MGRASP and MSAL Algorithms for Centralized Traffic Management of Large Wireless Sensor Networks

Maria Magdalena Czajko[1,2] and Jacek Wojciechowski[1]

[1] Warsaw University of Technology, Institute of Radioelectronics, Warsaw, Poland,
e-mail: mczajko@hotmail.com
e-mail: j.wojciechowski@ire.pw.edu.pl
[2] Telekomunikacja Polska S.A., Warsaw, Poland, email: mariamagdalena.czajko@telekomunikacja.pl

Abstract. This paper concerns a hop and degree constrained minimum spanning forest problem with the minimization of the number of trees (HDMSFMT). HDMSFMT arises during the calculation of the shortest paths in large wireless sensor networks (LWSNs) with centralized traffic management. This is a bi-criteria optimization problem, which is shown in this paper to be NP-hard. We formulate the HDMSFMT problem using formal mathematical notations. Two efficient heuristic algorithms for HDMSFMT are proposed and compared, namely Multi-objective Simulated Allocation (MSAL) and Multi-objective Greedy Randomized Adaptive Search Procedure (MGRASP). Numerical examples show that it is possible to achieve a set of feasible and satisfactory results for a wide range of graphs' types.

1 Introduction

LWSNs are invaluable in many civil applications, e.g. for monitoring of natural phenomena like earthquakes, volcanic activity, landslides, oceanic events or biological activity of living plants. They are also useful in military applications, e.g. battlefield monitoring. LWSNs are used for collecting, processing, and disseminating wide ranges of complex environmental data.

A typical wireless sensor network usually consists of hundreds to thousands of spatially distributed, autonomous, and low-power devices with limited computational capabilities using sensors to monitor environmental or physical conditions at different, unattended locations. Hierarchical structure has been considered as a necessity for large-scale systems. For this reason, all sensor nodes can play a sensing and/or gateway role. The sensing node can monitor environmental or physical conditions and collect data, whereas the gateway node can aggregate and process the data collected from all sensing nodes and send the data to a sink node. Sensors are usually powered by batteries with limited capacity, so the problem of energy economy is of paramount importance. We assume that sensor nodes posses all features, i.e. memory, interfaces or protocols we would like to use in our LWSN or it is possible to implement them easily.

In the literature we can find several hardware/software system architectures. We would like to focus on different approaches for traffic management problem for LWSNs. There are two basic approaches for traffic management in LWSNs: centralized [5], [6] and decentralized [4]. Decentralized scenarios assume that all sensor nodes take part in gateway selection, routing paths' calculation, etc. In case of centralized approaches we can make use of a high-energy control center or base station to determine the aforementioned parameters. Experimental results show that centralized scenarios perform better than decentralized. The key reason is that a control center utilizes its global knowledge of the network structure to produce routing paths or gateways' locations that require less energy for the data transmission. The above advantages encouraged us to focus on the centralized approach.

The next step after elaborating the communication approach is to work out an algorithm for efficient traffic management. The researchers propose generally two scenarios [5], [6]: clustering-based approaches (most popular), and multi-hop routing, which is partially based on routing schemes used in wired packet networks. In the clustering-based approach sensor nodes are grouped into clusters. Each cluster includes a dedicated head node, which processes data gathered directly from sensor nodes within this cluster range. The head node then transmits aggregated data to base station either directly [5] or indirectly, using routing algorithm between cluster heads [6]. The role of a cluster head may change randomly depending on the current energy level of sensor nodes, what leads to the improvement of power management [6]. Comparing the multi-hop and clustering based approaches we can find that the first is more power economical because sensor nodes can communicate with each other before sending data to the head node. In case of clusters, all sensor nodes within each cluster send data directly to cluster head. It can lead to higher overall power consumption than in the case of multi-hop solutions. From the reliability perspective, the clustering-based approaches perform better than the multi-hop solutions. The reason is that, in case of clustering-based scenarios, a damage on a sensor node has lower impact on data transmission than in case of multi-hop approaches.

