Min-energy Broadcast in Fixed-trajectory Mobile *Ad-hoc* Networks

J.M. Díaz-Báñez [*] R. Fabila-Monroy [†] D. Flores-Peñaloza [†]
M. A. Heredia [†] J. Urrutia [†]

Abstract

This paper concerns about mobile *ad-hoc* wireless networks, but with the added restriction that each radio station has a rectilinear trajectory. We focus on the problem of computing an optimal range assignment for the stations, which allows to perform a broadcast operation from a source station, while the overall energy deployed is minimized. An $O(n^3 \log n)$-time algorithm for this problem is presented, under the assumption that all stations have equally sized transmission ranges. However, we prove that the general version of the problem is NP-hard, and it is not approximable within a sub-logarithmic factor (unless P=NP). We then consider a special case of the general problem (also NP-hard), and present an approximation algorithm whose approximation factor is $(\ln n + 1)m$, where m is the size of the optimum solution.

1 Introduction

In recent years, optimization problems in ad-hoc wireless networks have attracted significant attention due to their potential applications in civil and military domains [5, 11]. Typically, the radio stations in such a network have a limited energy resource (battery for example), and consequently, energy efficiency is an important design consideration for these networks [2, 4]. On the other hand, the rapidly expanding technology of cellular communications, wireless LAN, and satellite servicesadds movement to the stations, making necessary to extend the concept of *ad-hoc* wireless networks to *mobile ad-hoc wireless networks.*

A *broadcast communication* is a task initiated by a source station which has to disseminate a message to all stations in the network [10]. In this paper, we focus on source-initiated broadcasting of messages in mobile ad-hoc wireless networks, yet we restrict the movement of the stations to rectilinear trajectories on the plane and constant velocity. This approach can be applied in settings like satellite networks [12], where the trajectories of the satellites are known beforehand, and the message sending is highly restricted by the positions and transmission ranges of the satellites.

The system model is as follows: Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of moving n points on the plane, our radio stations, moving independently, continuously, and at constant speed on a straight line each one. For a mobile station $s_i \in S$, the *transmission range* of s_i, C_{s_i} is a circle of radius $r \geq 0$ centered at s_i. A station s_j can receive a transmission from s_i at time t if and only if $s_j \in C_{s_i}$ at time t. A *transmission range assignment* for S is a function $R : S \to \mathbb{R}^+$ such that the station s_i has assigned a range size of $R(s_i)$.

The following conditions about message transmission are assumed throughout this paper: A message transmission can be completed in an instant of time. If s_i receives or generates a message at time t, then it will pass the message to every station with whom it can communicate at any time $t' \geq t$.

[*]Universidad de Sevilla, Spain,dbanez@us.es
[†]Universidad Nacional Autónoma de México, México. ruy@ciencias.unam.mx, dflores@matem.unam.mx, marco@ciencias.unam.mx, urrutia@matem.unam.mx

Given a source station $s \in S$, and a message M generated in s at time t_0, we are interested in completing a broadcast operation of M to the rest of the stations. This paper studies the following problems:

connectivity problem: Given a transmission range assignment R for S, decide if M arrives to every station in S.

min-equal-range problem: Find the minimum value r needed to perform the broadcast, supposing that each station has a range radius of r.

min-sum problem: Find a transmission range assignment R, such that the broadcast is performed and the sum of the squares of the range radii in R is minimized.

min-sum-binary problem: Compute the minimum number of stations needed to perform the broadcast, assuming that each station can only transmit with range radii either 1 or 0.

The power (energy) needed to correctly transmit data from a station s to a station t depends on the term $d(s,t)^\alpha$, where $d(s,t)$ is the Euclidean distance between s and t, and $\alpha \geq 1$ is the *distance-power gradient* (see [9]). In an ideal environment $\alpha = 2$.

This paper is organized as follows: Section 2 presents an algorithm for solving the connectivity problem. Section 3 proposes an $O(n^3 \log n)$-time algorithm to solve the min-equal-range problem. In Section 4 we prove that min-sum is an NP-hard problem, and discuss the impossibility of finding an algorithm whose approximation ratio is a sub-logarithmic factor from an optimum solution, unless P=NP. Next, in Section 5 we move to min-sum-binary problem, showing that min-sum-binary is also NP-hard, and then propose an approximation algorithm that achieves an approximation factor of $(\ln n + 1)m$, where m is the size of the optimum solution. Finally, we present our conclusions in Section 6.

