Algorithmica (2003) 35: 95–110
DOI: 10.1007/s00453-002-0985-2

Algorithmica

The Minimum Range Assignment Problem on Linear Radio Networks[1]

Andrea E. F. Clementi,[2] Afonso Ferreira,[3] Paolo Penna,[2] Stéphane Perennes,[3]
and Riccardo Silvestri[4]

Abstract. Given a set S of radio stations located on a line and an integer $h \geq 1$, the MIN ASSIGNMENT problem consists in finding a range assignment of minimum power consumption provided that any pair of stations can communicate in at most h hops. Previous positive results for this problem are only known when $h = |S| - 1$ or in the uniform chain case (i.e., when the stations are equally spaced). As for the first case, Kirousis et al. [7] provided a polynomial-time algorithm while, for the second case, they derive a polynomial-time approximation algorithm.

This paper presents the first polynomial-time, approximation algorithm for the MIN ASSIGNMENT problem. The algorithm guarantees a 2-approximation ratio and runs in $O(hn^3)$ time. We also prove that, for fixed h and for "well spaced" instances (a broad generalization of the uniform chain case), the problem admits a *polynomial-time approximation scheme*. This result significantly improves over the approximability result given by Kirousis et al.

Both our approximation results are obtained via new algorithms that exactly solve two natural variants of the MIN ASSIGNMENT problem: the problem in which every station must reach a fixed one in at most h hops and the problem in which the goal is to select a subset of *bases* such that all the other stations must reach one *base* in at most $h - 1$ hops.

Finally, we show that for $h = 2$ the MIN ASSIGNMENT problem can be exactly solved in $O(n^3)$ time.

Key Words. Ad hoc packet radio networks, Approximation algorithms, Dynamic programming.

1. Introduction. An *Ad-Hoc Packet Radio Network* [10] (in short, radio networks) is a finite set of radio stations located on a geographical region that are able to communicate by transmitting and receiving radio signals. In such networks, a transmission range is assigned to each station s and any other station t within this range can directly (i.e., within one *hop*) receive messages from s. Communication between two stations that are not within their respective ranges can be achieved by *multi-hop* transmissions. One of the main benefits of (multi-hops) Ad-Hoc networks is the reduction of the power consumption. This can be obtained by suitably varying the transmission ranges.

---

[1] A preliminary version of this paper has been presented at *ESA* '00. This work was partially supported by the RTN Project ARACNE and the Italian MURST Project REACTION. Part of this work has been done while the third author was visiting the research center of INRIA Sophia Antipolis.
[2] Dipartimento di Matematica, Università di Roma "Tor Vergata", Via della Ricerca Scientifica, 00133 Rome, Italy. {clementi,penna}@mat.uniroma2.it.
[3] MASCOTTE Project, I3S/INRIA Sophia Antipolis, CNRS/INRIA/Université de Nice-Sophia Antipolis, 2004 route des lucioles, B.P.93, F 06902 Sophia Antipolis, France. {ferreira,perennes}@sophia.inria.fr.
[4] Dipartimento di Scienze dell'informazione, Università di Roma "La Sapienza", Via Salaria 113, I00198 Roma, Italy. silver@dsi.uniroma1.it.

Received August 22, 2001; revised April 5, 2002. Communicated by S. Albers.
Online publication October 29, 2002.

It is reasonably assumed [10] that the power P_t required by a station t to transmit data correctly to another station s must satisfy the inequality

$$\frac{P_t}{d(t,s)^\beta} > \gamma, \tag{1}$$

where $d(t,s)$ is the distance between t and s, $\beta \geq 1$ is the *distance-power gradient*, and $\gamma \geq 1$ is the *transmission-quality* parameter. In an ideal environment, i.e., in the empty space, $\beta = 2$ but it may vary from 1 to more than 6 depending on the environmental conditions of the place where the network is located (see [10]). In the rest of the paper we fix $\beta = 2$ and $\gamma = 1$, however, our results can be easily extended to any $\beta, \gamma \geq 1$. Given a set $S = \{s_1, \ldots, s_n\}$ of radio stations on the D-dimensional Euclidean space, a *range assignment* for S is a function $r\colon S \to \mathcal{R}^+$ (where $\mathcal{R}^+$ is the set of non-negative real numbers), and the *cost* of r is defined as

$$\mathsf{cost}(r) = \sum_{i=1}^{n} r(s_i)^2.$$

Given an integer h and a set S of stations on the D-dimensional Euclidean space (D ≥ 1), the MIN D-DIM ASSIGNMENT problem consists in finding a minimum cost range assignment provided that the assignment ensures the communication between any pair of stations in at most h hops. The MIN 1-DIM ASSIGNMENT problem (i.e., the *linear case*) is simply denoted as MIN ASSIGNMENT. In this work we focus on the linear case, i.e., networks that can be modeled as sets of stations located along a line. As pointed out in [9], rather than a simplification, this version of the problem results in a more accurate analysis of the situation arising, for instance, in vehicular technology applications. It is common opinion to consider one-dimensional frameworks as in fact the most suitable ones in studying road traffic information systems [2], [5], [8], [9]: vehicles follow roads, and messages are to be broadcast along lanes. Typically, the curvature of roads is small in comparison with the transmission range (half a mile up to some few miles).

1.1. *Previous Results*. Tradeoffs between connectivity and power consumption have been obtained in [4], [7], [11], and [12].

As for the MIN D-DIM ASSIGNMENT problem, several results have been obtained for the *unbounded case* (i.e., when $h = n - 1$). Under this restriction, MIN 3-DIM ASSIGNMENT is APX-complete (in [7] a polynomial-time 2-approximation[5] algorithm is given, while in [3] the APX-hardness is proved); the MIN 2-DIM ASSIGNMENT problem is NP-hard [3]; finally, the MIN ASSIGNMENT problem is in P via an $O(n^4)$-time algorithm [7]. On the other hand, few results are known for the general case (i.e., for arbitrary h). In [7] the following bounds on the optimal cost have been proved for a strong restriction of MIN ASSIGNMENT.

THEOREM 1 (The Uniform Chain Case [7]). *Let N be a set of n points equally spaced at distance $\delta > 0$ on the same line*; *let $\mathsf{OPT}_h(N)$ be the cost of an optimal solution for*

[5] An algorithm for a minimization problem is said to be c-approximating if it returns a solution of cost at most c times the optimum; The value c is called the approximation ratio.