To achieve the compromise between the two routing schemes compared above we propose a tree-based routing approach with hop and degree constraints. Our solution is based on the centralized gateway nodes' election and determining a "forest" of sensor nodes. We assume that each sensor node is capable of sensing and gateway functionality. Gateway nodes send data directly to the control center. The developed architecture is presented in Fig. 1.

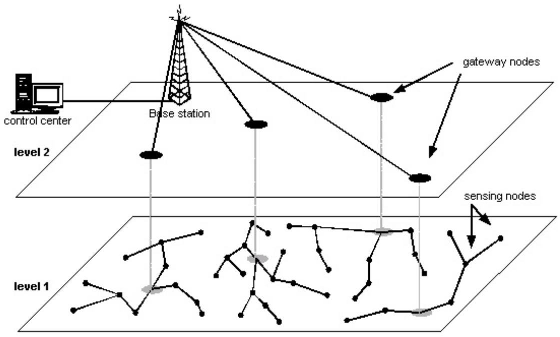

Figure 1. Reference architecture for LWSN.

In section 2 we describe the problem of the hop and degree constrained minimum spanning forest with the minimization of the number of trees (HDMSFMT), which is met during calculation of the shortest paths in our traffic management scheme. In section 3 we discuss the complexity of HDMSFMT. Sections 4 and 5 describe two proposed heuristic algorithms: MSAL and MGRASP followed by numerical results. Section 6 includes a test example and numerical results while section 7 contains concluding remarks.

2 Problem Description

Consider undirected graph G=(V, E) with set V of vertices representing sensor nodes and set E of edges representing wireless links between these nodes. All nodes in G are of the same type and their geographical locations are known. Undirected edge e connects two nodes i and j and is denoted by $\{i,j\}$. Cost c_{ij} representing a distance between nodes i and j is assigned to each edge.

Tree T_n=(V_{Tn}, E_{Tn}), where n=1, 2, ..., N, is a connected acyclic subgraph of G, where $V_{Tn} \subseteq V$ and $E_{Tn} \subseteq E$. E_{Tn} contains edges forming a tree spanned over vertices from V_{Tn}. In V_{Tn} we distinguish one node r, which represents a root (gateway node) of T_n. Spanning forest $F = \{T_1, T_2, ..., T_N\}$ is a set of N mutually vertex disjoint trees T_n (n = 1, 2, ..., N), i.e. $\bigcup^{N}_{n=1} V_{Tn} = V$, $n \neq m \Rightarrow V_{Tn} \cap V_{Tm} = \varnothing$.

Let's replace each edge e in E by two parallel arcs in opposite directions. We obtain then a bidirected graph G_B=(V,A), where A represents the set of arcs, i.e. A=$\{(i,j)$: $\{i,j\} \in E\}$. We assume that directed forest F_B of graph G_B consists of directed trees T_{Bn}. Tree T_{Bn} is obtained from T_n by assigning orientation to each edge towards root node. In such a transformed model, root nodes have only incoming edges while other nodes have one outgoing arc (see Fig. 2). Assume also that c_{ij} denotes the cost of arcs (i,j) and (j,i) linking vertices i and j. It is the same constant as in case of the undirected graph model.

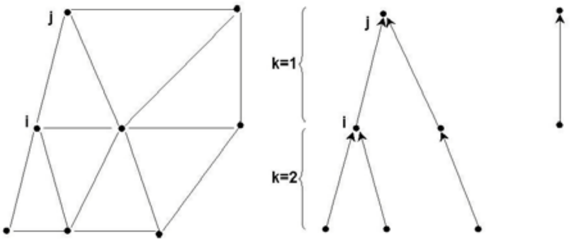

Figure 2. Graph and forest.

