizing network lifetime, that is the amount of time in which this monitoring activity can be performed by opportunely switching the sensors from active to sleep mode. Indeed, the lifetime of the network can be maximized by individuating subset of sensors (i.e., *covers*) and switching among them. Two important aspects need to be taken into account among others: (i) coverage: each determined cover has to cover the entire set of targets, and (ii) connectivity: each cover should provide satisfactory network connectivity so that sensors can communicate for data gathering or data fusion (*connected* covers). In this paper we consider the problem of determining the maximum network lifetime to monitor all the targets by means of connected covers. We analyze the problem and propose an exact approach based on column generation and two heuristic approaches, namely a greedy algorithm and a GRASP algorithm, to solve it. We analyze the performance of the heuristic approaches by comparing the obtained solutions with those provided by the exact approach when available. Our preliminary experimental results show the proposed solution algorithms to be promising in terms of trade-off between solution quality and computational effort.

Andrea Raiconi
University of Salerno,Via Ponte Don Melillo, 84084 Fisciano (SA) Italy e-mail: araiconi@unisa.it

Monica Gentili
University of Salerno,Via Ponte Don Melillo, 84084 Fisciano (SA) Italy e-mail: mgentili@unisa.it

1 Introduction

Wireless sensor networks have met a growing interest in the last years due to their applicability to a large class of contexts, such as traffic control, military and environmental applications. These networks are generally characterized by a large number of small sensing devices (*sensors*), often randomly disposed all over the region of interest in order to perform a monitoring activity on a set of target points. One of the key issues in this scenario involves the maximization of the amount of time during which this activity can be carried out taking into account the limited power of the battery of each sensor. This problem is usually known as *Maximum Network Lifetime Problem (MLP)*. The lifetime of the network can be maximized by individuating subsets of sensors (i.e., *covers*) and switching among them. Two important aspects need to be taken into account among others: (i) coverage: each determined cover has to cover the entire set of targets, and (ii) connectivity: each cover should provide satisfactory network connectivity so that sensors can communicate for data gathering or data fusion (*connected* covers). The MLP is extensively studied in the literature by mainly considering the coverage requirement: exact and heuristic methods to build the set of covers and to assign them appropriate activation times have been presented in several works ([2],[3],[10]), and also some variants of the problem solved in a similar way have been proposed ([4], [7],[11]).
In this paper we consider the problem of determining the maximum network lifetime to monitor all the targets by means of connected covers. To take into account communication between sensors, we consider an additional node (*root*) representing a central processing station and we assume there exists a *communication link* among each couple of sensors (or sensor and root) if they are close enough to communicate directly between each other. A solution composed by connected covers is such that the sensors in each cover are connected to the root through a path that uses communication links.

The connectivity requirement has also been addressed in the literature ([1], [6], [9], [12], [13], [14]). In [12] and [13] the authors propose sufficient conditions in the coverage level of each sensor to imply connectivity. The connectivity issue is also considered in [9] for an unreliable wireless sensor grid network, and a necessary and sufficient condition for coverage and connectivity is presented. In [6], an approximated algorithm is presented to maximize the lifetime of the network where it is assumed that each sensor only needs to know the distances between adjacent nodes in its transmission range and their sensing radii. Existing papers that are nearer to our study are [1] and [14]. The authors in [14] describe an energy consumption model of the sensors that takes into account their different role (either a relay role, a source role or both) in the cover. They define the problem as the Maximum Tree Cover problem and present a greedy heuristic and an approximation algorithm to solve it. Two solution approaches are presented in [1] to maximize network lifetime and maintain connectivity: an exact approach based on column generation and a heuristic algorithm aiming at a distributed implementation. To our knowledge, there is a lack of contributions regarding metaheuristic approaches related to the problem.

In this paper we present a GRASP metaheuristic, as well as a greedy heuristic (that is used as a subroutine inside the GRASP) and an exact method based on column generation that is used to evaluate the performance of our heuristics. Our column generation differs from the one proposed in [1] in the subproblem definition as it will be better clarified in Section 3. We evaluate the performance of our algorithms by comparing them on some benchmark instances. Our preliminary experimental results show the proposed solution algorithms to be promising in terms of trade-off between solution quality and computational effort.

The paper is organized as follows. Next section gives the needed notation, introduces the problem and contains the mathematical formulation. The Column Generation approach is presented in Section 3. Our greedy algorithm and GRASP algorithm are object of Section 4. Experimental results are shown in Section 5. Further research is discussed in Section 6.