MIN ASSIGNMENT *on input* h *and* N. *Then it holds that*

1. $\mathsf{OPT}_h(N) = \Theta(\delta^2 n^{(2^{h+1}-1)/(2^h-1)})$, *for any fixed positive integer* h;
2. $\mathsf{OPT}_h(N) = \Theta(\delta^2 (n^2/h))$, *for any* $h = \Omega(\log n)$.

Furthermore, the two above (implicit) upper bounds can be efficiently constructed.

Although the constructive method of Theorem 1 yields approximate solutions for the uniform chain case, no approximation result is known for more general configurations. Moreover, the approximation ratio guaranteed by the first result of Theorem 1 *increases with* h. We then observe that for *non-constant* values of h, such that $h = o(\log n)$, no approximation algorithm *even for the uniform chain* restriction is known.

As for the MIN 2-DIM ASSIGNMENT problem, some upper and lower bounds on the optimal cost function, for constant values of h, have been derived in [4].

1.2. *Our Results*. We present a polynomial-time algorithm for the MIN ASSIGNMENT problem that guarantees a 2-approximation ratio and runs in $O(hn^3)$ time. To the best of our knowledge, this is the first polynomial-time algorithm achieving a *constant*, worst-case approximation ratio for this problem.

Then we provide a better approximation algorithm that works on any family of *well spaced* instances and for any constant h; in such instances, the ratio between the maximum and the minimum distance among adjacent stations is bounded by a polylogarithmic function of n (see Section 5.2 for a formal definition). More precisely, we show that, for any well spaced instance and for any constant h, it is possible to compute in $O(hn^3)$ time a solution whose cost is at most $(1 + \varepsilon(n))$ times the optimum, where $\varepsilon(n) = o(1)$. Since uniform chains are a (very strong) restriction of well spaced instances, our result strongly improves that in Theorem 1 in the case $h = O(1)$. Indeed, the obtained approximation ratio tends to 1 (while, as already observed, the approximation ratio achieved by Theorem 1 is an increasing function of h).

Our approximability results are obtained by exploiting exact solutions for two natural variants of the MIN ASSIGNMENT problem that may have independent interest:

MIN ALL-TO-ONE ASSIGNMENT. Given a set S of stations on the line, a *sink* station $t \in S$, and an integer $h > 0$; find a minimum cost range assignment for S ensuring that any station is able to reach t in at most h hops.

MIN ASSIGNMENT WITH BASES. Given a set S of stations on the line, and an integer $h > 0$; find a minimum cost range assignment for S such that any station in S is either a *base* (a station is a base if it directly reaches any other station in S) or it reaches a base in at most $h - 1$ hops.

For each of the two above problems, we provide an algorithm, based on dynamic programming, that returns an optimal solution in $O(hn^3)$ time.

Finally, we prove that for $h = 2$, the MIN ASSIGNMENT problem can be solved in $O(n^3)$ time. This result is obtained by combining the algorithm for the MIN ASSIGNMENT WITH BASES problem with a simple characterization of the structure of any optimal two-hops range assignment.

Organization of the Paper. In Section 2 we provide some basic definitions and notation. An efficient solution for the MIN ALL-TO-ONE ASSIGNMENT and the MIN ASSIGNMENT WITH BASES problem is given in Sections 3 and 4, respectively. The approximability results are defined in Section 5. In Section 6 we describe an exact algorithm for the case $h = 2$ and, in Section 7, we discuss some open problems.

2. Preliminaries.

Let $S = \{s_1, \ldots, s_n\}$ be a set of n consecutive stations located on a line. We denote by $d(i, j)$ the distance between s_i and s_j. We define

$$\begin{aligned} \delta_{\min}(S) &= \min\{d(i, i+1) \mid 1 \le i \le n-1\}, \\ \delta_{\max}(S) &= \max\{d(i, i+1) \mid 1 \le i \le n-1\}, \\ D(S) &= d(1, n). \end{aligned}$$

Given a range assignment $r\colon S \to \mathcal{R}^+$, we say that s_i directly (i.e., in one hop) reaches s_j if $r(s_i) \ge d(i, j)$ (in short $i \to_r j$). Additionally, s_i reaches s_j in at most h hops if there exist $k \le h - 1$ stations $s_{i_1}, \ldots, s_{i_k}$ such that

$$i \to_r i_1 \to_r i_2, \ldots, \to_r i_k \to_r j \qquad \text{(in short } i \to_{r,h} j\text{)}.$$

We omit the subscript r when this will be clear from the context.

We say that r is an *h-assignment* $(1 \le h \le n-1)$ if for any pair of stations s_i and s_j, $i \to_{r,h} j$. Notice that h-assignments are exactly the feasible solutions for the instance (h, S) of MIN ASSIGNMENT. The cost of an optimal h-assignment for a given set S of stations is denoted as $\mathsf{OPT}_h(S)$.

We denote the station s_i simply as i. Finally, we denote the set of stations $\{s_{i+1}, \ldots, s_{j-1}\}$ by (i, j) and we also use $[i, j]$, $[i, j)$, and $(i, j]$ as a shorthand of $(i-1, j+1)$, $(i-1, j)$, and $(i, j+1)$, respectively.

3. The MIN ALL-TO-ONE ASSIGNMENT Problem.

In this section we present an efficient method for the MIN ALL-TO-ONE ASSIGNMENT problem which is based on a suitable use of dynamic programming. To this aim, we introduce the following functions that will also be used in Section 4 in order to solve the MIN ASSIGNMENT WITH BASES problem.

DEFINITION 1 (ALL (to one)). For any set S of n stations and for any $1 \le i \le j \le n$, we define

$$\begin{aligned} \overrightarrow{\mathsf{ALL}}_h(i, j) &= \min\{\mathsf{cost}(r) \mid \forall k \in [i, j],\ \ k \to_{r,h} j\}; \\ \overleftarrow{\mathsf{ALL}}_h(i, j) &= \min\{\mathsf{cost}(r) \mid \forall k \in [i, j],\ \ k \to_{r,h} i\}; \\ \overrightarrow{\mathsf{ALL}}^*_h(i, j) &= \min\{\mathsf{cost}(r) \mid \forall k \in (i, j],\ \ k \to_{r,h} j \wedge\ i \to_r j\}; \\ \overleftarrow{\mathsf{ALL}}^*_h(i, j) &= \min\{\mathsf{cost}(r) \mid \forall k \in [i, j),\ \ k \to_{r,h} i\ \wedge\ j \to_r i\}. \end{aligned}$$

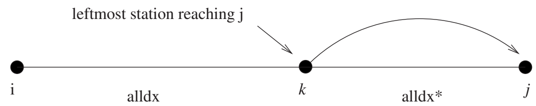

Fig. 1. The proof of Lemma 1.

