# Maximizing lifetime for a zone monitoring problem though reduction to target coverage

F. Carrabs, R. Cerulli, C. D'Ambrosio, A. Raiconi

**Abstract** We consider a scenario in which it is necessary to monitor a geographical region of interest through a network of sensing devices. The region is divided into subregions of regular sizes (zones), such that if a sensor can even partially monitor the zone, the detected information can be considered representative of the entire subregion. The aim is to schedule the sensor active and idle states in order to maximize the lifetime of the network. We take into account two main types of scenarios. In the first one, the whole region is partitioned into zones. In the second one, a predefined number of possibly overlapping zones are randomly placed and oriented inside the region. We present a reduction technique to transform any problem instance into a target coverage one, and solve it through a highly competitive column generation-based method available in the literature.

## 1 Introduction

The issue of monitoring efficiently geographical regions through sensor networks has been intensively studied in the literature. Given the limited amount of energy provided by the battery of each device, it is indeed of great relevance to optimize their usage in order to prolong the working time (or *lifetime*) of the network for as long as possible. This is particularly relevant in vast or hardly accessible areas,

---

F. Carrabs
Department of Mathematics,University of Salerno e-mail: fcarrabs@unisa.it

R. Cerulli
Department of Mathematics,University of Salerno e-mail: raffaele@unisa.it

C. D'Ambrosio
Department of Mathematics,University of Salerno e-mail: cdambrosio@unisa.it

A. Raiconi
Department of Civil Engineering,University of Salerno e-mail: araiconi@unisa.it

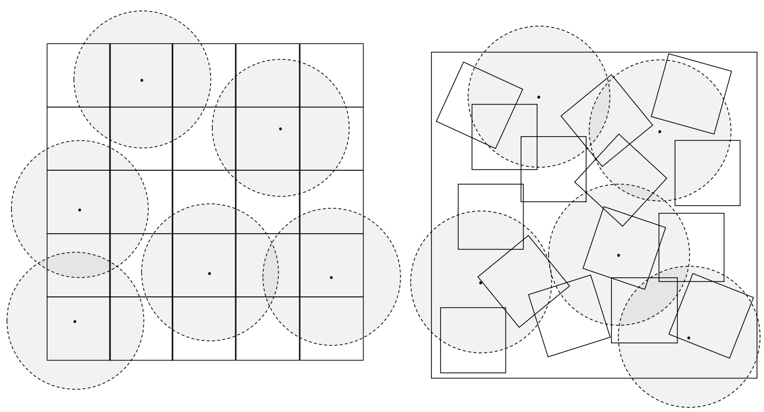
where frequent substitutions of the sensors could be not practical or impossible. In order to face this issue, many researchers have proposed approaches for the Maximum Lifetime Problem (MLP). The underlying idea is to activate at any given time only a subset of sensors, capable of performing the required monitoring task, while the others are kept idle in order to preserve their batteries. Such a subset of sensors is defined *cover*. Formally, the problem consists in finding a family of covers and in determining for how long each of them should be activated (activation time). The aim is maximize the sum of these activation times, while respecting the battery duration constraints of each sensor. The MLP problem is usually studied in terms of target coverage, meaning that we consider the existence of some special points of interest inside the area, defined *targets*. A subset of sensors is then a feasible cover if all targets fall within the sensing range of at least a sensor. The most effective resolution approaches proposed in the literature for MLP are based on the Column Generation (CG) technique. In such methods, the master problem is an LP formulation that individuates the optimal solution given a set a feasible covers, while the pricing subproblem identifies new covers that could be introduced into the set considered by the master in order to improve the incumbent solution. Given that the subproblem is NP-Hard, the main differentiating factor among such approaches is represented by the method proposed to solve the subproblem. A simple greedy heuristic is proposed in [9]. A genetic algorithm (GA) was instead proposed in [4]. To the best of our knowledge, this algorithm represents to date the most effective resolution approach for the target coverage MLP problem. CG based approaches have also been proposed to study several MLP variants with additional requirements, see for instance [1],[3],[5],[6],[7],[8],[10]. An alternate proposed definition of the problem considers area coverage, that is, the case in which we are interested to observe the entire area, rather than single points of it. Hence, the region resulting from the union of the sensing ranges for each cover should correspond to the whole area. It was however shown in [2] that any area coverage instance can be reduced to an equivalent target coverage one, by identifying in pre-processing specific targets, such that their coverage would induce coverage for the whole area.

