On the use of multiple sinks to extend the lifetime in connected wireless sensor networks

F. Castaño\textsuperscript{a,b,1} A. Rossi\textsuperscript{a} M. Sevaux\textsuperscript{a} N. Velasco\textsuperscript{b}

\textsuperscript{a} Université de Bretagne-Sud, Lab-STICC, Lorient, France
\textsuperscript{b} Universidad de los Andes, Departamento de Ingeniería Industrial, Bogotá, Colombia

Abstract

This paper addresses the maximum network lifetime problem in wireless sensor networks. In this problem, the purpose is to schedule a set of wireless sensors, keeping them connected and guaranteeing that all targets are covered, while network lifetime is maximized. Two variants of this problem are considered; in the first case it is assumed that a single sink (base station) is available, the second case considers the presence of several sinks. To solve the problem, a hybrid Column Generation-GRASP heuristic is proposed. The method is shown to be able to find optimal or near optimal solutions efficiently in both cases.

Keywords: Column Generation, GRASP, Wireless Sensor Networks, Lifetime, Multiple Sink

1 Introduction

A wireless sensor network (WSN) is a net made of a large amount of wireless battery-operated sensors used to accomplish a set of monitoring and compu-
tation tasks. These devices work collaboratively or individually to perform their assigned tasks and deliver or spread the collected data to a remote base station through a multihop path of active sensors.

When the sensors and the base station are randomly placed, lifetime is limited by the sensors that are one hop away from the base station [2]. These sensors are bottleneck because they are required to pass all the information to the base station and their energy is drained faster than the energy of the distant sensors. A possible solution consists in deploying multiple base stations. Thus, sensors are used more efficiently by avoiding to transfer all sensing data in a single area.

In dense networks, lifetime can be maximized by creating covers, *i.e.*, groups of sensors that are active at the same time. This strategy has been proven to be efficient in several applications of WSN [3, 4]. Following this idea, decomposition approaches as column generation (CG) have been largely used to identify and create schedules for the covers. As well as in the classical implementation, CG decomposes the problem into a restricted master problem (RMP) and an auxiliary problem (AP). The former optimizes the lifetime using an incomplete set of columns, and the latter is used to identify profitable columns.

In this paper, an hybrid CG-GRASP heuristic is proposed to maximize network lifetime. The GRASP [1] heuristic is used to solve AP identifying interesting covers. If GRASP fails to find an interesting column, CG-GRASP could be turned into an exact method by using exact approaches to solve AP. The paper is structured as follows. Section 2 introduces the notation and the problem. In section 3 the CG approach is presented and the GRASP heuristic used to solve AP is described. Finally, in Sections 4 and 5 the experimental results and conclusions are presented.

2 Problem description

Consider a set $\mathcal{K} = \{k_1, k_2, ..., k_m\}$ of targets with known locations and a set $\mathcal{S} = \{s_1, s_2, s_3...s_n\}$ of sensors deployed to cover the targets. A target is considered covered by a sensor if it lies within its sensing range $R_s$ (*observation link*). The sensor nodes are able to collect and (re)transmit the information to other sensors or a base station if they are within their communication range $R_c$ (*communication link*). All the information retrieved from the targets must be collected by a set $\mathcal{B} = \{b_1, b_2, b_3...b_r\}$ of base stations.

Let $E$ be the set of arcs, where $e(u, v) \in E$ exists if there exists a communication link between the elements $u, v \in \mathcal{S}$, $u \in \mathcal{S}$ and $v \in \mathcal{B}$ or if
there is an observation link between the elements $u \in K$ and $v \in S$. An additional, and artificial, super node $\Upsilon$ is defined to query the information collected at the sink nodes. In other words, a communication link exists between the base stations and the new super node ($\exists e(u, \Upsilon), \forall u \in B$).

Let $G = (\mathcal{V}, E)$ be a directed graph with $\mathcal{V} = K \cup S \cup B \cup \Upsilon$. A feasible cover $C_j \subseteq S$ is a subset of sensors such that for all pairs $(k_i, \Upsilon)$ there exists a path through the elements of $C_j \cup B$. The set of all the feasible covers of $S$ is denoted by $\Omega = \{C_1, C_2, \ldots, C_l\}$. A decision variable $t_j$ is introduced to identify the time interval allocated to a feasible cover $C_j$. The connected maximum network lifetime problem with multiple sink connectivity is to find a collection of pairs $(C_j, t_j)$ in such a way that network lifetime, $\sum_{j \in \Omega} t_j$, is maximized.