2 Notation and Problem Definition

Let $N = (T,S)$ be a wireless sensor network where T=$\{t_1,..,t_n\}$ is the set of target points and S=$\{s_0,s_1,..,s_m\}$ is the set composed by the sensor nodes and the special root node s_0. For each $s_i \in S$, let $T_{s_i} \subseteq T$ be the subset of targets covered by s_i; we assume $T_{s_0} \equiv \emptyset$ since the root node does not have coverage purposes. Given a subset $C \subseteq S$, we define the set of targets covered by C as $T_C = \bigcup_{s_i \in C} T_{s_i}$. By extension, each target in T_C is said to be covered by C. Moreover, for each $s_i \in S$, let $neigh(s_i) \subseteq S$ be the set of elements of S that are close enough to s_i to allow direct communication. We consider an energy consumption model of the sensors where it is assumed that the energy requirement to send data to each communication link is constant. Note that if $s_i \in neigh(s_j)$, then $s_j \in neigh(s_i)$. Given a subset $C \subseteq S$, we define $neigh(C) = \bigcup_{s_i \in C} neigh(s_i)$. Now, consider the induced undirected graph $G = (S,E)$ such that the communication link $(s_i,s_j) \in E$ if and only if $s_j \in neigh(s_i)$ $\forall s_i, s_j \in S$ (see Figure 1A for an example). We define a set $C \subseteq S$ to be a connected cover of N if: (i) $s_0 \in C$, (ii) $T_C \equiv T$, and (iii) C is such that there exists a path of communication links connecting each of its sensors to s_0 (see Figure 1B-1C).

The **Connected Maximum Network Lifetime Problem (CMLP)** is defined as follows: *Find a collection of pairs* (C_j,w_j), $j = 1,2,\dots,p$, *where* C_j *is a connected cover and* w_j *is its corresponding activation time, such that the sum of the activation times* $\sum_{j=1}^{p} w_j$ *is maximized, and, each sensor is used for a total time that does not exceed its battery:* $\sum_{j \in \{1,..p\} | s_i \in C_j} w_j \leq 1$ *for each* $s_i \in S \setminus \{s_0\}$.

The problem can be formulated as the classical MLP [2] with the only difference that each cover considered in the formulation is connected. Let $C_1,...,C_M$ be the collection of *all* the feasible connected covers in the network, let parameters a_{ij} be equal to 1 if sensor s_i belongs to cover C_j and equal to 0 otherwise, and, $w_1...,w_M$ be the decision variables denoting the activation times for each cover. The

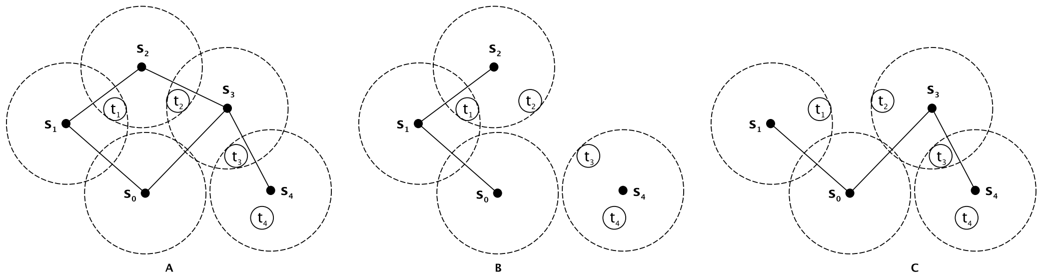

Fig. 1 **A:** An example network *N* and the corresponding induced graph *G*. **B:** Unfeasible (disconnected) cover. **C:** Feasible connected cover.

mathematical formulation of the problem is as follows:

$$\max \sum_{j=1}^{M} w_j \tag{1}$$

subject to

$$\sum_{j=1}^{M} a_{ij} w_j \leq 1 \qquad \forall s_i \in S \setminus \{s_0\} \tag{2}$$

$$0 \leq w_j \leq 1 \qquad \forall j = 1, ..., M \tag{3}$$

The objective function (1) maximizes the network lifetime. The set of constraints (2) state that the total energy consumption for each sensor cannot exceed its battery life that we normalized to be equal to 1. Finally, constrains (3) are non-negativity constraints and they define an upper bound for the activation time of each cover.