In this work, we aim to solve MLP within a context that does not strictly correspond neither to target nor to area coverage. We start by the observing that, in an area coverage context, the requisite to cover the whole area can usually be realistically relaxed in real-world applications. Suppose that, for instance, we are interested to collect average temperatures or monitor the occurrence of fires. Within small distances, a sensor would not detect significantly different values. We can then discretize the original area into sub-areas of appropriate size (*zones*), and guarantee a partial coverage for each of them in each cover. Analogously, in a target coverage case, we can imagine that collecting information in the close proximities of the chosen target location can generally be sufficient. Again, in this case we can define a zone around each target. By relaxing in both cases the coverage requirement, such an approach can bring improvements in the network lifetime, without decreasing the quality of the solution, given appropriately chosen zone sizes. As will be shown, this new problem can be reduced to an equivalent target coverage one as well.

The specific considered scenarios are described in Section 2. In Section 3 we illustrate the method use to determine whether each target is able to cover each zone. In Section 4 we discuss the reduction to target coverage, and resume the algorithm presented in [4] that we use to solve it. Finally, computational results are presented in Section 5.

## 2 Considered test scenarios

We consider zone monitoring in the context of two different test scenarios, that we call Type 1 and Type 2, respectively. The Type 1 scenarios are meant to model area coverage. Given a square area of size $L^2$, where $L$ is the length of the side, we partition the area into $(L/l)^2$ square zones with side $l$ and area $l^2$. An example of Type 1 instance (with $L/l = 5$) is shown in Figure 1(a), along with an example of feasible cover (only active sensors are shown). As can be seen, each zone is at least partially within a sensing range, even if a relevant portion of the area is uncovered.



(a) Type 1 scenario and feasible cover (b) Type 2 scenario and feasible cover

**Fig. 1** Example instances and covers for the two considered scenario types

Type 2 scenarios model the target coverage case instead. To build these instances, we first randomly dispose a predefined number of targets within the area. Then, we consider for each target a square zone, such that the target is the center point of it. To further generalize this case, each zone in a Type 2 instance is rotated by a randomly chosen angle. Note that, differently from Type 1 instances, the zones may present overlaps, and their union does not necessarily correspond to the whole area. An example of Type 2 instance and feasible cover is shown in Figure 1(b).

## 3 Coverage detection method

## 4 Reduction to Target Coverage and CG solution approach

Using the coverage detection method illustrated in Section 3, we preprocess each input instance and determine which sensors can monitor each zone. For each sensor $s_i \in S$ and zone $z_k \in Z$, we consider a binary parameter $\delta_{ik}$ which is equal to 1 if $s_i$ can keep $z_k$ under observation, and 0 otherwise.

We note that any instance of our problem can then be reduced to an equivalent instance of the MLP problem, in which there exists a target $t_k$ for each zone $z_k$, and such that any given sensor $s_i$ covers $t_k$ if and only if $\delta_{ik} = 1$.

In order to solve the problem, we apply the highly effective CG-based exact algorithm proposed in [4]. Let $\mathscr{C} = \{C_1, \ldots, C_h\}$ be a set of feasible covers. Note that the overall number of feasible covers can be exponential, hence the aim of the CG algorithm is to find the optimal solution while avoiding to generate most of them. The master problem is defined as follows:

$$\textbf{[MP]} \max \sum_{C_j \in \mathscr{C}} w_j \tag{1}$$

s.t.

$$\sum_{C_j \in \mathscr{C}: s_i \in C_j} w_j \leq b_i \qquad \forall s_i \in S \tag{2}$$

$$w_j \geq 0 \qquad \forall C_j \in \mathscr{C} \tag{3}$$

Each $w_j$ variable models the activation time of $C_j$ in the solution. The objective function maximizes the network lifetime that can be obtained using these covers, while the constraints (2) impose that sensor battery durations are respected. In the coefficient matrix of **[MP]**, the column associated to $w_j$ represents the encoding of $C_j$. Indeed, in the position corresponding to the $i$th constraint of type (2) it contains value 1 if $s_i \in C_j$, and 0 otherwise. The aim of the pricing subproblem is to identify a new cover with potential to improve the **[MP]** objective value if introduced in $\mathscr{C}$. The subproblem can be modeled using the following ILP formulation:

$$\textbf{[SP]} \qquad \min \sum_{s_i \in S} \pi_i x_i - 1 \tag{4}$$

s.t.