DEFINITION 2 (OR). For any set S of n stations and for any $1 \le i \le j \le n$, we define

$$\mathsf{OR}_h(i, j) = \min\{\mathsf{cost}(r) \mid \forall k \in [i, j],\ \ k \rightarrow_{r,h} i \ \vee\ k \rightarrow_{r,h} j\}.$$

The following three lemmas provides a useful characterization of the optimal solutions for the functions defined above.

LEMMA 1. *For any set S of n stations on the line, for any* $1 \le i \le j \le n$, *and for any* $h \ge 1$, *the following recursive equations hold*:

$$\overrightarrow{\mathsf{ALL}}_h(i, j) = \min_{i \le k < j}\{\overrightarrow{\mathsf{ALL}}^*_h(k, j) + \overrightarrow{\mathsf{ALL}}_{h-1}(i, k)\}; \tag{2}$$

$$\overleftarrow{\mathsf{ALL}}_h(i, j) = \min_{i < k \le j}\{\overleftarrow{\mathsf{ALL}}^*_h(i, k) + \overleftarrow{\mathsf{ALL}}_{h-1}(k, j)\}. \tag{3}$$

PROOF. Consider the function $\overrightarrow{\mathsf{ALL}}_h(i, j)$ and consider any feasible range assignment r for this function. Let k be the index of the *leftmost* station reaching j in one hop (see Figure 1). For any station $s \in [i, k)$ it holds that $s \rightarrow_{r,h} j$ but it does not hold that $s \rightarrow_r j$ (by definition of k). It thus easily follows that, for any $s \in [i, k)$, it must be the case that $s \rightarrow_{r,h-1} k$. We also remark that no station in $[i, k]$ uses "bridges" in the interval $(k, j]$. This implies that r, restricted to $[i, k)$, is a feasible range assignment for $\overrightarrow{\mathsf{ALL}}_{h-1}(i, k)$. Furthermore, for any $s \in [k, j]$, $s \rightarrow_{r,h} j$ without using any "bridge" in $[i, k)$. Hence, r restricted to $[k, j]$ is a feasible assignment for $\overrightarrow{\mathsf{ALL}}^*_h(k, j)$. It thus follows that

$$\mathsf{cost}(r) \ \ge\ \overrightarrow{\mathsf{ALL}}_{h-1}(i, k) + \overrightarrow{\mathsf{ALL}}^*_h(k, j).$$

Equation (3) can be proved by a symmetric argument. □

LEMMA 2. *For any set S of n stations on the line, for any* $1 \le i \le j \le n$, *and for any* $h \ge 1$, *the following recursive equations hold*:

$$\overrightarrow{\mathsf{ALL}}^*_h(i, j) = \min_{i < k \le j}\{\overrightarrow{\mathsf{ALL}}^*_h(k, j) + \mathsf{OR}_{h-1}(i, k)\} + d(i, j)^2; \tag{4}$$

$$\overleftarrow{\mathsf{ALL}}^*_h(i, j) = \min_{i \le k < j}\{\overleftarrow{\mathsf{ALL}}^*_h(i, k) + \mathsf{OR}_{h-1}(k, j)\} + d(i, j)^2. \tag{5}$$

PROOF. Consider the function $\overrightarrow{\mathsf{ALL}}^*_h(i, j)$ and consider any feasible range assignment r for this function. Let k be the index of the *leftmost* station, other than i, reaching j in one hop. For any station $s \in (i, k)$, it holds that $s \rightarrow_{r,h} j$ but it does not hold that $s \rightarrow_r j$

(by definition of k). It thus easily follows that, for any $s \in [i, k)$, it must be the case that $s \to_{r,h-1} k$ or $s \to_{r,h-1} i$. We also remark that no station in $[i, k]$ uses "bridges" in the interval $(k, j]$. This implies that r, restricted to (i, k), is a feasible range assignment for $\mathsf{OR}_{h-1}(i, k)$. Furthermore, for any $s \in [k, j]$, $s \to_{r,h} j$ without using any "bridge" in $[i, k)$. Hence, r restricted to $[k, j]$ is a feasible assignment for $\overrightarrow{\mathsf{ALL}}_h^*(k, j)$. It thus follows that

$$\mathsf{cost}(r) \geq \overrightarrow{\mathsf{ALL}}_h^*(k, j) + \mathsf{OR}_{h-1}(i, k) + d(i, j)^2.$$

Equation (5) can be proved by a symmetric argument. □

LEMMA 3. *For any set S of n stations on the line, for any $1 \leq i \leq j \leq n$, and for any $h \geq 1$, the following recursive equation holds*:

$$\mathsf{OR}_h(i, j) = \min_{i \leq k < j} \{\overleftarrow{\mathsf{ALL}}_h(i, k) + \overrightarrow{\mathsf{ALL}}_h(k+1, j)\}. \tag{6}$$

PROOF. We consider a feasible range assignment for the function $\mathsf{OR}(i, j)$. By contradiction, suppose that (6) does not hold. Hence, there must exist two stations l and r such that: (a) $i < l < r < j$; (b) $l \to_{r,h} j$ and $r \to_{r,h} i$; and (c) $l \not\to_{r,h} i$ and $r \not\to_{r,h} j$.

We fix two arbitrary sequences $l_1, \ldots, l_a$ and $r_1, \ldots, l_b$ of stations such that

$$l \to_r l_1 \to_r l_2 \to_r \cdots \to_r l_a \to_r j,$$

and

$$r \to_r r_1 \to_r r_2 \to_r \cdots \to_r r_b \to_r i,$$

with $a, b \leq h - 1$. We first observe that it must hold that l_a is on the right of r_b, since from the hypothesis $l_a \not\to_r i$ and $r_b \not\to_r j$. This implies that there exists a k with $1 \leq k < \min\{a, b\}$ such that

1. l_{a-k} is on the left of r_{b-k};
2. l_{a-k+1} is on the right of r_{b-k+1}.