3 Column Generation Approach

Delayed Column Generation, or simply Column Generation is an efficient technique to solve linear programming formulations when the set of variables is too large to consider all of them explicitly. Since most of them will be nonbasic and, therefore, they assume a value of zero in the optimal solution, the method aims to generate only variables which have potential to improve the objective function, while the others are implicitly discarded. In particular, the general iteration of the Column Generation considers a primal problem restricted only to a subset of variables (the so-called *Restricted Primal*) and optimally solves it. In order to determine whether the returned solution is optimal for the entire problem, all the reduced costs of the nonbasic variables should be computed and it should be verified whether they satisfy the optimality conditions. If this is the case the algorithm stops, otherwise a new variable (column) is generated, it is added to the restricted primal and the algorithm iterates. To check for the optimality condition an additional problem is solved (the so-called *Separation Problem*) whose solution either returns a new column to be added to the restricted primal or verifies the optimality of the current solution.

Let us consider the mathematical formulation given in the previous section restricted only to a subset p of feasible covers, and, let B and NB be the index sets of the optimal basic and non basic variables corresponding to the optimal solution of the restricted primal, respectively. Moreover, let π_i, $i = 1, 2, \ldots, m$, be the set of dual optimal multipliers associated with each primal constraint (that is, with each sensor s_i). The current primal solution is optimal if the reduced costs associated with the non basic variables are all non negative, that is, $\sum_{i:s_i \in C_j} \pi_i - c_j \geq 0$ for each $j \in NB$, where c_j is the coefficient of variable w_j in the objective function (1) of the primal problem. Note that, all the coefficients of the variables in the objective function (1) are equal to 1, therefore the optimality conditions reduce to $\sum_{i:s_i \in C_j} \pi_i - 1 \geq 0$ for each $j \in NB$. Instead of computing all the reduced costs, we could compute the minimum among all of them and check whether it is greater than or equal to zero. Such a minimum reduced cost can be computed solving the separation problem (described next), that returns the non basic connected cover with minimum reduced cost. The proposed formulation for the separation problem ensures the connectivity of each generated cover by selecting sensors and communication links that define a tree rooted in s_0. We consider three types of variables: the binary variable act_i associated with sensor s_i that is equal to 1 if s_i is activated in the new cover, and it is equal to 0 otherwise; the binary variable x_{ij} associated with the communication link (s_i, s_j) that is equal to 1 if sensor s_i and sensor s_j are both active in the cover and the communication link (s_i, s_j) is used in the cover and it is equal to 0 otherwise; non-negative variable f_{ij} associated with each communication link (s_i, s_j) that denotes the flow passing through (s_i, s_j). Parameters of the model are the dual prices π_i associated with each sensor, and the binary parameters g_{ki} equal to 1 if target point t_k is covered by s_i and is equal to 0 otherwise.

The formulation is a single-commodity formulation to find a tree connecting all the activated sensors and covering all the targets such that the sum of the cost of the sensors in the tree is minimized. Such a tree can be found by sending from the root node s_0 a unit of flow to each active sensor.

$$\min \sum_{s_i \in S \setminus \{s_0\}} \pi_i act_i \tag{4}$$

subject to

$$\sum_{s_i \in S \setminus \{s_0\}} g_{ki} act_i \geq 1 \qquad \forall t_k \in T \qquad (5)$$

$$\sum_{(s_0,s_i) \in E} f_{0i} = \sum_{s_i \in S} act_i \qquad (6)$$

$$\sum_{(s_i,s_j) \in E} f_{ij} - \sum_{(s_j,s_i) \in E} f_{ji} = act_j \qquad \forall s_j \in S \setminus \{s_0\} \qquad (7)$$

$$\sum_{(s_i,s_j) \in E} x_{ij} = act_j \qquad \forall s_j \in S \setminus \{s_0\} \qquad (8)$$

$$x_{ij} \leq f_{ij} \leq |S| x_{ij} \qquad \forall (s_i, s_j) \in E \qquad (9)$$

$$act_i \in \{0,1\} \qquad \forall s_i \in S \setminus \{s_0\} \qquad (10)$$

$$x_{ij} \in \{0,1\} \qquad \forall (s_i, s_j) \in E \qquad (11)$$

The objective function (4) minimizes the sum of the cost of the active sensors (that is, the sensors that are selected to be in the cover). Constraints (5) ensure all the targets are covered by the selected sensors. If a sensor is selected to be in the cover then exactly one (entering) communication link has to be selected, this is ensured by constraints (8). Constraint (6) guarantee that the total amount of flow that is sent from the root node s_0 is equal to the total number of active sensors in the cover. To ensure connectivity among the active sensors, flow conservation constraints are imposed by constraints (7). Constraints (9) guarantee there is a positive flow only on communication links that are selected. Finally, constraints (10) and (11) are binary constraints on the variables.