$$\sum_{s_i \in S: \delta_{ik}=1} x_i \geq 1 \qquad \forall z_k \in Z \tag{5}$$

$$x_i \in \{0,1\} \qquad \forall s_i \in S \tag{6}$$

Each binary variable $x_i$ represents the choice related to the inclusion of sensor $s_i \in S$ in the new cover. The constraints (5) make sure that at least a sensor is chosen among the ones that can monitor each zone. The $\pi_i$ values are the shadow prices associated to constraints (2) after solving **[MP]**. The objective function identifies the cover with minimum reduced cost, where the constant value 1 corresponds to the coefficient of each variable in (1). If the optimal **[SP]** solution value is greater or equal than 0, the incumbent feasible found by **[MP]** is also optimal, otherwise the new cover (which is said to be *attractive*) is added to $\mathscr{C}$ and **[MP]** is solved again. The CG procedure can be iterated until a proven optimal solution is found. The **[SP]** is an NP-hard covering problem, hence the algorithm proposed in [4] also integrates a GA to solve it heuristically. In more detail, after each **[MP]** resolution, the algorithm first calls the GA. If the final population of the GA contains one or more attractive covers, they are all added to $\mathscr{C}$. Otherwise, **[SP]** is used to solve the subproblem; clearly, the exact resolution of the subproblem is needed at least once, in order to certify the optimality of the solution. In the following, we briefly resume the GA procedure; for additional details, see [4]. Each chromosome corresponds to a solution for **[SP]**, and is therefore a binary string of size $|S|$, representing the encoding of a cover. The fitness function coincides with the objective function value of **[SP]**. The GA starts from a fixed-size population $P$, composed of randomly generated chromosomes. For a predefined number of iterations $it$, two parent chromosomes are chosen through binary tournament selection, and the following operators are applied in sequence to produce a child chromosome $c$:

- Crossover: The crossover builds $c$ by applying a bitwise AND operation on $p_1$ and $p_2$. Each sensor then belongs to $c$ if and only if it belongs to both $p_1$ and $p_2$.
- Mutation: Given the chromosome $c$ obtained after applying the crossover, the mutation operator switches the value of one of its elements, chosen randomly among those that have identical value among the two parents. If the parents are completely different, a random element of $c$ is mutated.
- Feasibility: The $c$ chromosome resulting from crossover and mutation could be unfeasible, that is, some zones could be uncovered. Iteratively, the feasibility operator choses at random a sensor that can cover additional zones, and sets the related bit to 1 in $c$, until feasibility is obtained.
- Redundancy: The redundancy operator is used to remove from $c$ sensors that are not needed for feasibility. Iteratively, it computes a list of such redundant sensors, choses a random element of it and sets the related bit to 0 in $c$.

Finally, if the new produced chromosome $c$ is not already in $P$, it is introduced in the population, and replaces an older chromosome, chosen among the $|P|/2$ ones with worse fitness function. Otherwise, it is discarded. The described GA is also used to generate the starting columns belonging to $\mathscr{C}$, used to first solve **[MP]**. In this case, random values are used for the shadow prices $\pi_i$, and the entire final population $P$ is assigned to $\mathscr{C}$.

## 5 Computational Results

**Table 1** Add caption

| | Targets | Radius | Sensors | Lifetime | Time |
|---|---|---|---|---|---|
| 500x500 | 25 | 70.71 | 500 | 67.80 | 3.36 |
| | | | 750 | 102.10 | 7.72 |
| | | | 1000 | 135.70 | 13.22 |
| | 100 | 35.35 | 500 | 34.00 | 3.38 |
| | | | 750 | 53.90 | 8.39 |
| | | | 1000 | 71.90 | 17.74 |
| | 400 | 17.67 | 500 | 21.40 | 8.97 |
| | | | 750 | 31.40 | 24.91 |
| | | | 1000 | 42.10 | 48.31 |

