Noname manuscript No.
(will be inserted by the editor)

Optimization of Sensor Battery Charging to Maximize Lifetime in a Wireless Sensors Network

Francesco Carrabs · Ciriaco D'Ambrosio · Andrea Raiconi

Received: date / Accepted: date

Abstract The Maximum Network Lifetime is a well known and studied optimization problem. The aim is to appropriately schedule the activation intervals of the individual sensing devices composing a wireless sensor network used for monitoring purposes, in order to keep the network operational for the longest period of time (network lifetime). In this work, we extend this problem by taking into account the issue of charging the sensor batteries. More specifically, it has to be decided how much charge should be provided to each sensor, given the existence of a charging device with limited energy availability. An exact Column Generation algorithm embedding a Genetic Algorithm for the subproblem is proposed. Computational results reveal that by appropriately choosing the charge levels, remarkable network lifetime improvements can be obtained, in particular when the available energy is scarce.

Keywords Optimal Battery Charging · Wireless Sensor Network · Column Generation

1. Introduction

Wireless Sensor Networks (WSNs) are nowadays an ubiquitous technology, with a wide set of applications (see for instance [13],[15]). While individual devices have generally limited hardware capabilities, due mostly to size and cost constraints, WSNs are used to perform complex tasks by intelligently coordinating their usage. It is therefore of paramount importance to develop

F. Carrabs, C. D'Ambrosio
Department of Mathematics, University of Salerno
E-mail: {fcarrabs,cdambrosio}@unisa.it

A. Raiconi
Department of Computer Engineering, Electrical Engineering and Applied Mathematics, University of Salerno
E-mail: araiconi@unisa.it

algorithms and methods to manage WSNs efficiently, as witnessed by the extended number of research contributions belonging to this field.

Among the considered issues, one that has generated a remarkable amount of research interest is related to the efficient scheduling of the active and idle states of sensors used to surveil, or to collect information from, some locations of interest positioned inside a geographical region (*targets*). Indeed, given a WSN deployed within such area, activating all sensors at the same time (and keeping them in active state until their batteries deplete) could not be their most efficient usage, given that they may provide a redundant level of coverage for one or more targets. If we are instead able to identify multiple, possibly overlapping subsets of sensors that can provide complete coverage (generally defined *covers*), one may keep only the sensors belonging to a cover active at any given moment of time, while the others are kept in a battery-preserving idle state. It is then easy to understand that by appropriately identifying covers and activation intervals for each of them, it is possible to significantly extend the *network lifetime*, that is, the operational time of the WSN.

The above illustrated problem, generally defined *Maximum Network Lifetime Problem* (MLP), is known to be NP-Complete (see [3]). It should be noted that the *area coverage* case, in which the whole area rather than specific targets must be kept under observation, can be reduced to the target coverage one (see [2],[21]).

A successful line of research for MLP has proposed exact resolution methods based on Column Generation (CG). In this type of approaches, the master problem is a linear programming formulation that, given a subset of all the feasible covers, determines the activation times that lead to the maximum lifetime. The other component is a subproblem that, given the dual prices derived from a master problem optimal solution, is able to identify a new *attractive* cover, that is a cover with potential to improve such solution. Resolution approaches of this type for MLP and several variants of the problem have been proposed, for instance in [8] the authors studied the case in which some sensors may interfere with each other. In [4,12] a predefined number of the targets is allowed to be not monitored, in [6,10] the authors considered connectivity issues, in order to route the collected information to a central processing unit, and in [5,7,9,11] both previous issues are considered together.

Recent advances in technology have raised interest in wireless networks in which the batteries can be recharged. In [17] the authors develop and study a hardware prototype of a WSN system that makes use of a mobile robot equipped with a charger and a base station used to monitor the battery status of the sensors. In [19], the authors compare three scheduling protocols for sensors that can recharge their batteries through solar panels, used for purposes such as autonomous organization, tracking of moving targets and avoiding threats in the case of moving sensors. The authors of [22] consider a moving charging device that periodically visits all the sensors of a network, aiming at maximizing the ratio between the time spent at the home station and the time spent traveling by this device. The authors show that the optimal path is the shortest Hamiltonian Cycle, prove several properties and find near-optimal solutions through piece-wise linear approximation. Again in the context of wireless networks recharged by a mobile device, different protocols are proposed by [1] and [18].

In [20] the authors present a multi-objective problem related to the detection of multiple events by a sensor network with devices that can harvest energy from the environment. The problem is transformed into a linear programming one through a weighted sum method, and the authors show how to find Pareto optimal solutions.

