Experimental Analysis of Practically Efficient Algorithms for Bounded-Hop Accumulation in Ad-Hoc Wireless Networks

Andrea E. F. Clementi* Miriam Di Ianni* Angelo Monti†

Gianluca Rossi*1 Riccardo Silvestri†

Abstract

The paper studies the problem of computing a minimal energy-cost range assignment in an ad-hoc wireless network which allows a set S of stations located in the 2-dimensional Euclidean space to perform accumulation (all-to-one) operations towards some root station b in at most h hops (2-Dim Min h-Accumulation Range Assignment problem).

We experimentally investigate the behavior of *fast* and *easy-to-implement* heuristics for the 2-Dim Min h-Accumulation Range Assignment problem on instances obtained by choosing at random n points in a square of side length L. We compare the performance of an easy-to-implement, very fast heuristic with those of three simple heuristics based on classical greedy algorithms (Prim's and Kruskal's ones) defined for the Minimum Spanning Tree problem. The comparison is carried out over thousands of random instances in several different situations depending on: the distribution of the stations in the plane, their density, the energy cost function.

1 Introduction

An *ad-hoc* wireless network consists of a set of radio stations connected by wireless links. In an ad hoc network, to every station is assigned a transmission range. The overall range assignment determines a transmission (directed) graph since one station s can transmit to another station t if and only if t is within the transmission range of s. The transmission range of a station depends, in turn, on the energy power supplied to the station. In particular, the power P_s required by a station s to correctly transmit data to another station t must satisfy the inequality

$$\frac{P_s}{\mathtt{dist}(s,t)^\alpha} > \gamma \qquad (1)$$

where $\mathtt{dist}(s,t)$ is the Euclidean distance between s and t, $\alpha \geq 1$ is the *distance-power gradient*, and $\gamma \geq 1$ is the *transmission quality* parameter. In an ideal environment (i.e., in empty space) it holds that $\alpha = 2$ but it may vary from 1 to more than 6 depending on the environment conditions at the location of the network (see [17]).

The fundamental problem underlying any phase of a dynamic resource allocation algorithm in ad-hoc wireless networks is the following. Find a transmission range assignment such that (1) the corresponding transmission graph satisfies a given connectivity property Π, and (2) the overall energy power required to deploy the assignment (according to Inequality (1)) is minimized (see for example [21, 13]). In [7], the reader may find an exhaustive survey on the previous results related to the above problem.

For certain wireless networks, such as ad hoc and sensor networks, a fundamental question is whether it is advantageous to route over many short hops (short-hop routing or, in the extreme case, nearest-neighbor routing) or over a smaller number of longer hops (long-hop routing). Short-hop routing gained a lot of support, mainly due to energy consumption considerations: in fact, if the distance-power gradient is grater than one, a long hop of length d is much more expensive than, say, h hops of length d/h. However, this argument stems from an oversimplified analysis of the energy consumption and neglects important issues such as delay, end-to-end reliability, and bias power consumption. Recently, this issue has been more deeply considered and several arguments in favor of routing over a smaller number of hops have been brought (see [12] for a detailed analysis of this issue). Among these arguments, we list end-to-end delay, end-to-end reliability (each hop may increase fault probability).

Clearly, the most compelling reason against many-hops routing is the end-to-end delay. Of course, energy and delay can be traded off against each other.

*Dipartimento di Matematica, Università degli Studi di Roma "Tor Vergata". E-mail: {clementi,diianni,rossig}@mat.uniroma2.it

†Dipartimento di Informatica, Università degli Studi di Roma "La Sapienza". Email: {monti,silver}@di.uniroma1.it

3 Partially supported by the European Union under the IST FET Project CRESCCO

So, for a fair comparison, we will impose a delay constraint and determine which range assignment consumes the least amount of energy. In particular, this paper addresses the case in which Π is defined as follows: *Given a set of stations and a specific root station b, the transmission graph must contain a spanning tree directed towards b of depth at most h.* The relevance of this case is due to the fact that any transmission graph satisfying the above property allows the source station to *accumulate* data from all stations in at most h hops. This task constitutes one of the typical activities of real life multi-hop radio networks such as sensor networks [22, 21].

Previous results The accumulation range assignment problem (denoted as 2-Dim Min h-Accumulation Range Assignment(α)), described above, is a *special case* of the well known Minimum Spanning Tree problem on general graphs in which a constraint on the height of the spanning tree, from a given root, has to be satisfied (in short, it will be denoted as h-HMST). Indeed, it is the case in which nodes are points of the plane and the weight of edge (u, v) equals $d(u, v)^\alpha$, for any pair u, v of nodes.

As for the h-HMST problem, Gouveia [9] and, successively, Gouveia and Requejo [11] provided and experimentally tested exact super-polynomial-time algorithms, based on the branch and bound technique. In [3] a polynomial-time $O(\log n)$-approximation algorithm is given, but its time complexity is $n^{O(h)}$.

The 2-HMST problem can be easily reduced to the classic Uncapacitated Facility Location Problem (UFLP). It thus follows that all the approximation algorithms for the latter problem apply to the 2-HMST as well. As for the *metric* FLP, several polynomial-time approximation algorithms have been presented in the literature. The first constant factor approximation algorithm was given by Shmoys et al in [20], they presented a 3.16 approximation algorithm. After this, a series of constant factor approximation algorithms was published, see [5, 14, 10]. Currently, the best factor for metric FLP is 1.52 due to Mahdian et al [16]. All such algorithms make use of Linear Programming relaxations that do not yield practically efficient implementations. Notice that, in [2] Alfandari and Paschos proved that metric 2-HMST is Max SNP-hard and, hence, PTAS cannot be found for this problem unless P = NP.