3 Decomposition approach

CG based approaches are widely used to tackle the design of energy-efficient wireless sensor networks [2,3,4]. In this approach the problem is divided into two subproblems. First, RMP, containing a reduced set of the feasible columns (covers) $\Omega' \subseteq \Omega$, is used to identify the optimal timings for the current covers. RMP is solved using linear programming, and the optimal dual variables values found are used as an input to solve AP. This one, is used iteratively to identify new profitable covers to be included in RMP. For each new cover the reduced cost is evaluated. If it is greater than zero, which means that the cover is interesting, it is added to RMP and a new CG iteration is performed. The CG process stops when no more profitable columns are found.

3.1 Master problem

Let $y_{si,j}$ be a binary parameter that takes the value of 1 if sensor $s_i$ is active in cover $C_j$ and 0 otherwise. The problem is formulated by:

Maximize: $\sum_{C_j \in \Omega'} t_j$ \hspace{1cm} (1)

Subject to:

$\sum_{C_j \in \Omega'} y_{si,j} t_j \leq b_{si}$ \hspace{0.5cm} $\forall \ s_i \in S$ \hspace{1cm} (2)

$t_j \geq 0$ \hspace{0.5cm} $\forall \ C_j \in \Omega'$ \hspace{1cm} (3)
The objective function (1) maximizes the network lifetime. The set of equations (2) enforces that all energy constraints are satisfied. Constraints (3) are the non-negativity constraints.

3.2 Auxiliary Problem

AP is used to identify a connected structure in $G$ that maximizes the reduced cost. We propose to model AP as a network flow problem. It is assumed that source nodes (targets) offer some information that is transferred to $\Upsilon$ through the transshipment nodes (sensors or the base stations). This means that similar approaches can be used to solve the problem with a single or multiple base stations. AP decides, based on the reduced cost criterion, the values $y_{vj}$ that indicate if a sensor $v \in S$ belongs to the new cover $C_j$. At each iteration of CG, the cost of using the sensor $v$ is denoted by $\pi_v$, the dual variable value associated with constraints (2), which is used to compute the reduced cost of a feasible solution. For each $e(u, v) \in E$, variable $x_{uv}$ indicates the flow of information transferred through this arc. It should be noted that, even although a feasible network structure must be defined, only the information concerning the optimal cover is required to be returned to MP.

By using the notation above, the auxiliary problem is represented by equations (4-10) as follows:

Maximize: $1 - \sum_{v \in S} y_{vj} \pi_v$ \hspace{1cm} (4)

Subject to:

\begin{align*}
\sum_{u \in S \mid \exists e(v, u)} x_{vu} &= 1 \quad \forall \ v \in K \hspace{1cm} (5) \\
\sum_{u \in V \mid \exists e(v, u)} x_{vu} - \sum_{u \in V \mid \exists e(u, v)} x_{uv} &= 0 \quad \forall \ v \in S \cup B \hspace{1cm} (6) \\
\sum_{u \in V \mid \exists e(u, \Upsilon)} x_{u\Upsilon} &= |K| \hspace{1cm} (7) \\
\sum_{u \in V \mid \exists e(v, u)} x_{vu} + \sum_{u \in V \mid \exists e(u, v)} x_{uv} &\leq 2|K|y_{vj} \quad \forall \ v \in S \hspace{1cm} (8) \\
x_{uv} \in \mathbb{Z^+} \cup \{0\} \quad \forall \ u, v \in V \hspace{1cm} (9) \\
y_{vj} \in \{0, 1\} \quad \forall \ v \in V \hspace{1cm} (10)
\end{align*}
By solving model (4-10) a network structure with optimal reduced cost can be found (4). Equations (5-7) are information flow conservation constraints. These constraints are used to guarantee that the flow arising at the nodes is transferred to Υ through the sensors and the information is not stored in sensor nodes. Constraints (8) guarantee that sensors transferring information are labeled as active in a solution.