A scenario closer to the MLP problem is considered by [23]. Indeed, in this work, the authors consider a scenario in which rechargeable sensors are used for target coverage. The time horizon is discretized into time slots and each sensor is assumed to be able to harvest energy at a given rate when it is not used for sensing. The authors present an LP formulation in which the final time slot T is a parameter, and the objective is to minimize the overall energy consumption while keeping all targets under observation, by determining the activation schedule of each individual sensor. They propose to find a suitable T value through binary search. Since the method is considered computationally expensive, the authors also propose a heuristic algorithm.

As mentioned, several works consider the existence of a high-capacity mobile charger able to recharge a WSN (see [1,17,18,22]). However, to the best of our knowledge, the MLP problem has not been studied in such a context. Indeed, optimizing the charge level for each different sensor depending on its position can be crucial to prolong the network lifetime, in particular if only a limited amount of energy is available.

For this reason, in this work we propose and study such a variant, that we call *Maximum Network Lifetime Problem with Chargeable Sensors* (MLP-C). In more detail, we consider a scenario in which a charging device with limited capacity is available, and the issue of deciding the charge level for each sensor of the network must be faced along with the individuation of the covers and their activation times. We propose an exact resolution approach based on CG. Since the subproblem is NP-Hard, we use a Genetic Algorithm (GA) for its resolution, and solve the subproblem using an ad-hoc ILP formulation only when the GA fails to identify attractive covers.

The proposed scenario is of particular relevance in cases in which we intend to reuse an existing network of sensors and, due to environmental conditions, it is difficult or not possible to keep them continually powered or replace their batteries. In more detail, a relevant application scenario is the one of underwater wireless rechargeable sensor networks, analyzed for instance in [14]. As described in this work, a promising way to replenish sensor batteries in this context is to send in their proximity charging moving devices, that the authors define *mules*. These devices can use wireless power transfer technologies, such as inductive charging (see also [16]) to transfer energy to sensor nodes. Our work can be considered complementary to [14], which focuses on the determination of routes for the charging devices that minimize their traveling costs.

In our computational tests, we will focus on the case in which the sensor batteries are empty before the intervention of the charging device. However, the proposed method is general enough to be applicable to scenarios in which a residual charge exists in one or more sensors. The main aim of the considered test case is to highlight a comparison with the classical scenario, in which each battery has a predefined battery charge. Indeed, it will be shown that, assuming to have an equal total amount of available energy for the two scenarios, MLP-C allows to obtain very significant extensions of the network lifetime, in particular when the available energy is scarce.

The rest of the paper is organized as follows. The problem is formally defined in Section 2. The CG approach is discussed in Section 3. An evaluation of an upper bound on the maximum theoretical improvement obtainable by our approach with respect to the classical one is provided in Section 4. Computational results are presented and commented in Section 5. Section 6 contains our final remarks.

2. Definitions and Notations

Let $S = \{s_1, \ldots, s_n\}$ be a network of sensing devices. We assume all devices to have equal hardware components, and therefore the same battery capacity b. As in [4] and other works, without loss of generality, we normalize batteries capacity to 1 time unit ($b = 1$). Let r_i be the battery charge of sensor s_i before the intervention of the charging device ($0 \leq r_i \leq b$).

Furthermore, let $T = \{t_1, \ldots, t_m\}$ be the set of targets to be kept under observation. We define a binary parameter δ_{ij} $\forall\ s_i \in S$, $t_j \in T$ which is equal to 1 if t_j is located within the sensing range of s_i, and 0 otherwise.

A subset of sensors $C \subseteq S$ is defined to be a cover if it can keep all targets under observation, that is $\sum_{s_i \in C} \delta_{ij} \geq 1\ \forall\ t_j \in T$. Clearly, in order to be able to monitor all targets, S must be a cover. Furthermore, any proper subset of S can potentially be a cover as well.