2 Connectivity

To solve the connectivity problem we propose an algorithm based on Dijkstra's algorithm, which, as a side result, also computes the first time at which each station receives the message.

Given a transmission range assignment R, since two different stations s_i and s_j move along different lines, s_j can lie in C_{s_i} only during one time interval, at which s_i can send a message to s_j. The *transmission interval* of s_j from s_i, $I_{s_i}(s_j)$, is the time interval $[t_a, t_b]$, where $t_a \in \mathbb{R}^+$ is the first instant of time in which s_j lies in C_{s_i}, and $t_b \in \mathbb{R}^+$ is the instant of time in which s_j leaves C_{s_i}.

We suppose that the trajectory of each $s_i \in S$ is given to us in a way that computing the *transmission interval* of two stations can be done efficiently, consequently this operation is assumed to be computed in constant time.

The *connectivity graph* G_R generated by a set of mobile stations S and a transmission range assignment R, is the directed graph having S as vertex set, and there is an arc from s_i to s_j in G_R if and only if $I_{s_i}(s_j)$ is a non empty interval. This arc is labeled with the interval $I_{s_i}(s_j)$. See the left side of Fig. 1 for an example.

If t_i is the first time in which the station s_i receives the message M, then s_i can pass the message to a station s_j if $t_i \leq t_b$, where $[t_a, t_b] = I_{s_i}(s_j)$. This concept can be expressed in G_R in the following way: assign the value t_i to the vertex s_i; consider the arc from s_i to s_j, labeled $[t_a, t_b]$, and assign the time t_j to s_j (the time at which s_j first receives M from s_i), where $t_j = t_i$ if $t_i \in [t_a, t_b]$ or $t_j = t_a$ if $t_i \leq t_a$.

The following algorithm, which we call *IS_CONNECTED*, solves the connectivity problem: Construct the connectivity graph G_R of S; then assign the time t_0 (time at which the source generates the message M) to the vertex s, the value of ∞ to the other vertices and run a modified Dijkstra's algorithm [3] from s, sorting the vertices in the priority queue (of the Dijkstra's algorithm) by the time at which they receive M.

Typical Dijkstra's algorithm assures that the vertex going out of the priority queue has assigned the minimum distance to the source. Thus, *IS_CONNECTED* assures that each vertex going out of the queue has assigned the minimum time at which it receives the message M. Therefore, if the tree obtained from *IS_CONNECTED* is a spanning tree of G_R, then the broadcast from s will succeed (see right side of Fig. 1). As a side result, we obtain the first time at which each vertex receives the message. The correctness and complexity of *IS_CONNECTED* follow from the correctness and complexity of Dijkstra's algorithm. As G_R could be a complete graph, then the total running time of *IS_CONNECTED* is $O(n^2)$ and we arrive to the following result::

Theorem 2.1. *The connectivity problem can be solved in $O(n^2)$ time.*

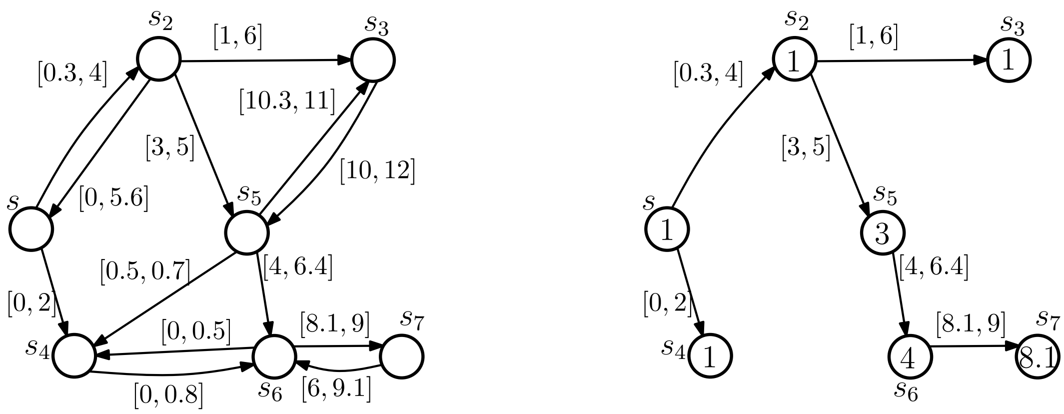

Figure 1: The graph G_R of a mobile network and the spanning tree obtained by the algorithm ($t_0 = 1$).