After solving the above separation problem, if its optimal objective function value is ≥ 1, we can safely deduce that the solution found by the master problem was optimal; otherwise, the returned new column (defined by the optimal solution value of variables act_i) is introduced into the master problem that is solved again and the process is iterated.

We solved the separation problem by solver CPLEX. It is easy to check that there always exists an optimum solution composed of minimal covers, therefore we added the following set of constraints ensuring only minimal covers are generated. Let $\{C_1, C_2, \ldots, C_u\}$ be the set of connected covers generated by the algorithm so far, then the set of constraints added to the separation problem is the following:

$$\sum_{s_i \in S \setminus \{s_0\}} a_{ij} act_i \leq \sum_{s_i \in S \setminus \{s_0\}} a_{ij} - 1 \qquad \forall j = 1, ..., u \qquad (12)$$

These inequalities ensure that the new connected cover returned by the separation problem differs from the already generated covers in at least one sensor, and, therefore it is not contained in none of them. One last issue concerns the initialization of the column generation with the first initial set of connected covers. In the preliminary results presented in Section 5 we used two different approaches. For small instances, we determined an initial set of covers by running the subproblem several times associating random weights with each sensor, while, for large instances we used a heuristic initializations provided by the GRASP procedure described in Section 4.

The above column generation approach differs from that presented in [1] in the definition of the subproblem that results from a different energy consumption model associated with each sensor. The authors in [1] consider all the sensors to be connected to each other (i.e., the underlying graph is complete) but with different costs depending on the role of the sensors and on the distance between them. Such an energy consumption model could be introduced in our definition of the problem and an analysis of such a case is object of further study as outlined in the concluding section.

4 Greedy and GRASP algorithms

Here we present a greedy heuristic (namely, the CMLP-Greedy) and a GRASP metaheuristic (namely, CMPL-GRASP) designed to solve the CMLP problem.

The greedy algorithm is based on the greedy heuristic presented in [4] for the classical Maximum Network Lifetime Problem and in [7] for the α-MLP variant. The algorithm iteratively produces new covers. Each cover is initialized with the root node and new sensors, chosen among the ones with positive residual energy, are added. Each new sensor to be included is also chosen such that it is directly connected to at least one element of the cover, to guarantee connectivity. The algorithm terminates when the sensors with positive residual energy cannot guarantee either coverage or connectivity.

Algorithm 1 contains a description of our CMLP-Greedy. Line 1 contains the input parameters. Granularity factor $gf \in (0,1]$ represents a maximum amount of activation time assigned to each generated cover during the algorithm execution. The S_R set initialized in line 2 contains the list of sensors with a residual energy > 0. Parameters R_{s_i} initialized in lines 3-5 represent the amount of residual energy for each sensor s_i and the SOL set initialized in line 6 will contain the final solution. Line 7 checks whether the remaining sensors can cover the whole set of targets T. Lines 8-9 create a new cover C_l and initialize it with the root node s_0. The T_U set initialized in Line 10 contains the targets that still need to be covered in C_l. The loop in lines 11-26 ensures that new sensors are added to C_l until either all targets are covered or a new cover cannot be achieved. In particular, lines 12-13 check whether there exist sensors with residual energy > 0 that are not in C_l yet but are connected to at least one of its sensors or the root; let S_{sel} be the set of these selectable sensors. If this set is empty, it is impossible to complete C_l and the current solution SOL is returned. Otherwise, one of the sensors in S_{sel} will be selected and included in C_l. As shown in lines 16-21, if S_{sel} covers targets that are uncovered in C_l so far, a target t_c is selected as *critical* and a sensor s_c covering t_c with the *greatest contribution* is selected (we will explain such concepts in more detail in the following), otherwise if S_{sel} does not cover new targets in C_l but a sensor in S_{sel} is needed for connection then s_c is directly selected. In lines 22-25, C_l and T_U are updated according to the selection of s_c. Line 27 sets an activation time for C_l equal to gf if each sensor of C_l has a residual energy $\geq gf$, otherwise the activation time of C_l is equal to the minimum of their residual energies. Lines 28-33 update the residual energies and check if the set R_S must be updated. Cover C_l is added to SOL in line 34. The final set of covers is returned in line 36. In our experimentations, we refined the network lifetime associated with the solution returned by CMLP-Greedy by solving the mathematical model given in Section 2 restricted to the set of generated covers.