Finally, let us assume to have an available charging device with a capacity equal to $\mathcal{R}_a > 0$ time units, and let $\mathcal{C}$ be the subset of all covers. Assuming all sensor batteries to be empty before the intervention of the charging device, MLP-C corresponds to the following Linear Programming formulation:

$$[\mathbf{P}] \max \sum_{C_j \in \mathcal{C}} w_j \tag{1}$$

$$s.t.$$

$$\sum_{C_j \in \mathcal{C}} a_{ij} w_j - y_i \leq r_i \qquad \forall s_i \in S \tag{2}$$

$$\sum_{s_i \in S} y_i \leq \mathcal{R}_a \tag{3}$$

$$y_i \leq 1 - r_i \qquad \forall s_i \in S \tag{4}$$

$$w_j \geq 0 \qquad \forall C_j \in \mathcal{C} \tag{5}$$

$$y_i \geq 0 \qquad \forall s_i \in S \tag{6}$$

Each binary parameter a_{ij} $\forall s_i \in S, C_j \in \mathcal{C}$ is equal to 1 if s_i belongs to C_j, and to 0 otherwise. Each w_j variable corresponds to the activation time assigned to the related cover C_j, while the y_i variables correspond to the amount of charge transferred from the charging device to the sensors. Clearly, the charge assigned to each sensor cannot exceed its battery capacity (see constraints (4)).

The objective function (1) maximizes the sum of the activation times and, therefore, the network lifetime. Constraints (2) state that the total activation time of the covers containing a given sensor s_i cannot exceed $y_i + r_i$. Finally, the constraint (3) limits the battery charge transferrable from the charging unit to sensor batteries to $\mathcal{R}_a$.

With respect to the $\mathcal{R}_a$ value, we note that the amount of energy necessary to fully charge all sensors would be $\mathcal{R} = \sum_{s_i \in S}(b - r_i) = \sum_{s_i \in S}(1 - r_i)$. But then, for $\mathcal{R}_a \geq \mathcal{R}$, no actual decision is necessary with respect to battery charging, since we may simply assume $y_i = 1 - r_i$ $\forall s_i \in S$. Indeed, in this case, the problem would reduce to the classical MLP one. However, for many real-world cases, this may not be feasible. Indeed, the constrained capacity of the mobile charger battery may not allow full charging of all the sensors in the network, in particular in the context of a WSN with a large number of sensor devices. Furthermore, due to changes in the monitoring activity to be performed, the recharge of some sensors may have limited or no utility. Indeed, their covered targets could no longer be of interest, or (recalling for instance the underwater application mentioned in Section 1) they may have changed their positions.

Given these reasons, in the following we will assume $\mathcal{R}_a < \mathcal{R}$, and therefore it becomes crucial to decide which sensors have to be recharged and how much, in order to maximize the lifetime of the network.

To highlight the importance of appropriate battery charging, consider the example in Figure 1, with $S = \{s_1, s_2, s_3, s_4\}$ and $T = \{t_1, t_2, t_3, t_4\}$. The wider circles represent sensor ranges, while the smaller ones are the targets. Furthermore, suppose that $r_1 = r_2 = r_3 = r_4$=0 and $\mathcal{R}_a = 3$.

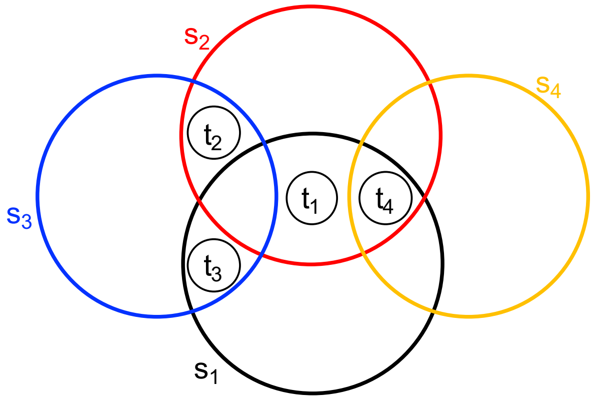

Fig. 1: Example network

By assigning the battery charges $y_1 = 1$, $y_2 = 1$, $y_3 = 1$, $y_4 = 0$, we can obtain a lifetime equal to 1.5 time units by activating in sequence the covers $\{s_1, s_2\}$, $\{s_1, s_3\}$ and $\{s_2, s_3\}$ for 0.5 time units each. If we assume to have the same overall amount of available energy and all batteries are constrained to have an equal charge, that is $y_i = 0.75\ \forall i = \{1, \dots, 4\}$, the maximum lifetime that can be achieved is 1.125, and it is obtained by activating the covers $\{s_1, s_2\}$, $\{s_1, s_3\}$ and $\{s_2, s_3\}$ for 0.375 time units each. Hence, we can note that prioritizing the most useful sensors allowed a significant lifetime improvement. A formal evaluation of the maximum solution gap between the two cases considered in the example is provided in Section 4.

Due to the potentially large number of feasible covers, it is not conceivable to use **[P]** to solve instances of the problem of non-trivial size. For this reason, a CG approach is presented in next section.