Notice that the above two conditions imply that $r_{b-k} \to_r l_{a-k+1}$. Hence, we obtain

$$r \to_{r,b-k} r_{b-k} \to_r l_{a-k+1} \to_{r,k} j,$$

thus implying $r \to_{r,b+1} j$. Since $a \leq h - 1$, we have a contradiction with the hypothesis $r \not\to_{r,h} j$. □

THEOREM 2. *There is an algorithm that, for any set of n stations on the line, for any $1 \leq i \leq j \leq n$, and for any $h \geq 1$, computes $\overleftarrow{\mathsf{ALL}}_h(i, j)$, $\overrightarrow{\mathsf{ALL}}_h(i, j)$, $\mathsf{OR}_h(i, j)$ in $O(hn^3)$ time.*

PROOF. In order to prove the theorem, we describe the correct "crossed" recursive computation that will return the outputs of the five functions in Lemmas 1–3. The overall computation goes over *h phases*; in the ℓth phase, all the functions will be computed for ℓ hops.

Phase $\ell = 1$ consists of computing the following values:

$$\forall i < j: \qquad \overleftarrow{\mathsf{ALL}}_1^*(i, j) = \overleftarrow{\mathsf{ALL}}_1(i, j) = \sum_{k=i+1}^{j} d(i, k)^2,$$

$$\forall i < j: \qquad \overrightarrow{\mathsf{ALL}}_1^*(i, j) = \overrightarrow{\mathsf{ALL}}_1(i, j) = \sum_{k=i}^{j-1} d(k, j)^2,$$

$$\mathsf{OR}_1(i, j) = \min_{i \le k < j} \{\overleftarrow{\mathsf{ALL}}_1(i, k) + \overrightarrow{\mathsf{ALL}}_1(k + 1, j)\}.$$

Notice that, for any $\ell \geq 1$ and for any i, it easily holds that

$$\overleftarrow{\mathsf{ALL}}_\ell^*(i, i) = \overleftarrow{\mathsf{ALL}}_\ell(i, i) = \overrightarrow{\mathsf{ALL}}_\ell^*(i, i) = \overrightarrow{\mathsf{ALL}}_\ell(i, i) = \mathsf{OR}_\ell(i, i) = 0.$$

Now, assume that, at the end of Phase $\ell - 1$, the algorithm has computed the values of the five functions for all possible segments in $[1, n]$ and for $\ell - 1$ hops. Then the function $\overleftarrow{\mathsf{ALL}}_\ell^*$ can be computed, by applying (5), for all the segments in the following order:

$$[1, 1], [1, 2], \ldots, [1, n], [2, 2], [2, 3], \ldots, [2, n], \ldots, [n - 1, n], [n, n].$$

The opposite order is instead used for computing the values of $\overrightarrow{\mathsf{ALL}}_\ell^*$ by applying (4). The next two steps (in any order) are the computations of functions $\overrightarrow{\mathsf{ALL}}_\ell$ and $\overleftarrow{\mathsf{ALL}}_\ell$ for any interval in $[1, n]$ by applying (2) and (3). The last values computed at Phase ℓ are the $\mathsf{OR}_\ell(i, j)$ for all possible segments (i, j) according to (6).

We finally observe that, at every phase ℓ, we need to compute $O(n^2)$ values, each of them requiring $O(n)$ time. □

The above lemma easily implies the following:

COROLLARY 1. *The* MIN ALL-TO-ONE ASSIGNMENT *problem can be solved in* $O(hn^3)$ *time*.

4. The MIN ASSIGNMENT WITH BASES Problem.

In order to provide exact solutions for the MIN ASSIGNMENT WITH BASES problem, we consider the following definitions.

DEFINITION 3 (Base Stations). Let r be a feasible solution for the MIN ASSIGNMENT problem on input h and S. A station i is a *base* (in short B) if $i \to_r 1 \wedge i \to_r n$. Moreover, r is of type B^* if there is at least one base and any station which is not a base reaches some base in at most $h - 1$ hops. Then $\mathsf{BASES}_h(S)$ denotes the cost of an optimal assignment of type B^*.

Notice that $\mathsf{BASES}_h(S)$ is the optimum for the MIN ASSIGNMENT WITH BASES problem on input h and S (see the definition in Section 1.2). The main contribution of this section can be stated as follows.

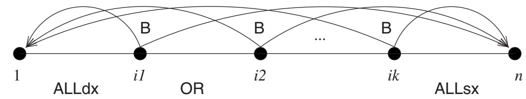

Fig. 2. The structure of an optimal solution for the base location problem.

THEOREM 3. *For any set S of n stations on the line and for any $1 \leq h \leq n-1$, it is possible to construct an optimal h-assignment of type* B^* *for S in $O(hn^3)$ time. Thus, the* MIN ASSIGNMENT WITH BASES *problem is in* P.

We provide here some properties of the structure of an optimal solution of the above problem. Indeed, consider the indices $i_1^*, \ldots, i_k^*$ of the k bases in any optimal solution (see Figure 2). Notice that, between two consecutive bases, any station must reach one of the two bases in at most $h-1$ hops (this is a necessary and sufficient condition, since reaching a base outside that interval would be more expensive). Additionally, the stations in $[1, i_1^*)$ (respectively, $(i_k^*, n]$) must reach in $h-1$ hops the base in i_1^* (respectively, i_k^*).

Thus, given the indices of the bases in an optimal solution, we can use the functions described in Section 3 to find the optimal assignment. Notice that if k would always be bounded by a constant, then we could try all the possible indices for the bases. However, this is not the case, so a more tricky approach is needed. Basically, we use the fact that every base "cuts" the instance into two *independent* intervals.

We define $\mathsf{BASES}_h(S, i)$ as the cost of a minimum h-assignment of type B^* with the further constraint that the rightmost base must have index i. Clearly,

$$\mathsf{BASES}_h(S) = \min_{1 \leq i \leq n} \mathsf{BASES}_h(S, i).$$

We then see how to compute $\mathsf{BASES}_h(S, i)$ for any i. To this aim, we need the function $\mathsf{CHANGE}_h(i, j)$ which, roughly speaking, corresponds to the change of the cost of $\mathsf{BASES}_h(S, i)$ when we set a *new* base $j > i$ (see Figure 3).