Algorithm 1 CMLP-Greedy algorithm

1: input: wireless network $N = (T, S)$, granularity factor $gf \in (0,1]$
2: $S_R \leftarrow S \setminus \{s_0\}$
3: **for** each $s_i \in S_R$ **do**
4: $R_{s_i} \leftarrow 1$
5: **end for**
6: $SOL \leftarrow \emptyset$
7: **while** $\bigcup_{s_i \in S_R} T_{s_i} \equiv T$ **do**
8: Create a new empty cover C_l
9: $C_l \leftarrow \{s_0\}$
10: $T_U \leftarrow T$
11: **while** $T_U \not\equiv \emptyset$ **do**
12: $S_{sel} \leftarrow (S_R \cap neigh(C_l)) \setminus C_l$
13: **if** $S_{sel} \equiv \emptyset$ **then**
14: **return** SOL
15: **end if**
16: **if** $T_U \cap T_{S_{sel}} \not\equiv \emptyset$ **then**
17: Find a **critical target** $t_c \in T_U \cap T_{S_{sel}}$
18: Select $s_c \in S_{sel}$ s.t. $t_c \in T_{s_c}$ and s_c has the **maximum contribution**
19: **else**
20: Select $s_c \in S_{sel}$ s.t. s_c has the **maximum contribution**
21: **end if**
22: $C_l \leftarrow C_l \cup \{s_c\}$
23: **for** each $t_j \in T_U$ s.t. $t_j \in T_{s_c}$ **do**
24: $T_U \leftarrow T_U \setminus \{t_j\}$
25: **end for**
26: **end while**
27: $w_l =$ **max feasible activation time** $\leq gf$ for C_l
28: **for** each $s_i \in C_l \setminus \{s_0\}$ **do**
29: $R_{s_i} \leftarrow R_{s_i} - w_l$
30: **if** $R_{s_i} = 0$ **then**
31: $S_R \leftarrow S_R \setminus \{s_i\}$
32: **end if**
33: **end for**
34: $SOL \leftarrow SOL \cup \{C_l\}$
35: **end while**
36: **return** SOL

***Critical Target*:** A critical target is the target that, among the ones which are covered by sensors in S_{sel}, is covered by sensors with the least positive residual energy; that is

$$t_c = \arg \min_{t_j \in T_U \cap T_{S_{sel}}} \Big\{ \sum_{s_i \in S_R | t_j \in T_{s_i}} R_{s_i} \Big\} \qquad (13)$$

***Sensor Contribution*:** If S_{sel} covers targets that are in T_U, and therefore we chose a critical target t_c, we consider the contribution of each selectable sensor s_i covering t_c as the number of uncovered targets in T_{s_i}, that is,

$$Contr(s_i) = |T_U \cap T_{s_i}| \quad \forall s_i \in S_{sel} | t_c \in T_{s_i} \tag{14}$$

Otherwise, since none of these sensors covers new targets (as in line 20 of CMLP-Greedy), the contribution of each selectable sensor s_i is given by the sum of the number of targets in T_U that can be covered by its neighbors with residual energy, i.e.

$$Contr(s_i) = \sum_{s_{i'} \in S_R \cap neigh(s_i)} |T_U \cap T_{s_{i'}}| \quad \forall s_i \in S_{sel} \tag{15}$$

If even these contributions are all equal to 0, a sensor belonging to S_{sel} is chosen randomly.

We embedded CMLP-Greedy into a GRASP scheme (CMLP-GRASP). GRASP is a metaheuristic consisting in a multi-start iterated local search. At each iteration, a new starting solution is generated according to a randomized heuristic; this solution is then refined by the local search phase. The best solution among those computed during the execution is then returned (for more detail the reader can refer to [5]). In order to implement these two steps, we developed two variants of CMLP-Greedy. CMLP-Greedy' takes as additional input parameter a positive integer rcl. Each time that a critical target or a sensor with the greatest contribution has to be selected, instead of performing the best greedy choice, we create a Restricted Candidate List (RCL) of the best *rcl* choices (or less, in case that the number of available choices is smaller than *rcl*). One element of RCL is then selected at random. In the local search phase, we build solution neighborhoods using a second variant, CMLP-Greedy". This variant imposes the inclusion of a cover (passed as input parameter) as part of the solution, while the others are generated using CMLP-Greedy.