3. Column Generation algorithm

As discussed in previous section, formulation **[P]** has a potentially exponential number of variables and, consequently, of columns in the coefficient matrix. By analyzing the structure of **[P]**, we can see however that the number of y variables and, consequently, of columns associated to them, is equal to $|S|$. Moreover, the structure of these columns is known in advance. On the other hand, **[P]** contains a column associated to a w variable for each feasible cover belonging to $\mathcal{C}$. In each of these columns, the position associated to the i-th constraint of type (2) contains a 1 if sensor s_i belongs to the related cover and a 0 otherwise, while the positions related to contraints (3) and (4) contain all zeros.

We may therefore consider a restricted version of **[P]**, which includes all the y variables and some of the w ones, associated to a subset of covers $\mathcal{C}' \subseteq \mathcal{C}$.

Such a formulation, referred to as [**RP**] from now on, provides the maximum lifetime that can be obtained using only the covers included in $\mathcal{C}'$.

We use [**RP**] as restricted master problem of a CG algorithm. Given that all y variables were included *a priori* in [**RP**], such a CG approach only has to introduce new columns related to w ones. Indeed, the optimal [**RP**] solution is not optimal for [**P**] if some w_j related to a cover $C_j \notin \mathcal{C}'$ has a negative reduced cost, in which case we say that it is attractive. We can evaluate such reduced cost as $\sum_{i:s_i \in C_j} \pi_i - c_j$, where π_i are the dual prices related to constraints (2) of [**RP**], and c_j is the coefficient of w_j in the objective function of [**P**]. Indeed, given the above mentioned structure of the columns of the w variables, it is straightforward to note that the dual prices related to the other constraints are not needed in the reduced costs computation.

Since $c_j = 1$ for each w_j (see (1)), it follows that w_j is attractive if $\sum_{i:s_i \in C_j} \pi_i < 1$. We formulate therefore the subproblem [**SP**], aimed at finding the cover that minimizes the sum of the dual prices associated to the sensors that are chosen to belong to it.

$$[\mathbf{SP}]\ \min \sum_{s_i \in S} \pi_i x_i \tag{7}$$

$$s.t.$$

$$\sum_{s_i \in S} \delta_{ij} x_i \geq 1 \qquad \forall t_j \in T \tag{8}$$

$$x_i \in \{0, 1\} \qquad \forall s_i \in S \tag{9}$$

Binary variables x_i that assume value 1 correspond to sensors chosen to belong to the cover, while constraints (8) ensure that each target is monitored by at least one of them.

If the [**SP**] optimal solution is greater than or equal to 1, no attractive covers exists and therefore the optimal [**RP**] solution is also optimal for [**P**]. Otherwise, we add to [**RP**] the w_j variable associated to the cover corresponding to the optimal [**SP**] solution, and solve [**RP**] again. Note that we can easily reconstruct the w_j column corresponding to such cover. Indeed, it will contain value y_i in the position corresponding to the constraint of type (2) associated to each sensor s_i, and zero in the other positions.

The process can be iterated until [**SP**] certifies the optimality of the last solution found by the master problem. We do note, however, that [**SP**] is a set covering problem and is NP-Hard. For this reason, we introduce a stopping criterion for [**SP**]**; that is, we prematurely interrupt the ILP resolution as soon as an attractive cover is found. Furthermore,** we decided to propose a method to solve the CG subproblem heuristically, as described in next subsection.

3.1 Subproblem GA

The proposed heuristic resolution method for the subproblem is a GA, which is an adaptation of the one proposed in [4] for the classical MLP problem.

In the GA, each chromosome is a binary vector of length $|S|$, which corresponds to the encoding of a cover. In the following, given a chromosome p, let $p[i]$ refer to the gene corresponding to sensor s_i.

The fitness function of each chromosome p corresponds to the objective function value of **[SP]**, that is, it can be computed as $\sum_{s_i \in S} \pi_i p[i]$. Therefore, each chromosome with a fitness function value that is lower than 1 is the encoding of an attractive cover.