3 Equally sized ranges

In this section we describe an $O(n^3 \log n)$ algorithm to solve the min-equal-range problem.

As the radius of C_{s_i} is equal to the radius of C_{s_j}, then $I_{s_i}(s_j) = I_{s_j}(s_i)$, so we will use $I_{s_i}(s_j)$ and $I_{s_j}(s_i)$ interchangeably. This fact will transform the connectivity graph into an undirected graph.

For any given $r \geq 0$, we call G_r the connectivity graph of S obtained by assigning to each station the transmission radius r, and T_r the tree obtained from running the algorithm *IS_CONNECTED* on G_r. The problem is then reduced to finding the minimum radius r_{MIN} in which $T_{r_{MIN}}$ is a spanning tree of $G_{r_{MIN}}$.

The key idea of our algorithm is to calculate a discrete set of possible values for r_{MIN}, and then search over those values. We focus on all radii r such that T_r and $T_{r-\epsilon}$ could be different, for all small $\epsilon > 0$, implying that $T_{r-\epsilon}$ might not be a spanning tree of $G_{r-\epsilon}$. We call these radii *critical radii.*

We say that r is a critical radius for S if by assigning r to all stations in S, one of the following cases arises:

a) Two stations in S, s_i and s_j, have only one time t of connection ($I_{s_i}(s_j) = [t, t]$). See Fig. 2 a).

b) Three different stations in S, s_i, s_j and s_k, have the property of $I_{s_i}(s_j) \cap I_{s_i}(s_k) = [t, t]$. See Fig. 2 b).

A radius of type a) corresponds to the addition of an edge in G_r that is not present in $G_{r'}$, with $r' < r$. The type b) corresponds to radii where an edge of T_r is possibly not present in $T_{r'}$, with $r' < r$. The set of all critical radii of S is denoted by $CR(S)$.

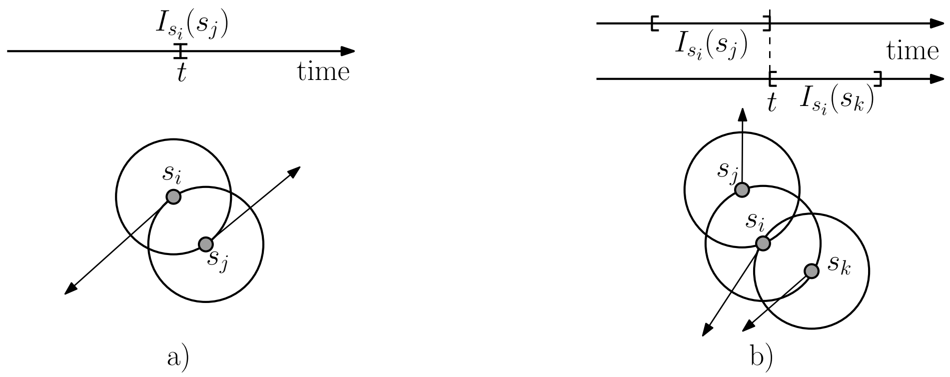

Figure 2: Example of the two cases of critical radius.

Given two different stations, s_i and s_j, let $f_{s_i,s_j}(t)$ be the squared Euclidean distance between s_i and s_j at time t. As s_i and s_j move along lines, f_{s_i,s_j} is a quadratic polynomial in t. Observe that $f_{s_i,s_j} = f_{s_j,s_i}$.

Consider the arrangement of $n-1$ functions involving s_i ($\{f_{s_i,s_j} \mid i \neq j\}$). Any two of these functions intersect at most twice, then the arrangement contains $O(n^2)$ intersections. Each of the $O(n^2)$ intersections gives us a (squared) radius that corresponds to the type b) of critical radii, and each of the $n-1$ function minima gives us a (squared) radius that corresponds to the type a) of critical radii. Since we have n different arrangements, then the size of $CR(S)$ is $O(n^3)$. Assuming that we can: obtain the minima of one of these functions and calculate the intersection of two of these functions in constant time; then we can compute $CR(S)$ in $O(n^3)$ time.