A high level outline of CMLP-GRASP is given in Algorithm 2. Line 1 describes the input parameters. Besides the input network and the value rcl, we consider a set of granularity factor values $\{gf_1, \dots, gf_k\}$ and a maximum number of iterations it_{max} from the last improvement in the objective function value (that is in the network lifetime). In lines 2-4 we initialize the best found solution *SOL*, its value lt and the current iteration number it. In order to evaluate the objective function lt of each solution generated throughout the algorithm execution, we solve the mathematical formulation given in Section 2 restricted to the set of covers composing the solution. The GRASP loop is contained in lines 5-22. In lines 7-8 a randomized solution is generated using a predefined granularity factor and evaluated. The local search is performed as described in lines 9-16; a new neighbor is generated and evaluated for each cover in the current solution and for each granularity factor value gf_i. The best solution found and its value are stored using a steepest descent approach, as shown in lines 12-14. If the incumbent optimum was improved during a GRASP iteration, it is updated and the *it* value is reset, as can be seen in lines 17-20. Finally, the best solution found is returned in line 23.

Algorithm 2 CMLP-GRASP algorithm

1: input: wireless network $N = (T, S)$, granularity factors $\{gf_1, \ldots, gf_k\}, gf_i \in (0, 1] \forall i = 1, \ldots, k$, max iterations parameter it_{max}, RCL size $rcl > 0$
2: $it \leftarrow 0$
3: $SOL \leftarrow \emptyset$
4: $lt \leftarrow 0$
5: **while** $it < it_{max}$ **do**
6:     $it \leftarrow it + 1$
7:     $SOL' \leftarrow$CMLP-Greedy'(N, gf_1, rcl)
8:     $lt' \leftarrow$ Evaluation(SOL')
9:     **for** each $C'_l \in SOL'$ **and** each $i \in 1, \ldots, k$ **do**
10:         $SOL'' \leftarrow$CMLP-Greedy"(N, gf_i, C'_l)
11:         $lt'' \leftarrow$ Evaluation(SOL'')
12:         **if** $lt'' > lt'$ **then**
13:             $SOL' \leftarrow SOL''$
14:             $lt' \leftarrow lt''$
15:         **end if**
16:     **end for**
17:     **if** $lt' > lt$ **then**
18:         $SOL \leftarrow SOL'$
19:         $lt \leftarrow lt'$
20:         $it \leftarrow 0$
21:     **end if**
22: **end while**
23: **return** SOL

5 Computational Results

In this section we compare our CG exact procedure and our two heuristic approaches on a preliminary testbed. As in [4], we generated instances with a small population of targets and a high number of sensors. Specifically, we set $n = 15$ and $m = 25, 50, 75, 100, 150, 200$. For each value of m, we generated 5 different test instances. In each instance, targets, sensors and the root node are randomly disposed on a grid and communication links as well as target coverages are determined accordingly.

Table 1 contains average instance characteristics, in terms of connectivity among sensors and number of targets covered by each sensor. In particular, the columns labeled with Sensor Connectivity report the degree of the sensor nodes and of the root node in the induced graph. The connectivity is expressed in percentage, that is, an element of S with m communication links would have 100% connectivity. For each scenario (that is, for each value of m) we report average values of the maximum, minimum and average degree of each sensor node. For example, instances with $m = 25$ are such that the element of S with the maximum degree has on average $25 \times 0.4 = 10$ links, the one with the minimum degree has on average 1.8 links, and finally the average degree is equal to 5.4. Sensor Coverage columns give information about the number of targets covered by each sensor; that is, they give information about the quantity $|T_{s_i}|$, $s_i \in S$. In particular, for each scenario column *max* gives the average value (over the five instances) of the size of the set T_{s_i} whose cardinality is maximum, column *min* gives the average value of the size of the set T_{s_i} whose cardinality is minimum, while the last column gives the average value of the size of all the sets T_{s_i}.

Table 1 Instance characteristics (average values)

	Sensor Connectivity (%)			**Sensor Coverage**		
m	*max*	*min*	*avg*	*max*	*min*	*avg*
25	40.0	7.2	21.6	7.4	1.2	2.8
50	36.4	6.0	21.6	7.4	1.0	2.8
75	37.6	6.1	22.6	7.4	1.0	3.0
100	36.0	7.0	22.0	7.8	1.0	3.0
150	34.9	7.2	22.6	7.8	1.0	3.0
200	35.2	7.4	22.9	7.8	1.0	3.0

After a tuning phase, we determined the values to be assigned to the parameters used by the algorithms. For the CMLP-Greedy heuristic, we set gf equal to 0.2. For CMLP-GRASP, we used the set of granularity factors $\{0.2, 0.3, 0.5, 0.7\}$, and chose $rcl = 5$ and $it_{max} = 20$. We also considered a one hour time limit for each execution of the Column Generation algorithm. As already mentioned in Section 3, for larger instances (those where $m \geq 75$), we initialized the Column Generation with heuristic solutions provided by the GRASP procedure: since good starting solutions help to speed up the convergence of the method. Computational times reported for the Column Generation on these instances will be therefore the sum of the times of both the procedures. For instances with $m \leq 50$, instead, we obtained better computational times by initializing our CG with covers generated by repeated executions of the subproblem where a random weight is associated with each sensor. All tests were performed on a workstation with Intel Core 2 Duo processor at 2.4Ghz and 3GB of RAM. All the procedures have been coded in C++, using IBM ILOG CPLEX 11.2 and the Concert Technology library to solve the mathematical formulations. Detailed results and computational times can be found in Table 2, while average values are shown in Table 3.