The population P has a predefined fixed size, and is composed of randomly generated chromosomes. In our computational tests, we chose $|P| = 50$. Each GA iteration generates a new child chromosome p_c by combining two parents p_1 and p_2. The parents are chosen through binary tournament; that is, to choose each of them we first extract at random two chromosomes from the current population, and then select the one with the best (i.e. lower) fitness function. The new chromosome p_c is then obtained by applying in sequence the following four operators:

- **Crossover:** Each gene $p_c[i]$ $(i = 1, \ldots, |S|)$ is set to 1 if both $p_1[i] = 1$ and $p_2[i] = 1$, and 0 otherwise.
- **Mutation:** A random gene i whose value is equal between p_1 and p_2 is chosen (that is, such that $p_1[i] = p_2[i]$), and the value of $p_c[i]$ is switched from 0 to 1 or vice versa. If p_1 and p_2 differ in all genes, a random gene of p_c is switched. The i-th gene value will be switched back only if strictly necessary for the Feasibility and Redundancy operators, described next.
- **Feasibility:** The feasibility operator makes sure that p_c actually corresponds to a cover. If the sensors encoded by genes set to 1 do not cover all targets, a random sensor s_i that can cover at least a new target is chosen, and the related gene $p_c[i]$ is set to 1. The procedure iterates until all targets are covered.
- **Redundancy:** This operator makes sure that the cover corresponding to p_c does not contain unneeded sensors. The operator builds a list containing each sensor s_i such that $p_c[i] = 1$ but could be switched to 0 without affecting feasibility. A random element of the list is chosen, and the related gene is set to 0. The list is then recomputed, and the procedure iterates until no additional sensor can be removed.

Each newly built chromosome p_c is discarded if an identical one already exists in P. Otherwise, it replaces an older chromosome, chosen at random among the $|P/2|$ worst ones in terms of fitness function.

The above described Feasibility and Redundancy operators are also used for buiding the individuals composing the initial population. Each of them is built by applying the two operators in sequence, starting from a binary vector in which all positions are set to 0, again discarding eventual duplicates.

Two stopping criteria are considered, namely a maximum number of iterations without improvements with respect to the incumbent best solution encountered, and a maximum number of consecutive identical chromosomes generated. For the computational tests reported in Section 5, values 1500 and 100 have been chosen for these two parameters, respectively.

As soon as a stopping criterion is reached, we check if the final population contains at least one chromosome corresponding to an attractive cover. If this is the case, all the newly individuated attractive covers are added at once to [**RP**], as an attempt to reduce the number of CG iterations, and the master problem is solved again. If no attractive covers were found by GA, the subproblem is solved using the [**SP**] formulation. As previously mentioned, the [**SP**] resolution is interrupted as soon as an attractive cover is found.

4. Upper bound on lifetime improvement

In this section, we aim to identify the maximum theoretical improvement that can be obtained by our proposed approach that identifies the optimal charge levels for each sensors, with respect to a scenario in which the same overall amount of energy is equally distributed among all of them. In more detail, we want to compare the maximum lifetimes achievable by the following two approaches:

- OC (Optimal Charging), our proposed CG algorithm to solve MLP-C, presented in Section 3;
- UC (Uniform Charging), an exact algorithm for MLP, in which each sensor is given a battery charge equal to $\beta = \frac{\mathcal{R}_a}{|S|}$, where $\mathcal{R}_a$ is the charging device capacity considered by OC (note that, by definition, $0 < \beta < 1$).

We assume all batteries to be empty before the uniform or optimized battery charges are given, in order to fully evaluate the advantage provided by OC with respect to UC.

The aim of presenting this upper bound is to provide additional insights on the computational results discussed in Section 5, in which OC and UC solutions are experimentally evaluated and compared.

Let $LF(A)$ be the objective function value obtained by executing a given algorithm A for a reference instance. In Section 5, we will report percentage gaps between the lifetimes obtained by the two considered approaches through the formula $GAP = 100 \times \frac{LF(OC) - LF(UC)}{LF(UC)}$.

First of all, we note that such gap can never be negative. Indeed, by definition OC can only find solutions that are greater than or equal to the ones found by UC, since the fixed battery setting considered for UC is feasible for OC.

Furthermore, let LT_{max} be the maximum lifetime value obtainable by solving to optimality the classical MLP problem, giving to all sensors a battery charge equal to 1 time unit. Since UC solves to optimality the same problem, with battery capacities scaled of a factor β, it follows that

$$LF(UC) = \beta \times LT_{max} \tag{10}$$

Finally, we observe that LT_{max} represents a trivial upper bound on the maximum lifetime achievable for the same instance through OC for any $0 < \beta < 1$, since it is the maximum lifetime that would be obtained by the algorithm for $\mathcal{R}_a = \mathcal{R}$. The following upper bound on the GAP value can then be derived:

$$100 \times \frac{LT_{max} - \beta \times LT_{max}}{\beta \times LT_{max}} = 100 \times \frac{1-\beta}{\beta} \tag{11}$$

5. Computational tests

In this section, we evaluate the effectiveness of the proposed approach on a set of benchmark instances. In more detail, we compare the OC and UC algorithms, described in the previous section. As previously mentioned, OC is an implementation of the CG algorithm described in Section 3. UC was obtained starting from the same code, and by constraining in the master problem each sensor to receive β as battery charge.