It is easy to see that the new base j is useless for the stations in $[1, i]$. Indeed, if a station in $[1, i]$ reaches j in at most $h-1$ hops, then it also reaches i within the same number of hops, thus making j useless. Moreover, the cost of an optimal assignment for $\mathsf{BASES}_h(S, i)$ restricted to $(i, n]$ is $\overleftarrow{\mathsf{ALL}}_{h-1}(i, n)$. The "new cost" due to base j in the interval $(i, n]$ is given by

$$\mathsf{OR}_{h-1}(i, j) + \overleftarrow{\mathsf{ALL}}_{h-1}(j, n) + \max\{d(1, j), d(j, n)\}^2.$$

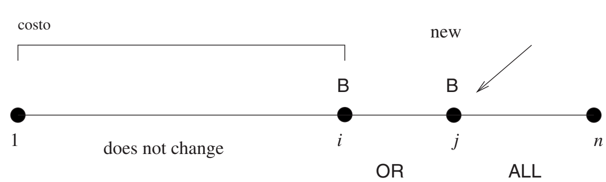

Fig. 3. The proof of Theorem 3.

We can thus define the following function.

DEFINITION 4 (Change). Given a set S of n stations and for any $1 \le i \le j \le n$, we define

$$\begin{aligned}&\mathsf{CHANGE}_h(i, j)\\&\quad = \mathsf{OR}_{h-1}(i, j) + \overleftarrow{\mathsf{ALL}}_{h-1}(j, n) + \max\{d(1, j), d(j, n)\}^2 - \overleftarrow{\mathsf{ALL}}_{h-1}(i, n).\end{aligned}$$

In order to compute $\mathsf{BASES}_h(S, \cdot)$, we first consider the basic case in which we impose the use of only one base station.

LEMMA 4. *Let* $\mathsf{BASE}^1_h(S, i)$ *be the minimum cost of the range assignment in which there exists only one base and its index is* i. *Then it holds that*

$$\mathsf{BASE}^1_h(S, i) = \overrightarrow{\mathsf{ALL}}_{h-1}(1, i) + \max\{d(1, i), d(i, n)\}^2 + \overleftarrow{\mathsf{ALL}}_{h-1}(i, n). \tag{7}$$

PROOF. Since only one base in position i occurs, by definition of BASE^1_h, any non-base station must reach such a base within $h-1$ hops. Also notice that a station l that is on the left of B and that reaches within $h-1$ hops another station r on the right of B, will also cross B within $h-1$ hops. Hence, l does not exploit the assignment of those stations on the right of B. So, the two intervals (on the left and on the right of B) are independent. This implies the lemma. □

PROOF OF THEOREM 3. Theorem 2, Definition 4, and Lemma 4 easily imply that the functions

$$\overrightarrow{\mathsf{ALL}}_{h-1}(i, j), \quad \overleftarrow{\mathsf{ALL}}_{h-1}(i, j), \quad \mathsf{OR}_{h-1}(i, j), \quad \mathsf{CHANGE}_h(i, j),$$
$$\text{and} \quad \mathsf{BASE}^1_h(S, i)$$

(for all $1 \le i < j \le n$) can be computed in $O(hn^3)$ time. Then the computation of $\mathsf{BASES}_h(S, j)$ works by induction over n phases. Phase $j = 1$ corresponds to computing

$$\mathsf{BASES}_h(S, 1) = \mathsf{BASE}^1_h(S, 1). \tag{8}$$

We now assume that, at the end of Phase $j-1$, $\mathsf{BASES}_h(S, i)$ have been computed for any $1 \le i \le j-1$. Then the computation of $\mathsf{BASES}_h(S, j)$ can be carried out according to the following recursive equation:

$$\mathsf{BASES}_h(S, j) = \min_{1 \le i < j}\{\mathsf{BASE}^1_h(S, j),\ \mathsf{BASES}_h(S, i) + \mathsf{CHANGE}_h(i, j)\}.$$

The correctness of the above equation follows from the fact that an optimal assignment for $\mathsf{BASES}_h(S, j)$ either contains only one base and its index is j, or it contains at least one base other than j. In the latter case the assignment in $[1, i]$ is not exploited by any station outside (here we use the same argument used in the proof of Lemma 4). Thus, such an assignment must be optimal for $\mathsf{BASES}_h(S, i)$.

Finally, each of the n phases takes $O(n)$ time. Hence, the overall time complexity is dominated by the computation of the functions in Theorem 2. This proves the theorem. □

5. Approximation Algorithms. In this section we present two polynomial-time approximation algorithms for the MIN ASSIGNMENT problem. In particular, we first describe a general 2-approximation algorithm working for any instance and then we show that the problem restricted to "well spaced" instances admits a polynomial-time approximation scheme.

5.1. *A 2-Approximation Algorithm for* MIN ASSIGNMENT. We consider a feasible assignment r for the instance $(h, S = \{s_1, \ldots, s_n\})$ of MIN ASSIGNMENT; by definition, it should be clear that r is a feasible assignment also for $\overleftarrow{\mathsf{ALL}}_h(1, n)$ and $\overrightarrow{\mathsf{ALL}}_h(1, n)$. This fact implies the following useful lower bound on the optimum of MIN ASSIGNMENT.

LEMMA 5. *For any set of stations $S = \{s_1, \ldots, s_n\}$ and for any $h \geq 1$, it holds that*

$$\mathsf{OPT}_h(S) \geq \max\{\overrightarrow{\mathsf{ALL}}_h(1, n), \overleftarrow{\mathsf{ALL}}_h(1, n)\}.$$

By combining the above lemma with the algorithm of Lemma 2, we get the following result.