The Column Generation algorithm is able to find certified optimal solutions for every instance with up to 75 sensors within the considered time limit; it reaches the time limit once for $m = 100$, three times for $m = 150$ and three times for $m = 200$. We run again the procedure for up to 6 hours on these instances, obtaining 4 new optimal solutions (1 for $m = 100$, 2 for $m = 150$ and 1 for $m = 200$). We reported these new optima in square brackets and bold in Table 2. For the three remaining instances, we reported naive upper bounds in round brackets and italic in Table 2. Upper bounds are computed considering the coverage level of the least covered target of the network. For $m \geq 100$, the average values reported in brackets in Table 3 are evaluated considering optimal solutions where available and upper bounds otherwise. On average, CG solutions found within the 1-hour time limit are 0.73%

worse than this value for $m = 100$, 8.57% worse for $m = 150$ and 12.45% worse for $m = 200$.

As could be easily expected, CMLP-Greedy is the fastest algorithm, computing the solutions in less than 0.1 seconds on average for instances with up to 75 sensors, and in 1.19 seconds on average for $m = 200$. It finds the optimum in 11 instances out of 30. On datasets where certified optimal solutions could be computed for each instance ($25 \leq m \leq 100$), the returned solutions are on average smaller than the optimum of a percentage between 9.76% ($m = 100$) and 14.29% ($m = 75$). While much slower than CMLP-Greedy, the CMLP-GRASP algorithm retains reasonable computational times, running on average in less than 5 minutes in the worst case ($m = 200$). It also provides better solutions, finding the optimum on 19 instances. On the datasets where we have certified optimal solutions for every instance, the returned solutions are on average smaller than the optimum of a percentage between 2.02% ($m = 50$) and 11.5% ($m = 25$).

6 Conclusions

We addressed the problem of maximizing network lifetime of a wireless sensor network when all the targets need to be covered and connectivity among sensors needs to be ensured. The problem is studied in the literature but an efficient metaheuristic algorithm still is missing. In this paper we present a GRASP metaheuristic, as well as a greedy heuristic (that is used as a subroutine inside the GRASP) and an exact method, based on column generation, that is used to evaluate their performances.

Our preliminary computational results show the proposed solution algorithms to be promising in terms of tradeoff between solution quality and computational effort. Our very next step in this research topic is a wider experimentation of the approaches aimed at evaluating their performances on different datasets (namely scattering and design datasets [8]). We also intend, in order to speed up our exact approach, to solve the subproblem by means of a heuristic approach for generating covers with positive reduced cost, and use the MIP only when the heuristic fails. Moreover, we will try to enhance the MIP formulation by improving constraints (9) by means of better upper bounds on the flows.

Additional steps will focus on applying the same approaches to solve the problem when the energy consumption of the sensors for transmission and connectivity activities is more detailed. Moreover, we will also investigate how our approaches can be modified to solve the connected variant of α-MLP [7] when the connected covers are not required to cover all the targets but only a portion α of them.

Table 2 Computational results

	CMLP-Greedy		**CMLP-GRASP**		**CG**	
m	sol	time	sol	time	sol	time
25	1	0.01	1	1.58	1	0.07
	0	0.01	0	0.13	0	0
	1	0.01	1	1.5	1	0.03
	2	0.01	2	7.93	2	0.25
	1	0.01	1	1.5	1.66	0.64
50	2	0.03	2	4.87	2	1.28
	2	0.03	2.5	6	2.5	2.49
	3	0.03	3	5.39	3	1.57
	4	0.04	4	9.99	4	3.83
	3	0.03	4	12.44	4.66	452.35
					CMLP-GRASP + CG	
75	5	0.08	6	39.4	7	2200.16
	4	0.09	4	19.78	4	23.1
	3	0.04	3	6.63	3	7.56
	6	0.13	7	42.92	7	50.89
	6	0.07	6	31.21	7	143.61
100	8	0.18	9	53.67	9.68**[10]**	TL
	6	0.37	7	53.53	7	64.28
	6	0.1	7	28.26	7	40.64
	9	0.27	9	74.71	9	88.2
	8	0.32	8	139.83	8	156.76
150	12	0.4	15	111.9	15*(20)*	TL
	11	0.54	13	108.84	13.43**[14]**	TL
	10	0.27	12	49.1	14.99**[16]**	TL
	12	1.01	13	202.98	13	297.05
	13	0.46	14	211.81	14	466.74
200	18	0.94	20	259.49	20.69*(32)*	TL
	17	1.37	18	228.29	18.32 *(20)*	TL
	13	0.44	14.95	252.2	18.52**[19]**	TL
	16	2.1	18	363.79	18	1150.66
	18	1.13	19	352.99	19	894.09