The algorithm *MIN_RADIUS*, for solve the min-equal-range problem, can be defined as follows: Compute the $CR(S)$ set ($O(n^3)$ time), sort the elements in $CR(S)$ ($O(n^3 \log n)$), and look for the minumim radius r_{MIN} in which $T_{r_{MIN}}$ is a spanning tree of $G_{r_{MIN}}$, by using binary search and applying the *IS_CONNECTED* algorithm ($O(n^2)$) at each step. The total running time is then $O(n^3 \log n)$.

In summary, we have proven the following result:

Theorem 3.1. *The min-equal-range problem can be solved in $O(n^3 \log n)$ time.*

4 The general min-sum problem

This section focus on showing that the general version of min-sum problem is intractable (unless P=NP), by reducing the well-known weighted-set-cover problem [1, 3] to the min-sum problem.

An instance of the weighted-set-cover problem consists of a finite set A; a family B of subsets of A, such that every element of A belongs to at least one subset in B; and a weight function $w : B \to \mathbb{R}^+$. We say that a subset $c \in B$ *covers* the elements of A with weight $w(c)$. The problem is to find a minimum-weight *cover* $\mathcal{C} \subseteq B$ whose members cover all of A, where the weight of $\mathcal{C}$ is $\sum_{c \in \mathcal{C}} w(c)$.

It is known that the decision version of weighted-set-cover problem is NP-complete and the optimization (minimization) version is NP-hard [1, 3]. We can prove the NP-hardness of the min-sum problem by reducing the weighted-set-cover problem to the min-sum problem.

Take an instance of the weighted-set-cover problem: $A = \{a_1, a_2, \ldots, a_k\}$ (the covered set), $B = \{b_1, b_2, \ldots, b_l\}$ (the covering set) and the weight function $w : B \to \mathbb{R}^+$. The transformation to an instance of the min-sum problem is as follows (see Figure 3):

Take a set of stations $SB = \{sb_1, sb_2, \ldots, sb_l\}$, collinear in a horizontal line $\mathcal{L}$, at distance δ (large enough) from each other, and moving leftwards with the same speed. We assign each station sb_i of SB to the element b_i in B. The source s, that generates the message M at time t_0, moves rightwards in a line parallel to $\mathcal{L}$ whose distance to $\mathcal{L}$ is 1, in a way that after a certain time, say $t_1 > t_0$, s could have transmitted the message M to each station of SB if its transmission radius were equal to 1.

Now let $SA = \{sa_1, sa_2, \ldots, sa_k\}$ be a set of static points in the plane far away from the trajectories of SB and s, that we call the *meeting points*. Each element a_i in A is assigned to the point sa_i of SA.

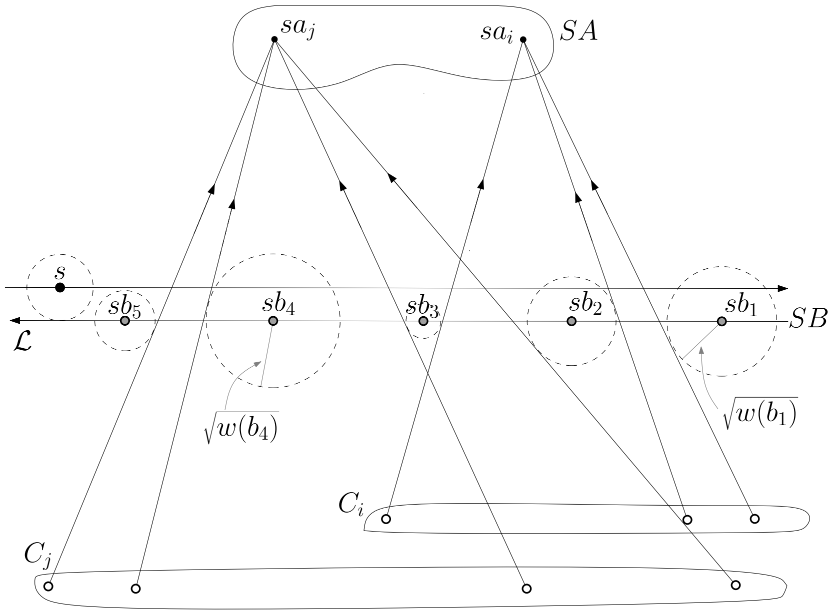

Figure 3: Reduction of weighted-set-cover problem to min-sum problem.