The optimization problem that we define below is constrained. H is the maximum acceptable distance measured as the number of hops between the root and vertices in T_{Bn} (n = 1, 2, ..., N). In our model k is the level of each arc in T_{Bn} (see Fig. 2). D is the maximum acceptable value of the sum of outdegree and indegree of vertex in F_B.

The objective of the HDMSFMT problem is to find a spanning forest F_B in G_B with the minimum number N of trees and the minimum total cost under D and H constraints. We state the mathematical formulation of HDMSFMT below:

indices

i, j, l, m, n = 1, 2, ..., N	nodes
k = 1, 2, ..., H	levels

constants

c_{ij}	cost of edge (i,j)

D maximum degree (sum of indegree and outdegree) of each node
H maximum length of a path between root and a non-root node in tree T_{Bn}

variables
y_n = 1 if node *n* is a root of tree T_{Bn}, 0 otherwise
x_{ijk} = 1 if directed edge (*i*,*j*) of graph G_B is on level *k* in tree T_{Bn}, 0 otherwise

objectives

f1: minimize $\sum_i \sum_j c_{ij} \sum_k x_{ijk}$ (1)

f2: minimize $\sum_n y_n$ (2)

constraints

$$\sum_i x_{ij1} - D \cdot y_n \leq 0 \qquad n = j = 1, 2, ..., N \tag{3}$$

$$\sum_j \sum_k x_{ijk} + y_n = 1 \qquad n = i = 1, 2, ..., N \tag{4}$$

$$x_{ijk} + \sum_l x_{lm(k+1)} \leq D \qquad i,j = 1, 2, ..., N \qquad i = m \qquad k = 1, 2, ..., H\text{-}1 \tag{5}$$

$$x_{ij(k+1)} - \sum_m x_{lmk} \leq 0 \qquad i,j = 1, 2, ..., N \qquad l = j \qquad k = 1, 2, ..., H\text{-}1 \tag{6}$$

$$\sum_i \sum_j \sum_k x_{ijk} + \sum_n y_n = N \tag{7}$$

Objective function $\boldsymbol{f_1}$ (1) returns the total cost of F_B, which is the sum of costs assigned to the edges that form F_B. The second objective function $\boldsymbol{f_2}$ (2) returns the total number of trees in F_B. Inequality (3) assures that if node *n* is not a root, then there is no arc directed to *n*, which is on the first level (see Fig. 2). Moreover, when *n* is a root node, then maximum degree of *n* can not exceed *D*. Constraint (4) forces only one outgoing edge at *n* when *n* is not a root node and the lack of outgoing edges if *n* is a root node. Inequality (5) assures that the sum of incoming edges at node *i*, which are on level *k*+1, and outgoing edges at node *i*, which are on level *k*, should not exceed *D*. Existence of an incoming edge at node *j* on level *k*+1 implies the existence of outgoing edge at node *j* on level *k* (6). Equation (7) defines the relationship between the number of trees and the number of nodes and edges in the forest.

3 Complexity of HDMSFMT Problem

The hop constrained [3] with the degree-constrained minimum spanning tree problem [2] is NP-hard and can be reduced to HDMSFMT, so the HDMSFMT problem is also NP-hard.

The HDMSFMT problem is additionally complicated by the fact that it belongs to a set of multi-criteria optimization problems. Multi-objective optimization is the process of searching one or more decision variables that simultaneously satisfy all constraints, and optimize an objective function vector [1]. In our case we have two objectives: the minimization of the total cost and the minimization of the number of trees in the forest, which conflict with each other.

While solving a multi-criteria problem we obtain not only one solution vector of decision variables. It will be a set of decision vectors, which is known as Pareto-optimal set. Pareto-optimal solutions are not dominated by any other solution in the feasible space. A solution *x* is dominated if there exists a feasible solution *y* that is at least as good as *x* with respect to every dimension of objective space and strictly better than *x* on at least one objective. Thus, a non-dominated solution is any solution that is not dominated by any other feasible solution.