Optimal values found after the time limit are reported in square brackets and bold. Upper bounds are reported in round brackets and italic.

Table 3 Computational results (average values)

	CMLP-Greedy		**CMLP-GRASP**		**CG**	
m	sol	time	sol	time	sol	time
25	1	0.01	1	2.53	1.13	0.2
50	2.8	0.03	3.1	7.74	3.23	92.3
					CMLP-GRASP + CG	
75	4.8	0.08	5.2	27.99	5.6	485.06
100	7.4	0.25	8	70	8.14 **[8.2]**[1]	TL
150	11.6	0.54	13.4	136.93	14.08 *(15.4)*[2]	TL
200	16.4	1.19	17.99	291.35	18.91 *(21.6)*[3]	TL

[1] *1 optimum found after TL used in the average.*
[2] *2 optima found after TL and 1 upper bound used in the average.*
[3] *1 optimum found after TL and 2 upper bounds used in the average.*

References

1. Alfieri, A., Bianco, A., Brandimarte, P., Chiasserini, C.F.: Maximizing system lifetime in wireless sensor networks. European Journal of Operational Research **181**, 390–402 (2007)
2. Berman, P., Calinescu, G., Shah, C., Zelikovsky, A.: Power Efficient Monitoring Management in Sensor Networks. In: Proceedings of the Wireless Communications and Networking Conference, 2329 – 2334 (2004)
3. Cardei, M., Du, D.-Z.: Improving Wireless Sensor Network Lifetime through Power-Aware Organization. ACM Wireless Networks **11**, 333–340 (2005)
4. Cardei, M., Thai, M. T., Li, Y., Wu, W.: Energy-Efficient Target Coverage in Wireless Sensor Networks. In: Proceedings of the 24th conference of the IEEE Communications Society (INFOCOM), **3**, 1976–1984 (2005)
5. Festa, P., Resende, M.G.C.: GRASP: Basic components and enhancements. Telecommunication Systems, **46**, 253–271 (2011)
6. Kasbekar, G.S., Bejerano, Y., Sarkar, S.: Lifetime and coverage guarantees through distributed coordinate-free sensor activation. In: Proceedings of the 15th annual international conference on Mobile computing and networking, 169–180 (2009)
7. Gentili, M., Raiconi, A.: α-Coverage to Extend Network Lifetime on Wireless Sensor Networks. Technical Report **2-2010**, Department of Mathematics and Computer Science, University of Salerno (2010)
8. Lopes, L., Gentili, M., Efrat, A., Ramasubramanian, S.: Scheduling Redundant Sensors Optimally for Maximum Lifetime. Technical report **11-2010**, Department of Mathematics and Computer Science, University of Salerno (2010)
9. Shakkottai, S., Srikant, R., Shroff, N.: Unreliable sensor grids: coverage, connectivity and diameter. In: IEEE INFOCOM, 1073–1083 (2003)
10. Slijepcevic, S., Potkonjak, M.: Power Efficient Organization of Wireless Sensor Networks. IEEE International Conference on Communications **2**, 472–476 (2001)
11. Thai, M.T., Wang, F., Du, D. H., Jia, X.: Coverage Problems in Wireless Sensor Networks: Designs and Analysis. International Journal of Sensor Networks **3**, 191–200 (2008)
12. Wang, X., Xing, G., Zhang, Y., Lu,C., Pless, R., Gill, C.: Integrated coverage and connectivity configuration in wireless sensor networks. In: Proceedings of the 1st international conference on Embedded networked sensor systems, 28–39 (2003)
13. Zhang, H., Hou, J.: Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. Ad Hoc & Sensor Wireless Networks **1**
14. Zhao, Q., Gurusamy, M.: Lifetime maximization for connected target coverage in wireless sensor networks. IEEE/ACM Transactions on Networking **16**, 1378–1391 (2008)