Finally we create the *cover relation* by adding station sets $C = C_1 \cup C_2 \cup \ldots \cup C_k$ in the following way: Take an element a_i in A and take B_i as the set of elements in B that covers a_i. For each b_l in B_i we create a new station $c_{i,l}$ in the set C_i, in a way that: $c_{i,l}$ passes at distance $\sqrt{w(b_l)}$ from sb_l at time $t_{i,l} > t_1$, passes over sa_i at time $t_{a_i} > t_{i,l}$, and at any time the minimum distance between sb_l and $c_{i,l}$ is $\sqrt{w(b_l)}$. Intuitively, every station sb_l that *covers* sa_i sends a station to sa_i, and all stations in C_i arrive to sa_i at the same time (t_{a_i}). Notice that we can take suitable times and distances so all the events occur independent from each other.

In some sense we only permit that each element $sb_l \in SB$ have two choices, transmit with radius $\sqrt{w(b_l)}$ or 0. On the other hand, all the stations in C_i can receive M at time t_{a_i} if one of them has it and transmit with radius 0.

Note that the above transformation can be done in polynomial time. A range assignment R that minimizes the sum of the squared radii and allows a broadcast from s, is one that: assign radius 1 to s; takes $\mathcal{D} \subseteq SB$ such that $\sum_{sb_l \in \mathcal{D}} R(sb_l)^2$ is minimized, where $sb_l \in \mathcal{D}$ has assigned a radius $R(sb_l) = \sqrt{w(b_l)}$; leaves the stations of $SB \setminus \mathcal{D}$ with radius 0; and assures that at least one of the elements in each C_i have the message M. But this assignment maps to a subset of B of minimum weight that cover A, solving then the weighted-set-cover problem. Since weighted-set-cover problem can only be approximated to within a $\ln |A| + 1$ factor (unless P=NP), no polynomial time approximation algorithm for min-sum problem achieves a smaller approximation factor. The following result can be then established:

Theorem 4.1. *The min-sum problem is NP-hard and, unless P=NP, it is not approximable within a sub-logarithmic factor.*

5 The min-sum-binary problem

In this section we first prove that the min-sum-binary problem also belongs to the NP-hard complexity class, and then we describe an approximation approach based on the well-known greedy algorithm for the weighted-set-cover problem [1, 3].

5.1 NP-hardness

A similar proof to that of the NP-hardness of min-sum, shows that the min-sum-binary problem is NP-hard. By taking an instance of the weighted-set-cover problem, but with all the weights of the elements in B set to 1, we obtain an instance of the classic set-cover problem [7], In the set-cover problem we need to find a minimum-size subset of elements of B whose members cover all of A. The set-cover problem is also NP-hard and approximable within a $\ln |A| + 1$ factor (unless P=NP) [7, 3, 6].

The same construction used in the previous section can be used to reduce the set-cover problem to the min-sum-binary problem, by assigning radius 1 or 0 to the elements of SB.

5.2 Approximation algorithm

As we cannot obtain a polynomial time approximation algorithm (for the min-sum-binary problem) that achieves a sub-logarithmic approximation ratio, then we propose an approach based on a greedy algorithm for the weighted-set-cover problem [1, 3] whose approximation ratio is $\ln |A| + 1$.

Roughly speaking, at each step of the greedy algorithm, the subset that covers the most elements with the lowest weight is chosen. Specifically, for every subset the number of elements it covers (not yet covered) is divided by its weight. The subset with the highest ratio is then chosen.

In order for s_i to be able to transmit the message M to another station s_j, then s_i must have received M before the transmission interval between them ends. Then for each station s_i we consider the times $t_{i,1} \leq t_{i,2} \leq \cdots \leq t_{i,k_i}$ at which s_i loses connectivity with a station (with $k_i \leq n-1$, as we only consider those stations that are ever within transmission range of s_i). We denote by $s_{i,j}$, the set of stations that lose connectivity with s_i at or after time $t_{i,j}$. By definition, $s_{i,j+1} \subset s_{i,j}$.

We construct a instance of the weighted-set-cover problem. The set $A = S$ of all stations is the set to be covered. As the family B of subsets used to cover S, we use the subsets $s_{i,j}$. We define the weight function $w : B \to \mathbb{R}^+$ such that $w(s_{i,j})$ is the minimum number of stations, including s_i, that must be turned on (have transmission range equal to 1) in order for s_i to receive the message at or before time $t_{i,j}$.