The algorithms were coded in C++ on a macOS platform, running on an Intel Core i5 3.1 GHz processor with 8 GB of RAM. The mathematical formulations within the CG algorithms were solved through IBM ILOG CPLEX 12.8. Default CPLEX parameters and single thread mode were used.

In Section 5.1 we describe the considered instances, while in Section 5.2 the obtained numerical results, along with some comments on them, are presented.

5.1 Test instances

We executed our algorithms on a dataset of randomly generated instances, using the following values:

- Number of sensors $|S|$ equal to 500, 750, 1000, 1250 or 1500;
- $r_i = 0 \; \forall s_i \in S$;
- Number of targets $|T|$ equal to 15 or 30;
- Charging device capacity $\mathcal{R}_a = \beta\mathcal{R} = \beta|S|$, with β equal to 0.25, 0.50 or 0.75.

We generated 10 different instances for each combination of values for $|S|$ and $|T|$. In more detail, we considered a square area of size 500×500, and generated at random the spatial coordinates of each sensor and target within this space. Each sensor is given a sensing range equal to 100. During the instance generation phase we also make sure that, in each instance, each target is covered by at least one sensor and each sensor covers at least one target. For each of these base 100 instances, tests were executed considering the three mentioned β values.

We recall that, in the case of UC, the β value corresponds to the predefined battery charge assigned to each sensor of the network. According to (11), the maximum percentage gap between OC and UC is therefore equal to

- 300% for $\beta = 0.25$
- 100% for $\beta = 0.50$
- $33.\overline{3}$% for $\beta = 0.75$

Clearly, as β decreases the related upper bound is likely to be looser. Indeed, achieving a 300% gap on an instance with $\beta = 0.25$ would mean that we were able to achieve the same lifetime (LT_{max}) that we would have achieved with fully charged batteries using only a quarter of the same overall energy.

5.2 Test results

In order to evaluate the contribution of the genetic algorithm, we start by comparing OC with a basic version (from now on called basicOC) in which the subproblem is always solved by using **[SP]**. As previously mentioned, in both algorithms we interrupt the **[SP]** resolution as soon as it finds an attractive column. In preliminary tests, this policy showed to significantly improve the performances of both algorithms. The results of the comparison between basicOC and OC are contained in Table 1.

Each row of the table contains average values computed on a test scenario, composed of 10 instances corresponding to the same values of β, $|S|$ and $|T|$. The first three columns report, for each of the considered scenarios, the values of these three parameters. The following four columns contain average lifetime values and computational times in seconds for basicOC and OC, respectively.

We note that the computational times of basicOC are influenced in particular by the β value. In particular, the computational time for $|S| = 1500$, $|T| = 30$ and $\beta = 0.25$ is equal to 24.01 seconds, while it grows to 153.14 seconds for the same instances and $\beta = 0.5$, and to 294.60 seconds for $\beta = 0.75$. On the other hand, OC solves 22 out of 30 scenarios within 10 seconds and requires more than 30 seconds only twice, with a peak equal to 66.12 seconds. Understandably, having a larger charging device enlarges the solution space and allows more complex solutions; in this context, the GA contribution appears to be more effective.

We now present the comparison between UC and OC algorithms. As previously mentioned, both algorithms are based on the same code, and both embed the GA algorithm for the subproblem. The results of this comparison are shown in Table 2, where the headings of the first seven columns have the same meaning that they have for Table 1, with UC instead of basicOC. The additional column contains percentage gaps between the lifetime values of the two algorithms for each scenario, computed and described in Section 4.