THEOREM 4. *There is an algorithm that, for any set $S = \{s_1, \ldots, s_n\}$ and any $h \geq 1$, computes in $O(hn^3)$ time an h-assignment r_{apx} for S such that*

$$\mathsf{cost}(r_{\text{apx}}) \leq 2 \cdot \mathsf{OPT}_h(S).$$

PROOF. Let r_{sx} and r_{dx} be two optimal range assignments for $\overleftarrow{\mathsf{ALL}}_h(1, n)$ and $\overrightarrow{\mathsf{ALL}}_h(1, n)$, respectively. We define

$$r_{\text{apx}}(s_i) = \max\{r_{sx}(s_i), r_{dx}(s_i)\}, \qquad \text{for any} \quad s_i \in S.$$

It is easy to see that r_{apx} is a feasible h-assignment for S since s_i is able to reach both s_1 and s_n (and thus any other station in S) in at most h hops. Moreover,

$$\mathsf{cost}(r_{\text{apx}}) \leq \overleftarrow{\mathsf{ALL}}_h(1, n) + \overrightarrow{\mathsf{ALL}}_h(1, n).$$

Thus, Lemma 5 implies $\mathsf{cost}(r_{\text{apx}}) \leq 2 \cdot \mathsf{OPT}_h(S)$. □

5.2. *A Polynomial-Time Approximation Scheme for Well Spaced Instances.* This section is devoted to the construction of a better approximation for *well spaced instances* of the MIN ASSIGNMENT problem.

DEFINITION 5 (Well Spaced Instances). A family $\mathcal{S}$ of *well spaced* instances is a family of instances S such that

$$\delta_{\max}(S) = O(\delta_{\min}(S)\text{poly}(\log n)).$$

The following lemma extends some of the results of Theorem 1 to families of well spaced instances.

LEMMA 6.

(a) *For any set S of n stations on the line and for any constant $h \geq 1$, it holds that*

$$\mathsf{OPT}_h(S) = \Omega(\delta_{\min}(S)^2 n^{(2^{h+1}-1)/(2^h-1)}).$$

(b) *Let $\mathcal{S}$ be a family of well spaced instances. Then, for any instance $S^w \in \mathcal{S}$ of n stations and for any $h \geq 1$,*

$$\mathsf{OPT}_h(S^w) = \Omega\left(D(S^w)^2 \frac{n^{1/(2^h-1)}}{\mathrm{poly}(\log n)}\right).$$

PROOF. (a) Without loss of generality, assume that $\delta_{\min}(S) = 1$. Consider any feasible assignment r for S. It suffices to observe that this assignment is also feasible for the instance N of n stations at unit distance. Hence, the lower bound of Theorem 1 applies.

(b) From item (a) we obtain

$$\mathsf{OPT}_h(S^w) = \Omega(\delta(S^w)^2 n^{(2^{h+1}-1)/(2^h-1)}). \tag{9}$$

Since S^w is well-spread it holds that

$$\delta(S^w) \geq \frac{\delta_{\max}(S^w)}{\mathrm{poly}(\log n)}.$$

By replacing the above bound into (9), we get

$$\mathsf{OPT}_h(S^w) = \Omega\left(\frac{\delta_{\max}(S^w)^2}{\mathrm{poly}(\log n)} n^{(2^{h+1}-1)/(2^h-1)}\right) = \Omega\left(\frac{D(S^w)^2 n^{1/(2^h-1)}}{\mathrm{poly}(\log n)}\right),$$

where the last equality is a consequence of $\delta_{\max}(S^w)n \geq D(S^w)$. □

THEOREM 5. *There is an algorithm that, for any family $\mathcal{S}$ of well spaced instances, for any instance $S^w \in \mathcal{S}$, and for any constant $h > 0$, computes in $O(hn^3)$ time an h-assignment r^{apx} such that*

$$\mathsf{cost}(r_{\mathrm{apx}}) = (1 + \varepsilon(n)) \cdot \mathsf{OPT}_h(S^w),$$

where $\varepsilon(n) = o(1)$.

PROOF. We will prove that if $S^w \in \mathcal{S}$, then the optimum $\mathsf{BASES}_h(S^w)$ of the MIN ASSIGNMENT WITH BASES problem on instance (h, S^w) is such that

$$\mathsf{BASES}_h(S^w) = (1 + o(1)) \cdot \mathsf{OPT}_h(S^w).$$

Let r^{apx} be the range assignment yielded by the algorithm of Theorem 3 in $O(hn^3)$ time. Without loss of generality, assume that $h \geq 2$. If there is an optimal h-assignment for S^w of type B^* (see Definition 3), then r^{apx} is optimal for S^w. So, assume that all the optimal h-assignments for S^w are not of type B^*. Consider an optimal h-assignment r^{opt}. Let L

be the rightmost station reaching station 1 in one hop. It is not hard to see that all the stations that do not reach a base within $h-1$ hops, must reach L. We consider the range assignment r^B in which all the stations different from L keep the range as in r^{opt} while L now has a range sufficient to be a basis.

In this assignment any station reaches some base in at most $h-1$ hops, so r^B is an h-assignment of type B^*. It turns out that

$$\mathsf{cost}(r^{\mathrm{apx}}) \leq \mathsf{cost}(r^B) \leq \mathsf{OPT}_h(S^w) + D(S^w)^2.$$

Since $\mathcal{S}$ is well spaced, from Lemma 6(b) we have that, for some constant $c > 0$,

$$\mathsf{OPT}_h(S^w) \geq c D(S^w)^2 \frac{n^{1/(2^h-1)}}{\mathrm{poly}(\log n)}.$$

By combining the last two inequalities, we get

$$\frac{\mathsf{cost}(r^{\mathrm{apx}})}{\mathsf{OPT}_h(S^w)} \leq 1 + \frac{D(S^w)^2}{\mathsf{OPT}_h(S^w)} \leq 1 + c\frac{\mathrm{poly}(\log n)}{n^{1/(2^h-1)}}.$$

For constant h and for $n \to \infty$, the above value tends to 1. Hence the theorem follows. □

Notice that the approximation ratio of Theorem 5 tends to 1 as n goes to infinity *independently* of the cost of the optimum. Hence, the following result holds.

COROLLARY 2. *For any family $\mathcal{S}$ of well-spread instances, for any constant $h > 0$, and for any constant $\varepsilon > 0$, there exists a $(1+\varepsilon)$-approximation algorithm for* MIN ASSIGNMENT *restricted to $\mathcal{S}$ that works in $O(hn^3)$ time.*

PROOF. Consider the algorithm used to prove Theorem 5. For any family $\mathcal{S}$ and for any $\varepsilon > 0$, a constant $\bar{n}$ exists such that, on instances with at least $\bar{n}$ stations, this algorithm has an approximation ratio at most $1+\varepsilon$ (this follows from Theorem 5). On the other hand, for any instance in $\mathcal{S}$ having less than $\bar{n}$ stations, we can compute the optimum in $O(\bar{n}^{\bar{n}}) = O(1)$ time. Hence the corollary follows. □

6. A Polynomial-Time Exact Algorithm for Two Hops. A suitable application of the algorithms given in Lemma 2 and Theorem 3 will allow us to obtain an algorithm that solves MIN 2-HOPS ASSIGNMENT in $O(n^3)$ time. With this aim, we need to distinguish three types of possible optimal 2-assignments.