The Pareto-optimal frontier represents images corresponding to Pareto-optimal solutions in the objective space and is formed by the solutions whose performance on one objective cannot be improved without sacrificing performance on at least one other [2], [7], and [8].

As our problem is NP-hard we propose two heuristics presented below to solve it.

4 MSAL Algorithm

A Multi-objective Simulated Allocation algorithm (MSAL) was developed to solve HDMSFMT. A single-objective version of Simulated Allocation (SAL) is presented in [7].

Generally, MSAL uses two procedures: *allocate* and *disconnect*. The procedure *allocate* randomly selects nodes to classify them as root or non-root nodes in forest F_i taking into account edges' existence, cost, and D and H constraints. The procedure *disconnect* removes a defined number of nodes from the forest in a random way (*percentage* parameter). Both operations are run sequentially. The pseudo-code of MSAL is presented below:

```
initialize: $G=(V,E)$, $Pareto=\emptyset$, $step$, $limit$, $H$, $D$, $percentage$, $i=1$
allocate all nodes to create a forest $F_i$
add this solution to $Pareto$
repeat
    $i$++
    disconnect randomly a $percentage$ of all nodes from $F_i$
    allocate all nodes to create forest $F_i$
    if new solution is non-dominated then update $Pareto$
    $step$++
until $step = limit$
```

5 MGRASP Algorithm

A Multi-objective Greedy Adaptive Search Procedure (MGRASP) was developed as an alternative method to solve HDMSFMT. MGRASP is divided into two phases that are run sequentially, predefined number of times. The first phase (lines: 3-11) allows obtaining as low number of trees as possible and a low value of the total cost of the forest, not necessary optimal, and satisfying the constraints. The second phase (lines: 12-16) improves the results.

```
0: initialize: $Pareto=\emptyset$, $H$, $D$, $step1=0$, $limit1$, $step2=0$, $limit2$, $p$
1: repeat
2:     initialize: $B=\emptyset$, $G=(V,E)$, $T_i=\emptyset$, $i=0$
3:     while $V \neq \emptyset$ do
4:         $i$++
5:         choose the root node of tree $T_i$, e.g. randomly
6:         determine spanning tree $T_i$ of $G$ while satisfying constraints $D$ and $H$
7:         add the remaining nodes and edges that do not create $T_i$ to backup set $B$
8:         fill $T_i$ up with nodes from $B$ with probability $p$ taking into account $H$, $D$
9:         overwrite graph $G=(V,E)$ by backup set $B$
10:    endwhile
11:    if a new solution is non-dominated then update $Pareto$
12:    repeat
13:        choose randomly two nodes to be exchanged or to switch the branch
14:        if a new solution is non-dominated then update $Pareto$
15:        $step2$++
16:    until $step2 = limit2$
17:    $step1$++
18: until $step1 = limit1$
```

6 Computational Results

To evaluate quality of both algorithms numerical experiments were carried out. The experiments were performed on the test graphs of the following types:

- graphs with 50 nodes and 100 edges (connected), and 100 nodes and 200 edges (disconnected); the edges were generated randomly,
- connected graphs with 50 nodes and 613 edges, and 100 nodes and 2475 edges; the edges were generated randomly.

Costs of links (weights of edges) were generated randomly from the (0, 100> interval. Also the root nodes were selected randomly. Simulations were performed for three combinations of the values of *H* and *D* constraints: *D*=2 and *H*=4, *D*=4 and *H*=3, *D*=6 and *H*=5. In the first phase of MGRASP the probability *p* of the node linking to a tree, was set to 0, 0.3, 0.6 and 1. *Limit1* and *limit2* parameters were set to 100 and 50000, respectively. The values of *percentage* parameter for MSAL that we checked were set to 10%, 50%, and 80%. The *limit* parameter was 50000.