As discussed, gaps can never be negative. We can however observe that, in practice, OC always improved the UC solutions by a significant margin. Indeed, the gap is always equal to $33.\overline{3}$% for $\beta = 0.75$, it ranges between

β	|S|	|T|	basicOC lifetime	basicOC time	OC lifetime	OC time
0.25	500	15	22.126	1.50	22.126	1.03
		30	16.696	1.42	16.696	3.31
	750	15	32.874	3.69	32.874	2.97
		30	24.923	3.57	24.923	9.61
	1000	15	43.351	7.45	43.351	3.60
		30	33.713	7.59	33.713	19.82
	1250	15	56.671	13.53	56.671	5.95
		30	39.978	13.88	39.978	32.00
	1500	15	66.576	23.33	66.576	9.90
		30	51.267	24.01	51.267	66.12
0.5	500	15	37.896	4.43	37.896	1.11
		30	28.273	4.46	28.273	3.17
	750	15	52.788	12.15	52.788	2.30
		30	37.838	12.98	37.838	6.18
	1000	15	79.397	38.51	79.397	6.03
		30	51.842	33.39	51.842	13.53
	1250	15	101.433	72.11	101.433	11.21
		30	61.532	73.65	61.532	20.56
	1500	15	122.255	149.81	122.255	17.27
		30	81.267	153.14	81.267	27.25
0.75	500	15	41.900	5.63	41.900	0.89
		30	29.100	6.11	29.100	0.91
	750	15	56.900	18.51	56.900	1.72
		30	38.900	19.96	38.900	1.68
	1000	15	84.200	57.55	84.200	3.08
		30	53.300	50.98	53.300	2.97
	1250	15	111.400	120.69	111.400	5.10
		30	65.300	143.25	65.300	4.61
	1500	15	130.200	256.92	130.200	7.41
		30	85.800	294.60	85.800	7.06

Table 1: Comparison between OC and its version without the genetic algorithm.

80.89% and 94.54% for $\beta = 0.50$ and ranges between 103.49% and 156.28% for $\beta = 0.25$. It is remarkable to note that the theoretical upper bound is therefore always reached for $\beta = 0.75$, and almost reached for $\beta = 0.50$.

Furthermore, the gaps consistently increase as β decreases. This matches the intuitive expectation that, the scarcer is a resource, the more critical it is to optimize its usage. In more detail, for small values of β, we can imagine OC to highly prioritize sensors that, because of their favorable position, allow to achieve a greater lifetime if given a greater charge. These same sensors, in UC, are constrained to a low charge.

We also note, for any value of $|S|$ and $\beta = 0.25$, a further gap increase between 18% and 47% for the case $|T| = 30$ with respect to $|T| = 15$. That

β	|S|	|T|	UC Lifetime	UC Time	OC Lifetime	OC Time	GAP
0.25	500	15	10.475	0.90	22.126	1.03	111.22%
		30	7.275	0.95	16.696	3.31	129.50%
	750	15	14.225	1.70	32.874	2.97	131.10%
		30	9.725	1.61	24.923	9.61	156.28%
	1000	15	21.050	3.07	43.351	3.60	105.94%
		30	13.325	2.86	33.713	19.82	153.01%
	1250	15	27.850	5.39	56.671	5.95	103.49%
		30	16.325	4.40	39.978	32.00	144.89%
	1500	15	32.550	7.21	66.576	9.90	104.53%
		30	21.450	6.57	51.267	66.12	139.01%
0.5	500	15	20.950	0.87	37.896	1.11	80.89%
		30	14.550	0.94	28.273	3.17	94.32%
	750	15	28.450	1.75	52.788	2.30	85.54%
		30	19.450	1.59	37.838	6.18	94.54%
	1000	15	42.100	3.08	79.397	6.03	88.59%
		30	26.650	2.98	51.842	13.53	94.53%
	1250	15	55.700	5.35	101.433	11.21	82.11%
		30	32.650	4.49	61.532	20.56	88.46%
	1500	15	65.100	7.32	122.255	17.27	87.79%
		30	42.900	6.57	81.267	27.25	89.43%
0.75	500	15	31.425	0.84	41.900	0.89	33.33%
		30	21.825	0.95	29.100	0.91	33.33%
	750	15	42.675	1.71	56.900	1.72	33.33%
		30	29.175	1.63	38.900	1.68	33.33%
	1000	15	63.150	3.03	84.200	3.08	33.33%
		30	39.975	2.91	53.300	2.97	33.33%
	1250	15	83.550	5.25	111.400	5.10	33.33%
		30	48.975	4.48	65.300	4.61	33.33%
	1500	15	97.650	7.17	130.200	7.41	33.33%
		30	64.350	6.46	85.800	7.06	33.33%

Table 2: Comparison between the optimal and the uniform charging algorithms.

is, the waste of energy due to constrained uniform charging appears to have a greater impact on the final solution when there are more targets to cover, and therefore more sensors are likely to be needed in each cover.