DEFINITION 6. Let r be a 2-assignment and let L (R) denotes the rightmost (leftmost) non-base station reaching 1 (n) in one hop. We say that r is of type

- B^* if all the stations reach one base in one hop (this is the same as in Definition 3);
- LR if there are no bases and L is on the left of R;
- $\mathsf{B}^*\mathsf{LRB}^*$ if there is at least a base station and there are no bases between L and R and L is on the left of R.

Informally speaking, the next lemma states that in order to compute a solution with optimal cost, we only need to compute the optimal solutions among those whose type is listed in Definition 6.

LEMMA 7. *Let S be a set of stations on the line. If r is a 2-assignment for S, then it is either of type* B^*, LR, *or* $\mathsf{B}^*\mathsf{LRB}^*$.

PROOF. Without loss of generality, assume that r is not of type B^* nor LR. By contradiction, if r is not of type $\mathsf{B}^*\mathsf{LRB}^*$, then one of the following two conditions must hold:

1. In r there is an L but no R (or vice versa). In this case any non-base station s that reaches L in one hop must also reach a B (s cannot directly reach the rightmost station since in this case there would be an R). Hence r is of type B^*.
2. In r there is a B between L and R. Again, consider a station s that reaches both L and R (otherwise it has to reach a base). Such s will also reach the B in between L and R, thus implying that r is of type B^*.

In both cases we have a contradiction, thus implying the lemma. □

Our next goal is to provide efficient algorithms to compute the optimal 2-assignment for each of the three types listed in Definition 6.

LEMMA 8. *For any set S of stations on the line, the optimal assignment of type* B^* *can be computed in $O(n^3)$ time.*

PROOF. The proof easily follows from Theorem 3. □

LEMMA 9. *For any set S of stations on the line, the optimal assignment of type* LR *can be computed in $O(n^3)$ time.*

PROOF. For each possible pair L and R, we determine the optimal assignment by using the following property. Since no station can be a base, a necessary (and sufficient) condition for every station s, to reach both 1 and n in two hops, is that s reaches in one hop both stations L and R. For instance, if $s \in [1, \mathsf{L}]$, then by definition of R, $s \to \mathsf{R}$. This implies that s will also reach L in one hop. By using similar arguments, we can show that the same condition holds for the other two cases: $s \in (\mathsf{L}, \mathsf{R})$ and $s \in [\mathsf{R}, n]$, respectively. □

LEMMA 10. *For any set S of stations on the line, the optimal assignment of type* $\mathsf{B}^*\mathsf{LRB}^*$ *can be computed in $O(n^3)$ time.*

PROOF. We first describe a simple algorithm that works in $O(n^5)$ time.

We consider the configuration in Figure 4 where the rightmost B before L (respectively, the leftmost B after R) has index i (respectively, l). As for the interval $[1, i)$, we compute the optimal 2-assignment of type B^* by considering the instance obtained by removing

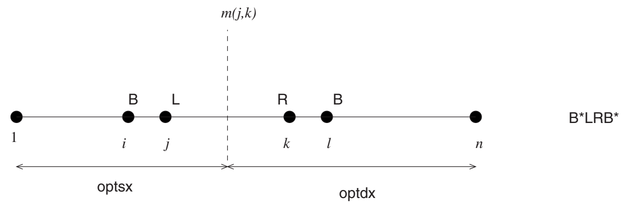

Fig. 4. The proof of Lemma 10.

all stations in (i, n). To this aim, we apply Theorem 3. Similarly, we compute the 2-assignment for $(l, n]$. Any $s \in (i, l)$, must reach one of the two bases or both L and R, i.e.,

$$r(s) = \min\{d(s, i), d(s, l), \max\{d(s, j), d(s, k)\}\}.$$

Clearly, the overall task requires $O(n^5)$ time.

In order to speed up the above algorithm, we make a suitable decomposition of the interval (i, l) into independent subintervals. To this aim, let $m(j, k)$ be the rightmost station $s \in [j, k]$ such that $d(s, j) \leq d(s, k)$. Observe that, given the configuration in Figure 4, any optimal 2-assignment in $[1, m(j, k)]$ must satisfy

1. $\forall s \in [1, i]$: s is a base or it reaches a base in $[1, i]$ in one hop;
2. $\forall s \in (i, m(j, k)]$: s reaches in one hop the base of index i or R.

Also notice that, because of the definition of $m(j, k)$, the above conditions are also sufficient. We therefore define $\mathsf{OPT}^{sx}(i, j, k)$ as the minimum cost among all the assignments r for the stations in $[1, m(j, k)]$ satisfying the following conditions:

(a) $\forall s \in [1, m(j, k)], \quad s \rightarrow_{2,r} 1 \wedge s \rightarrow_{2,r} n$ (i.e., r is a 2-assignment for $[1, m(j, k)]$);
(b) $\forall s \in (i, m(j, k)], \quad s \not\rightarrow_{r} n$ (by definition of R);
(c) $\forall s \in (j, m(j, k)], \quad s \not\rightarrow_{r} 1$ (by definition of L).

By a symmetrical argument, we define $\mathsf{OPT}^{dx}(j, k, l)$ with respect to the interval of stations $(m(j, k), n]$ (see Figure 5). Also let

$$\mathsf{OPT}^{sx}(j, k) = \min_{1 \leq i < j} \mathsf{OPT}^{sx}(i, j, k) \quad \text{and} \quad \mathsf{OPT}^{dx}(j, k) = \min_{k < l \leq 1} \mathsf{OPT}^{sx}(j, k, l).$$

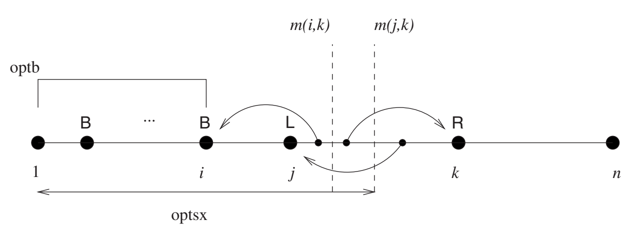

Fig. 5. The proof of Lemma 10.