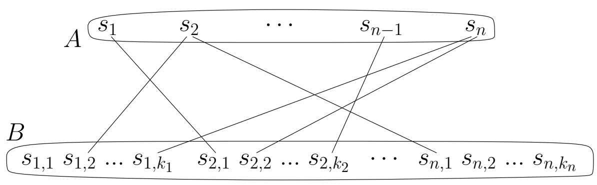

Figure 4: Transforming min-sum-binary problem to weighted-set-cover problem (edges represent the cover relation).

Using the greedy algorithm for weighted set cover at each step we choose the subset $s_{i,j}$ with the best ratio between new stations covered and $w(s_{i,j})$.

Denote by $\mathcal{D}'$ the set of stations (and times) $s_{i,j}$ chosen by the greedy algorithm. It is known that the greedy solution is at most at a $\ln |A| + 1$ factor of the optimal solution of the weighted set cover problem. In particular it is at most at a $\ln |A| + 1$ factor from any other solution.

Consider an optimal solution $\mathcal{D}_{\mathcal{O}}$ for the min-sum-binary problem. Let $s_i \in \mathcal{D}_{\mathcal{O}}$. Assume s_i receives the message for the first time at time t. Replace s_i with $s_{i,j}$, where $t_{i,j}$ is the first moment in time after t at which s_i loses connectivity with another station. Doing this for every element $s_i \in \mathcal{D}_{\mathcal{O}}$, we obtain a covering set ${\mathcal{D}_{\mathcal{O}}}'$ for our weighted instance of set cover. Therefore:

$$(\ln |A| + 1) \left(\sum_{s_{i,j} \in {\mathcal{D}_{\mathcal{O}}}'} w(s_{i,j}) \right) \geq \sum_{s_{i,j} \in \mathcal{D}'} w(s_{i,j}) \tag{1}$$

From $\mathcal{D}'$ we obtain a solution for the min-sum-binary problem. For every $s_{i,j}$: we turn on the corresponding s_i; then we turn on the other $w(s_{i,j}) - 1$ stations that enable s_i to receive the message at or before time $t_{i,j}$. Denote by $\mathcal{D}$ the set of stations turned on in this manner.

By construction of $\mathcal{D}$, we have that $|\mathcal{D}| \leq \sum_{s_{i,j} \in \mathcal{D}'} w(s_{i,j})$. Also for every $s_{i,j} \in \mathcal{D_O}'$, its cost cannot be greater than $|\mathcal{D_O}|$. Therefore $\sum_{s_{i,j} \in \mathcal{D_O}'} w(s_{i,j}) \leq |\mathcal{D_O}||\mathcal{D_O}|$ and together with (1), we have:

$$((\ln |A| + 1)|\mathcal{D_O}|)\, |\mathcal{D_O}| \geq |\mathcal{D}| \tag{2}$$

As $|A| = n$, our greedy algorithm thus yields an approximation ratio of $(\ln n + 1)m$, where m is the size of an optimal solution for the min-sum-binary problem.

We now describe how to obtain the weights $w(s_{i,j})$. For the time being, assume that all stations are turned on. For any moment in time t let G_t be the graph with S as vertex set and two stations adjacent if they are currently within transmission range of each other.

Let t_0 be the moment in time at which the source receives the message for the first time. Let $t_1 \leq t_2 \leq ... \leq t_p$ be the moments in time at which communication is gained or lost between any two stations.

Consider the sequence of graphs $G_{t_0}, G_{t_1}, \ldots, G_{t_p}$. Two consecutive graphs in the sequence differ by an edge, that is either removed or added. Also since two stations meet each other at most once, the sequence has at most $O(n^2)$ terms.

The minimum number of stations that need to be turned on for s_i to receive the message at time t_l or before is none other than the minimum number of times that the message has been transmitted to reach s_i before or at time t_l.

Starting from the source in G_{t_0} we apply Dijkstra's algorithm and compute the distance of every station to the source. This distance (plus one) is the number of hops that the message must have made to reach a given station. For each station we keep track of: the number of hops, the path of stations and the time at which this was achieved.