3.3 A GRASP heuristic for the auxiliary problem

In this paper, the use of a GRASP [1] heuristic is proposed to solve AP. A general description of our approach is outlined in Algorithm 1. The method iteratively defines connected structures in $G$ by using a greedy randomized construction procedure (line 4) and a local search to improve these ones (line 5). A parameter $\alpha$ is used to manage the level of randomization of the constructive phase. To stop the heuristic two criteria are used, the number of iterations without improvement ($Max\_iters$) and the maximum running time ($Max\_time$). Note that the objective of the algorithm is to minimize $\sum_{v\in S} y_{v} \pi_{v}$ (lines 6-8), that maximizes the objective function for AP (3).

**Algorithm 1** Greedy randomized adaptative search procedure

GRASP($\pi_{v}, G, \alpha$)

1: $C_{j} \leftarrow \emptyset; \hat{C}_{j} \leftarrow \emptyset; Max\_time \leftarrow 0.5s; time \leftarrow 0; iter \leftarrow 0; f(C_{j}) \leftarrow \infty$
2: while $time \leq Max\_time \& iter \leq Max\_iters$ do
3: $S_{av} \leftarrow S \cup B, S_{act} \leftarrow \emptyset, K_{cov} \leftarrow \emptyset, s_{0} \leftarrow \emptyset$
4: $\hat{C}_{j} \leftarrow \text{GR\_DFS}(S_{av}, S_{act}, K_{cov}, \pi_{v}, G, s_{0}, \alpha)$
5: $\hat{C}_{j} \leftarrow \text{Local\_Search}(\hat{C}_{j})$
6: if $f(\hat{C}_{j}) \leq f(C_{j})$ then
7: $C_{j} \leftarrow \hat{C}_{j}; iter \leftarrow 0$
8: end if
9: $time \leftarrow \text{Update\_time}(); iter \leftarrow iter + 1$
10: end while
11: return $C_{j}$

3.3.1 Greedy randomized construction

The constructive phase uses a randomized greedy depth first search strategy (GR\_DFS) to create connected structures in $G$. A general overview of the algorithm is presented in Algorithm 2. First, an initial tree containing only $\emptyset$ is generated. Next, among the elements sharing a communication link with this, a restricted candidate list (RCL) of nodes is created (lines 2-8). A new
node belonging to this list is randomly selected (line 9), added to the tree and used as a reference to elaborate a new RCL. The targets with an observation link with it are marked as covered (lines 12-14). If RCL is an empty list, the previously visited nodes are used again to create the RCL and to continue the construct phase. GR_DFS stops when all the targets are marked as covered.

Algorithm 2 Greedy randomized depth first search

\[\text{GR\textunderscore DFS}(S_{av}, S_{act}, K_{cov}, \pi_v, G, s_0, \alpha)\]

1: \textbf{while} \(|K_{cov}| < |K|\) \textbf{do}
2: \hspace{1em} \(RCL = \emptyset\)
3: \hspace{1em} \textbf{for} \(u \in B\) \textbf{do}
4: \hspace{2em} \(\pi_u \leftarrow \min_v: \{\exists e(v,u) \land v \in S_{av}\} \pi_v\)
5: \hspace{1em} \textbf{end for}
6: \hspace{1em} \(c_{min} = \min_v: \{v \in S_{av} \land \exists e(v,s_0)\} \pi_v\)
7: \hspace{1em} \(c_{max} = \min_v: \{v \in S_{av} \land \exists e(v,s_0)\} \pi_v\)
8: \hspace{1em} \(RCL \leftarrow \{v \in S_{av} \mid \exists e(v,s_0) \land \pi_v \leq c_{min} + \alpha(c_{max} - c_{min})\}\)
9: \hspace{1em} \(s_0 \leftarrow \text{Random\textunderscore Selection}(RCL)\)
10: \hspace{1em} \(S_{act} = S_{act} \cup s_0\)
11: \hspace{1em} \(S_{av} = S_{av} \setminus s_0\)
12: \hspace{1em} \textbf{for} \(u \in K \mid \exists e(u,s_0)\) \textbf{do}
13: \hspace{2em} \(K_{cov} = K_{cov} \cup u\)
14: \hspace{1em} \textbf{end for}
15: \hspace{1em} \text{GR\textunderscore DFS}(S_{av}, S_{act}, K_{cov}, \pi_v, G, s_0, \alpha)
16: \hspace{1em} \textbf{end while}
17: \textbf{return} \(S_{act}\)

3.3.2 Local search

The local search procedure is based on the sequential exploration of two neighborhood structures following a first improvement strategy. First, a remove neighborhood is proposed to check for useless sensors that are not required to meet the constraints. Then, a remove-insert neighborhood is considered to exchange one active sensor by an inactive sensor with lower cost that keeps the feasibility of the cover.