It should then be clear that

$$\mathsf{OPT}_2(i,n) = \min_{1\le j<k\le n}\{\mathsf{OPT}^{sx}(j,k) + \mathsf{OPT}^{dx}(j,k)\}.$$

In what follows, we show how to compute the functions $\mathsf{OPT}^{sx}(i, j, k)$ and OPT^{dx} (j, k, l) efficiently. From Condition 1 above (see also Figure 4), the contribution of $[1, i]$ in $\mathsf{OPT}^{sx}(i, j, k)$ is equal to

$$\mathsf{BASES}_2([1,n], i) - \overleftarrow{\mathsf{ALL}}(i,n). \tag{10}$$

We define $m(i, k)$ to be the rightmost station $s \in (i, k]$ such that $d(s, i) \le d(s, k)$. Notice that Condition 2 above implies that if $s \in (i, m(i, k)]$, then $s \to i$, else (i.e., $s \in [m(i, k) + 1, m(j, k)]$) $s \to k$. Hence, the contribution of $(i, m(j, k)]$ is

$$\overleftarrow{\mathsf{ALL}}(i, m(i,k)) + \overleftarrow{\mathsf{ALL}}(m(i,k)+1, k) - \overrightarrow{\mathsf{ALL}}(m(j,k)+1, k). \tag{11}$$

Finally, by combining (10) and (11), we obtain

$$\begin{aligned}\mathsf{OPT}^{sx}(i,j,k) \;=\; & \mathsf{BASES}_2([1,n], i) - \overleftarrow{\mathsf{ALL}}(i,n) + \overleftarrow{\mathsf{ALL}}(i, m(i,k)) \\ & + \overrightarrow{\mathsf{ALL}}(m(i,k)+1, k) - \overrightarrow{\mathsf{ALL}}(m(j,k)+1, k).\end{aligned}$$

From Theorem 2 and from the proof of Theorem 3, all such functions can be computed, for all posible arguments, in $O(n^3)$ time in a preliminary phase once and for all. So, for every fixed triple i, j, and k, $\mathsf{OPT}^{sx}(i, j, k)$ can be computed in constant time. A symmetric argument applies to OPT^{dx}. □

We are now in a position to prove the main result of this section.

THEOREM 6. *The* MIN 2-HOPS ASSIGNMENT *problem can be solved in* $O(n^3)$ *time*.

PROOF. Lemma 7 implies that an optimal cost 2-assignment can be obtained by computing the optimal assignments of type LR, B^*, and $\mathsf{B}^*\mathsf{LRB}^*$ and then taking the one having minimal cost. Lemmas 9 and 10 imply that the above task can be performed in $O(n^3)$ time. □

7. Open Problems. In 1997 Kirousis et al. [7] wondered whether efficient solutions for the bounded hops case of the MIN ASSIGNMENT problem can be obtained. This paper provides a first positive answer to this question since, in most practical applications, an easy-to-obtain 2-approximation can be reasonably considered good enough. On the other hand, the main question left open is whether MIN ASSIGNMENT ∈ P. However, we believe that exact solution can be obtained only by algorithms whose time-complexity is $\Omega(n^h)$ (unless P = NP). We thus leave as another important open question whether an algorithm exists for MIN ASSIGNMENT having time-complexity $n^{O(h)}$. Finally, it is interesting to observe that a similar "state of the art" holds for another important "Euclidean" problem, i.e., the *Geometric Facility Location Problem* (in short, GFLP) [6]. Indeed, also in this case the Euclidean properties of the support space have been used to derive very good approximation results [1] (much better than those achievable on non-geometric versions of FLP). On the other hand, it is still unknown whether or not GFLP $\in$ P.

References

[1] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for the euclidean *k*-medians and related problems. In *Proc*. 30*th Annual ACM Symposium on Theory of Computing* (*STOC*), pages 106–113, 1998.

[2] M.A. Bassiouni and C. Fang. Dynamic channel allocation for linear macrocellular topology. *Proc*. *ACM Symposium on Applied Computing* (*SAC*), pages 382–388, 1998.

[3] A. Clementi, P. Penna, and R. Silvestri. Hardness results for the power range assignment problem in packet radio networks. *Proc*. *RANDOM-APPROX* '99, *Randomization*, *Approximation and Combinatorial Optimization*, pages 197–208. LNCS 1671. Springer-Verlag, Berlin, 1999.

[4] A. Clementi, P. Penna, and R. Silvestri. The power range assignment problem in radio networks on the plane. *Proc*. 17*th Annual Symposium on Theoretical Aspects of Computer Science* (*STACS*), pages 651–660. LNCS 1770. Springer-Verlag, Berlin, 2000.

[5] K. Diks, E. Kranakis, D. Krizanc, and A. Pelc. The impact of knowledge on broadcasting time in radio networks. *Proc*. 7*th Annual European Symposium on Algorithms* (*ESA*), pages 41–52. LNCS 1643. Springer-Verlag, Berlin, 1999.

[6] D. Hochbaum. Heuristics for the fixed cost median problem. *Math*. *Programming*, 22:148–162, 1982.

[7] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoret*. *Comput*. *Sci*., 243:289–306, 2000. (An extended abstract appeared also in *Proc*. 14*th Annual STACS* '97).

[8] E. Kranakis, D. Krizanc, and A. Pelc. Fault-tolerant broadcasting in radio networks. *Proc*. 6*th Annual European Symposium on Algorithms* (*ESA*), pages 283–294, LNCS 1461. Springer-Verlag, Berlin, 1998.

[9] R. Mathar and J. Mattfeldt. Optimal transmission ranges for mobile communication in linear multihop packet radio netwoks. *Wireless Networks*, 2:329–342, 1996.

[10] K. Pahlavan and A. Levesque. *Wireless Information Networks*. Wiley-Interscience, New York, 1995.

[11] P. Piret. On the connectivity of radio networks. *IEEE Trans*. *Inform*. *Theory*, 37:1490–1492, 1991.

[12] S. Ulukus and R.D. Yates. Stochastic power control for cellular radio systems. *IEEE Trans*. *Comm*., 46(6):784–798, 1998.