We move from graph to graph in sequence. When we move to the next graph G_{t_l} and a new edge is added, a station may be able to receive the message in fewer hops. Suppose that the added edge was $\{s_i, s_j\}$ and that, say s_i received the message in h hops, where as s_j received the message in more than $h+1$ hops. s_j can now receive the message from s_i in fewer hops. Also any station currently adjacent to s_j might receive the message in fewer hops and so on. We compute this new values (keeping the old ones for further reference) by applying again Dijkstra's algorithm on the new graph G_{t_l}, with s_i as the starting vertex; with the exception that we relax only the edges that leave vertices that now receive the message in fewer hops. The distance of a station to s_i plus the number of hops used to reach s_i is the number of hops that can now be used to reach the station. This number may or may not be less than the current number of hops. If it is less, we add a new entry with the number of hops, time at which this was achieved and the path used. If linked lists are used to represent paths, this can be done in constant time per station. If the next graph comes from the deletion of an edge, we delete this edge and move on to the next graph.

Once we have visited all graphs in the sequence, every vertex contains a decreasing list of number of hops, time and path. Using this list we can calculate the costs of each $s_{i,j}$ at a linear cost per station, thus $O(n^2)$ in total.

Finally since we have an instance of the weighted-set-cover problem with a set of n elements and a family of $O(n^2)$ subsets, the greedy algorithm can be implemented to run in time $O(n^4)$. Thus the approximation algorithm runs in $O(n^4)$ time in total. We end this discussion by stating the following theorem:

Theorem 5.1. *Given a set of n mobile stations, a $(\ln n + 1)m$-approximation of the min-sum-binary problem can be computed in $O(n^4)$ time, where m is the size of the optimum solution.*

6 Conclusions

In this paper, we worked on the problem of energy-efficient source-initiated broadcasting in a mobile wireless network, where every station moves at constant velocity along a linear trajectory. First, we showed that the problem of minimizing the sum of the squares of the range radii for a broadcast operation is polynomial, when all stations have the same radius, and it is NP-hard in the general setting. We then presented a polynomial-time approximation algorithm for a special case of the min-sum problem for which the stations transmit with radius either 1 or 0. Our approach performs approximation within a factor $(\ln n + 1)m$ of the optimal, where m is the size of the optimum solution.

It is worth mention that we never use the fact that the stations move at the same velocity, consequently, our results follows even when the speeds of the stations are different, as long as we forbid infinite speeds. On the other hand, the complexity of the problem does not change if we assume algebraic motions instead of rectilinear motions, as long as intersections and distances calculation between two trajectories can be computed efficiently.

The proof of the NP-hardness of the min-sum problem can be easily modified to proof the NP-hardness of any problem whose goal is to minimize the sum of a power of the range radii, which in terms of energy, makes these problems intractable (unless P=NP) no matter which *distance-power gradient* $\alpha \geq 1$ we choose to optimize, not only for $\alpha = 2$ (see [9]). This fact contrast with the static case of the problem, which is solvable in polynomial time for $\alpha = 1$ (see [2]).

Finally, we must note that the *IS_CONNECTED* algorithm can be used to obtain a simple heuristic method for the min-sum problem, by applying a tabu search strategy [8].

References

[1] V. Chvatal. *A greedy heuristic for the set covering problem.* Math. Oper. Res., Vol. 4, 233–235, 1979.

[2] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, P. Vocca. *On the Complexity of Computing Minimum Energy Consumption Broadcast Subgraphs.* In Proc. of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS-01), LNCS 2010, 121–132, 2001.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest. *Introduction to Algorithms.* MIT Press, McGraw-Hill, second edition, 2001.

[4] G. K. Das, S. Das, S. C. Nandy. *Range assignment for energy efficient broadcasting in linear radio networks.* Theoretical Computer Science, Vol. 352, 332–341, 2006.

[5] D. Estrin, R. Govindan, J. Heidemann, S. Kumar. *Next century challenges: scalable coordination in sensor networks.* Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, 263 – 270, 1999.

[6] U. Feige. *A Threshold of ln n for Approximating Set Cover.* Journal of the ACM (JACM), Vol. 45 n.4, 634 – 652, 1998.

[7] M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, 1979.

[8] F. Glover, M. Laguna. *Tabu Search.* Kluwer Academic, 1997.

[9] K. Pahlavan, A. Levesque. *Wireless information networks.* Wiley-Interscience, 1995.

[10] D. Peleg. *Time-Efficient Broadcasting in Radio Networks: A Review.* In Proc. of the 4th Int. Conf. on Distributed Computing and Internet Technology (ICDCIT), 2007.

[11] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, S. Roundy. *PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking.* Computer, Vol. 33, 42–48, IEEE Comp. Soc. Press, 2000.

[12] M.G.C. Resende, P.M. Pardalos. *Handbook of Optimization in Telecommunications.* Springer, 2006.