## References


1. A. Alfieri, A. Bianco, P. Brandimarte, and C. F. Chiasserini. Maximizing system lifetime in wireless sensor networks. *European Journal of Operational Research*, 181(1):390–402, 2007.
2. P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky. Power efficient monitoring management in sensor networks. In *Proceedings of the Wireless Communications and Networking Conference*, volume 4, pages 2329 – 2334, 2004.
3. F. Carrabs, C. Cerrone, C. D'Ambrosio, and A. Raiconi. Column generation embedding carousel greedy for the maximum network lifetime problem with interference constraints. In Sforza A. and Sterle C., editors, *Optimization and Decision Science: Methodologies and Applications. ODS 2017.*, volume 217 of *Springer Proceedings in Mathematics & Statistics*, pages 151–159. Springer, Cham, 2017.
4. F. Carrabs, R. Cerulli, C. D'Ambrosio, and A. Raiconi. A hybrid exact approach for maximizing lifetime in sensor networks with complete and partial coverage constraints. *Journal of Network and Computer Applications*, 58:12–22, 2015.
5. F. Carrabs, R. Cerulli, C. D'Ambrosio, and A. Raiconi. Extending lifetime through partial coverage and roles allocation in connectivity-constrained sensor networks. *IFAC-PapersOnline*, 49(12):973–978, 2016.
6. F. Carrabs, R. Cerulli, C. D'Ambrosio, and A. Raiconi. An exact algorithm to extend lifetime through roles allocation in sensor networks with connectivity constraints. *Optimization Letters*, 11(7):1341–1356, 2017. To appear in Optimization Letters. DOI: 10.1007/s11590-016-1072-y.
7. F. Carrabs, R. Cerulli, C. D'Ambrosio, and A. Raiconi. Exact and heuristic approaches for the maximum lifetime problem in sensor networks with coverage and connectivity constraints. *RAIRO - Operations Research*, 51(3):607–625, 2017.
8. F. Castaño, E. Bourreau, N. Velasco, A. Rossi, and M. Sevaux. Exact approaches for lifetime maximization in connectivity constrained wireless multi-role sensor networks. *European Journal of Operational Research*, 241(1):28–38, 2015.
9. K. Deschinkel. A column generation based heuristic for maximum lifetime coverage in wireless sensor networks. In *SENSORCOMM 11, 5th Int. Conf. on Sensor Technologies and Applications*, volume 4, pages 209 – 214, 2011.

**Table 2** Add caption

| | Radius Quadrato | Sensors | Lifetime | Time | | Radius Quadrato | Sensors | Lifetime | Time |
|---|---|---|---|---|---|---|---|---|---|
| Area 500x500 - Targets 15 | 0 | 500 | 43.10 | 1.83 | Area 1000x1000 - Targets 15 | 0 | 500 | 27.30 | 1.32 |
| | | 750 | 58.80 | 3.71 | | | 750 | 42.10 | 3.22 |
| | | 1000 | 79.60 | 6.21 | | | 1000 | 58.50 | 5.76 |
| | 25 | 500 | 48.10 | 1.91 | | 25 | 500 | 28.70 | 1.35 |
| | | 750 | 71.30 | 4.44 | | | 750 | 42.70 | 3.20 |
| | | 1000 | 88.10 | 6.53 | | | 1000 | 59.50 | 5.67 |
| | 50 | 500 | 54.70 | 2.11 | | 50 | 500 | 29.50 | 1.24 |
| | | 750 | 84.30 | 5.13 | | | 750 | 43.10 | 2.90 |
| | | 1000 | 97.60 | 6.74 | | | 1000 | 60.80 | 5.50 |
| | 75 | 500 | 63.40 | 2.32 | | 75 | 500 | 30.00 | 1.23 |
| | | 750 | 97.40 | 5.83 | | | 750 | 46.10 | 3.06 |
| | | 1000 | 107.60 | 7.76 | | | 1000 | 63.40 | 5.47 |
| | **Radius Quadrato** | **Sensors** | **Lifetime** | **Time** | | **Radius Quadrato** | **Sensors** | **Lifetime** | **Time** |
| Area 500x500 - Targets 30 | 0 | 500 | 23.90 | 1.41 | Area 1000x1000 - Targets 30 | 0 | 500 | 12.30 | 0.80 |
| | | 750 | 40.80 | 3.91 | | | 750 | 20.50 | 1.95 |
| | | 1000 | 66.90 | 8.43 | | | 1000 | 28.10 | 3.37 |
| | 25 | 500 | 29.20 | 1.64 | | 25 | 500 | 12.90 | 0.91 |
| | | 750 | 51.70 | 4.36 | | | 750 | 21.40 | 2.06 |
| | | 1000 | 87.50 | 11.32 | | | 1000 | 30.00 | 3.49 |
| | 50 | 500 | 33.00 | 2.06 | | 50 | 500 | 14.20 | 0.83 |
| | | 750 | 65.30 | 5.29 | | | 750 | 23.30 | 1.96 |
| | | 100 | 104.80 | 11.99 | | | 1000 | 33.70 | 3.83 |
| | 75 | 1000 | 38.70 | 2.34 | | 75 | 500 | 16.50 | 1.01 |
| | | 750 | 79.50 | 7.19 | | | 750 | 25.30 | 2.24 |
| | | 1000 | 121.60 | 13.42 | | | 1000 | 36.50 | 4.03 |

10. A. Rossi, A. Singh, and M. Sevaux. Lifetime maximization in wireless directional sensor network. *European Journal of Operational Research*, 231(1):229–241, 2013.