4 Results and discussion

In order to evaluate the proposed method, four sets of instances with \(|S|=50, 100, 150, \text{ and } 200\) sensors are used. Two sets of targets \(|K|=15, 30\) complete the instances. The effect of using multiple sinks is studied using up to 3
base stations. The sensors are assumed homogeneous and \( b_{sv} = 1 \ \forall v \in S \). The approach is coded in C++ and the Gurobi optimization engine is used to solve RMP. Computational experiments are performed on an Intel Core i-5 processor @ 1.6 GHz with 2 GB of RAM running under OS-X Lion. When the GRASP heuristic is not able to find an interesting cover, AP is solved using an integer programming solver, so it is possible to check optimality; however, the additional time consumed is not reported.

Table 1 presents a summary of the results. The columns labeled \textbf{Avg. time(s)} present the average computational time required for CG-GRASP. The columns \textbf{Opt/#Ins} indicate the fraction of optimal solutions found with CG-GRASP at each instance group. The results show that, as expected, computational time increases with instance size. Furthermore, it is observed that the increase on the number of sinks increases as well as the average computational time to reach an optimal solution.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Sinks = 1</th>
<th>Sinks = 2</th>
<th>Sinks = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>S</td>
<td>\times</td>
<td>K</td>
</tr>
<tr>
<td>50 15</td>
<td>0.19</td>
<td>1.00</td>
<td>0.36</td>
</tr>
<tr>
<td>100 15</td>
<td>20.02</td>
<td>0.80</td>
<td>77.24</td>
</tr>
<tr>
<td>150 15</td>
<td>50.54</td>
<td>0.80</td>
<td>87.06</td>
</tr>
<tr>
<td>200 15</td>
<td>20.56</td>
<td>1.00</td>
<td>96.38</td>
</tr>
<tr>
<td>50 30</td>
<td>0.24</td>
<td>1.00</td>
<td>0.47</td>
</tr>
<tr>
<td>100 30</td>
<td>30.61</td>
<td>0.80</td>
<td>85.48</td>
</tr>
<tr>
<td>150 30</td>
<td>47.21</td>
<td>0.80</td>
<td>114.18</td>
</tr>
<tr>
<td>200 30</td>
<td>29.69</td>
<td>1.00</td>
<td>60.20</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>24.88</strong></td>
<td><strong>0.90</strong></td>
<td><strong>65.17</strong></td>
</tr>
</tbody>
</table>

Table 1
Computational results for CG-GRASP.

Solving the problem by using a pure CG method requires extensive computational effort. The detailed results of the pure CG approach can be found at the authors website (http://or-labsticc.univ-ubs.fr/). These results are used as a basis for counting the number of problem instances for which CG-GRASP finds the optimal solution. The ratio of optimal solutions found is shown on columns 4, 6 and 8 of Table 1.

An additional set of experiments has been performed by using the set of instances proposed by Raiconi and Gentili [3]. CG-GRASP is compared with the CMLP-GRASP heuristic proposed by the authors. As presented in our previous work, the results indicate that CG-GRASP outperforms their results in terms of solution quality and computational time [5].
5 Conclusions

This paper addresses the problem of maximizing lifetime of connected wireless sensor networks with single and multiple sinks. The proposed solution approach combines column generation with GRASP to identify profitable covers and extend network lifetime. Furthermore, the GRASP heuristic is shown to be efficient to tackle the auxiliary problem arising at each iteration of CG.

The experimental results confirm that the method is able to obtain near optimal solutions in low computational times. The experiments show that the average time required for CG-GRASP was below 2.5 minutes for the larger instances. Moreover, it is observed that the optimal solution was found in most experiments.

Future research will consider extensions to problems with partial coverage. Additionally, similar approaches will be applied to problems including the effect of distance sensor-sensor and sensor-target on the energy consumed by transmission and detection respectively.

References