Table 1 includes the number of pseudo-Pareto-optimal solutions found by both of the proposed algorithms. Light grey shaded records in the table show the highest values of the parameters for each case (graph type, *D* and *H* combination) among all values obtained by MSAL and MGRASP, respectively. It can be observed that MSAL is more effective than MGRASP because it provides larger size of the pseudo-Pareto set. As expected, MSAL with *percentage*=50% and 80% allowed obtaining a larger size of pseudo-Pareto set than MSAL with *percentage*=10% because of higher diversification.

Table 1. Number of Pseudo-Pareto-optimal solutions and domination parameter.

type of *G*	*D, H*	No. of Pseudo-Pareto-Optimal Solutions							Domination Parameter						
		MSAL			MGRASP				MSAL			MGRASP			
		10%	50%	80%	*p* = 0	0,3	0,6	1	10%	50%	80%	*p* = 0	0,3	0,6	1
50n,100e	2, 4	12	10	12	8	6	6	7	9	3	1	0	1	3	3
	4, 3	11	13	14	6	7	7	8	7	3	6	6	0	1	0
	6, 5	14	17	16	5	3	3	3	9	2	6	5	0	0	0
50n,613e	2, 4	2	3	2	2	3	3	3	2	1	0	1	1	0	0
	4, 3	3	4	3	4	2	2	2	3	0	0	4	1	0	0
	6, 5	3	3	2	2	1	1	1	2	1	0	2	0	0	0
100n,200e	2, 4	14	12	12	7	8	7	5	14	1	0	3	3	2	1
	4, 3	13	13	17	8	7	3	7	7	6	5	7	0	1	0
	6, 5	15	16	16	5	3	4	4	12	5	4	5	1	0	0
100n,2475e	2, 4	2	1	2	1	2	1	3	1	1	0	1	0	0	0
	4, 3	3	3	2	3	3	2	2	2	0	1	3	0	2	0
	6, 5	2	3	3	2	1	1	1	2	1	0	0	0	1	0

It can happen that one algorithm provides many pseudo-Pareto-optimal solutions but most of them are dominated by solutions obtained thanks to the other algorithm. Thus, we proposed and checked an additional parameter, a domination parameter, which allows for comparing the algorithms more precisely. The domination parameter defines the number of non-dominated solutions generated by the algorithm out of pseudo-Pareto-optimal solutions obtained by all compared methods. It could be seen from Table 1 that MSAL with *percentage*=10% and MGRASP with *p*=0 produces a higher value of the dominance parameter than in the remaining cases. It means, that MSAL with *percentage*=10% provides the highest number of non-dominated solutions among all solutions with *percentage*=10%, 50%, and 80%. Similar situation is observed in case of MGRASP. It seems that too strong diversification in MSAL by disconnecting a high number of nodes (*disconnect* procedure) leads to worse solutions. In case of MGRASP we observe, that filling in the trees with as many nodes as possible gives worse solutions than in case of MGRASP with *p*=0.

In the next step we calculate the domination parameter to compare MSAL with *percentage*=10% and MGRASP with *p*=0, as the best values of domination parameter were obtained in these cases. It can be observed in Table 2 that the domination parameter is the highest for MSAL with *percentage*=10% for sparse graphs (50 nodes, 100 edges; 100 nodes, 200 edges), what means that MSAL provides the best solutions. On the other hand MGRASP allowed obtaining better solutions for denser graphs (50 nodes, 613 edges; 100 nodes, 2475 edges). But generally, MSAL provides the highest values of the domination parameter.