With respect to computational times, UC is faster than OC in most scenarios, which is again an expected results given the more limited solutions space. The worst case is $\beta = 0.25$, $|S| = 1500$, $|T| = 30$, in which OC is one order of magnitude slower. That said, OC is still very fast in absolute terms, requiring as previously mentioned 66.12 seconds in the worst case.

6. Conclusion

In this work, we proposed a method to optimize the charge of sensor batteries in the context of a wireless sensor network used for monitoring purposes. After formally defining the underlying problem, we proposed a Column Generation solution approach, embedding an effective genetic algorithm for the subproblem. The computational results validate our idea, showing potential for large increases in terms of network lifetime when the charging is optimized, with respect to the traditional approach that considers uniform predefined charges for the sensors. The obtainable improvements are particularly relevant when the available energy is scarce.

References

1. C.M. Angelopoulos, S. Nikoletseas, T.P. Raptis, C. Raptopoulos, and F. Vasilakis. Efficient energy management in wireless rechargeable sensor networks. In *Proceedings of the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 309–316, 2012.
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. M. Cardei, M. T. Thai, Y. Li, and W. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proceedings of the 24th conference of the IEEE Communications Society*, volume 3, pages 1976–1984, 2005.
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.
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. Carrabs, R. Cerulli, C. D'Ambrosio, and A. Raiconi. Prolonging lifetime in wireless sensor networks with interference constraints. In M. Au, A. Castiglione, K.K. Choo, F. Palmieri, and K.C. Li, editors, *Green, Pervasive, and Cloud Computing. GPC 2017. Lecture Notes in Computer Science*, volume 10232, pages 285–297. Springer, Cham, 2017.
9. F. Castaño, E. Bourreau, A. Rossi, M. Sevaux, and N. Velasco. Partial target coverage to extend the lifetime in wireless multi-role sensor networks. *Networks*, 68(1):34–53, 2016.
10. 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.
11. F. Castaño, A. Rossi, M. Sevaux, and N. Velasco. An exact approach to extend network lifetime in a general class of wireless sensor networks. *Information Sciences*, 433-434:274–291, 2018.
12. M. Gentili and A. Raiconi. $\alpha-$coverage to extend network lifetime on wireless sensor networks. *Optimization Letters*, 7(1):157–172, 2013.
13. H.S. Jennath, S. Adarsh, and V.S. Anoop. Distributed IoT and applications: A survey. *Studies in Computational Intelligence*, 771:333–341, 2019.

14. C. Lin, K. Wang, Z. Chu, K. Wang, J. Deng ans M. S. Obaidat, and G. Wu. Hybrid charging scheduling schemes for three-dimensional underwater wireless rechargeable sensor networks. *The Journal of Systems and Software*, 146:42–58, 2018.
15. R.E. Mohamed, A.I. Saleh, M. Abdelrazzak, and A.S. Samra. Survey on wireless sensor network applications and energy efficient routing protocols. *Wireless Personal Communications*, 101(2):1019–1055, 2018.
16. T. Orekan and P. Zhang. *Underwater Wireless Power Transfer - Smart Ocean Energy Converters*. SpringerBriefs in Energy. Springer, Cham, 2019.
17. Y. Peng, Z. Li, W. Zhang, and D. Qiao. Prolonging sensor network lifetime through wireless charging. In *31st IEEE Real-Time Systems Symposium*, pages 129–139, 2010.
18. Y. Peng, W. Zhang, and D. Qiao. Charging and routing activity scheduling to prolong sensor network lifetime. In *Wireless Power Transfer Algorithms, Technologies and Applications in Ad Hoc Communication Networks*, pages 607–644, 2016.
19. V. Pryyma, D. Turgut, and L. Bölöni. Active time scheduling for rechargeable sensor networks. *Computer Networks*, 54:631–640, 2010.
20. Z. Ren, W. Xu, Y. Dai, and L. Jiang. Multi-event detection with rechargeable sensors. *Peer-to-Peer Networking and Applications*, 10:708–716, 2017.
21. S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *IEEE International Conference on Communications*, volume 2, pages 472–476, 2001.
22. L. Xie, Y. Shi, T. Hou, and H.D. Sherali. Making sensor networks immortal: An energy-renewal approach with wireless power transfer. *IEEE/ACM Transactions on Networking*, 20(6):1748–1761, 2012.
23. C. Yang and K.-W. Chin. Novel algorithms for complete targets coverage in energy harvesting wireless sensor networks. *IEEE Communications Letters*, 18(1):118–121, 2014.