Table 2. Domination for MSAL (*percentage*=10%) and MGRASP (*p*=0)

type of *G*	***D, H***	**Domination**	
		MSAL	**MGRASP**
		10%	***p* = 0**
50n,100e	2, 4	12	0
	4, 3	10	1
	6, 5	12	3
50n,613e	2, 4	0	2
	4, 3	3	0
	6, 5	0	2
100n,200e	2, 4	11	4
	4, 3	13	2
	6, 5	12	4
100n,2475e	2, 4	0	1
	4, 3	2	3
	6, 5	1	1

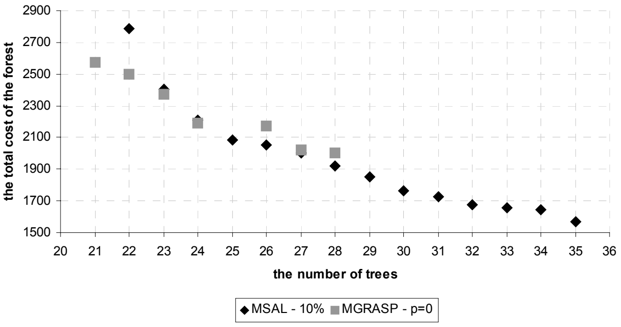

Figure 3. Example of the pseudo-Pareto frontier for MSAL with *percentage* = 10% and MGRASP with *p* = 0. Graph with 100 nodes, 200 edges, *D* = 2 and *H* = 4. The number of trees in the forest is assigned to the horizontal axis and the total cost of the forest is assigned to the vertical axis.

Fig. 3 shows an example of the pseudo-Pareto frontier for MSAL with *percentage*=10%, and MGRASP with *p*=0. It can be seen that MSAL gives more pseudo-Pareto-optimal solutions than MGRASP. On the other hand MGRASP sometimes allows obtaining lower cost for the minimum number of trees than MSAL as we can see in Fig. 3.

7 Conclusions

In this paper we defined the so-called hop and degree constrained minimum spanning forest problem with the minimization of the number of trees (HDMSFMT), which could be met during the shortest paths' calculation in centralized managed LWSNs. We have not found any formulation of this problem in the literature known to us. Since HDMSFMT is NP-hard, we developed two efficient multi-objective heuristics: MSAL and MGRASP algorithms. Numerical examples show that it is possible to achieve a set of feasible and satisfactory results for a wide range of graphs' types. Generally, the best solutions were obtained by the MSAL algorithm with *percentage*=10%, but it sometimes can be observed that MGRASP allows obtaining lower cost for the minimum number of trees than MSAL.

Further research will be to improve the methods presented in this paper i.e. a tabu-list can be implemented to forbid selection of solutions got in previous iterations or additional new rules for root selection. Adding a condition to build as balanced spanning forest as possible could also extend the analyzed problem. Another task will be to develop decision algorithm to choose the most preferred solution from the pseudo-Pareto-optimal set.

Bibliography

[1] Baños, R., Gil, C., Paechter, B., Ortega, J. Parallelization of population-based multi-objective meta-heuristics: An empirical study. *Applied Mathematical Modelling* 30 (2006) 578-592.

[2] Da Cunha, A. S., Lucena, A. Algorithms for the degree-constrained minimum spanning tree problem. *Electronic Notes in Discrete Mathematics* 19 (2005) 403-409.

[3] Gouveia, L. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research* 95 (1996) 178-190.

[4] Heinzelman, W. B., Chandrakasan, A. P., Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. *Proc. 33rd Hawaii Int'l. Conf. Sys. Sci.* (2000).

[5] Heinzelman, W. B., Chandrakasan, A. P., Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications* (2002) vol. 1, Issue 4, 660 – 670.

[6] Muruganathan, S. D., Ma, D. C., Bhasin, R. I., Fapojuwo, A. O. A centralized energy-efficient routing protocol for wireless sensor networks. *IEEE Communications Magazine* (2005), vol. 43, Issue 3, S8 – 13.

[7] Pióro, M., Medhi, D. *Routing, flow, and capacity design in communication and computer networks*. Elsevier, 2004.

[8] Rahimi-Vahed, A. R., Rabbani, M., Tavakkoli-Moghaddam, R., Torabi, S. A., Jolai, F. A multi-objective scatter search for a mixed-model assembly line sequencing problem. *Advanced Engineering Informatics* 21 (2007) 85-99.