As for the *Euclidean* case (notice that this corresponds to "our" 2-Dim Min 2-Accumulation Range Assignment(1)) problem, the best result is the PTAS given by Arora et al in [4]. The algorithm works also in higher dimensions; however, it is based on a very complex dynamic programming technique that makes any implementation very far to be practically efficient.

In [19], an *evolutionary*-based heuristic, for the 2-Dimensional Euclidean h-HMST problem, has been presented and experimentally compared to two greedy heuristics based on the classic Prim's algorithm for the MST problem. A more detailed description of these results is given in the next Subsection.

An asymptotical, probabilistic analysis of practically-efficient heuristics for the 2-Dimensional Euclidean h-HMST (corresponding to our 2-Dim Min h-Accumulation Range Assignment(1) problem) is provided in [6], for any *constant h*. The analysis adopts the *random uniform model*, i.e., a set of n points selected uniformly and independently at random from a square of size L. The major motivation for studying this input model is twofold: on one hand, uniform distribution is the best choice when *nothing* is known about the *real* input distribution; on the other hand, uniform random instances well model *well spread* instances which are typical in sensor networks. More precisely, they first prove that a divide-et-impera based heuristic (called Party) yields, *with high probability*[1], constant approximation factor. It is important to observe that this approximation factor depends on h. Then, they analyze three greedy heuristics: one heuristic (Kruskal) based on the well-known Kruskal's algorithm and two heuristics - a deterministic one (Prim) and a randomized one (Randomized Prim) based on the well-known Prim's algorithm. They prove that, with high probability, *all* such greedy heuristics do not asymptotically behave better than the trivial algorithm that directly connects each node to the root: Their cost is $\Theta(n \cdot L)$ while the optimal cost is $\Theta(L \cdot n^{\frac{1}{2}+\frac{1}{2^{h+1}-2}})$.

Informally speaking, such asymptotical analysis would lead us to *always* prefer the divide et impera heuristic to solve the 2-Dim Min h-Accumulation Range Assignment(1) problem.

Our results. The asymptotical analysis in [6] is unable to capture the role played by the constants. In fact, it states that, with high probability, *up to constant factors depending on the constant h, there exists a constant* n_0 such that, whenever the number of nodes is greater than n_0, the divide-et-impera heuristic (Party) behaves much better than any of the greedy heuristics. However, it does not provide any bound on both the constant factors and the constant n_0. It should be clear that such bounds are

[1] With the term *with high probability*, we mean that the event holds with probability at least $1 - e^{-cn}$ where n is the node number and c is a fixed positive constant.

crucial in any practical application: the cost of the solutions achieved by the greedy approach might be better than that achieved by the divide-et-impera one, for a large and important range of the input parameters (i.e. n and h). The specific aim of this work is to determine and compare the quality of the solutions returned by the above heuristics for *practically-important* ranges of the input parameters. This is accomplished by performing a wide experimental analysis of the above heuristics on random instances.

As already cited, experimental evaluation of some heuristics for 2-Dim Min h-Accumulation Range Assignment(1) has been performed in [19]. In particular, the authors have experimentally compared a heuristic based on the Evolutionary technique to the same two greedy heuristics based on the Prim's algorithm for the MST problem analyzed in [6] and in this paper. The experiments show that Randomized Prim significantly outperforms Prim and that the Evolutionary heuristic behaves slightly better than the Randomized Prim. However, the running time of the Evolutionary heuristics is much worse than the running time of both of the Prim-based heuristics. In fact, it heavily limits the range of the experiments sizes which have been carried out only on "small" instances, i.e., with a small number of nodes (from 50 to 1000).

While the results in [6, 19] *only hold* for the 2-Dim Min h-Accumulation Range Assignment(1) problem, we want to consider here the 2-Dim Min h-Accumulation Range Assignment(2) problem too, due to its more relevance for wireless networks. To this aim, we apply the same four heuristics considered in [6] for the 2-Dim Min h-Accumulation Range Assignment(1) problem to the 2-Dim Min h-Accumulation Range Assignment(2) one. Concerning the latter, the lower bound on the cost of any h-hops accumulation range assignment has been proved in [8], as well as the asymptotical optimality of the divide-et-impera heuristic. Moreover, The *asymptotical bad* behavior of the greedy heuristics easily follows from the proof techniques presented in [6].

We widely test the four heuristics presented in [6] for the 2-Dim Min h-Accumulation Range Assignment(α) problem on *random* instances, for some fixed $h > 0$ and $\alpha = 1, 2$. In particular, we consider two different distributions: the above mentioned *uniform* distribution and the distribution deriving from a Poisson point process in the plane. Our tests have been carried out on random instances with size that varies from 100 up to $80,000$ points. The parameter h is set to 3, 5 and 8. In what follows, we summarize the results coming from our tests and we remind to the next Sections for a further discussion of the obtained results.

- For $h = 3$, the experiments show that the Party heuristic is *always* better than the other ones (see Figures 2.a and 3.a).

- For $h = 5$, the Party heuristic behaves better than the other heuristics for instance of at least $1500 - 2000$ points (see Figure 2.b and 3.b).

- Finally, for $h = 8$ the Randomized Prim heuristic is the hardest to overwhelm in fact it is the best heuristic for networks of size up to some dozen of thousand of points. Moreover, we notice a greater dependence on the parameter α: indeed, when $\alpha = 1$ the Party heuristic is better than Randomized Prim starting at $16,000$ points but when $\alpha = 2$ the behavior of Randomized Prim is "always" (for a number of points less than $30,000 - 35,000$!) much better than α-Party for both the distributions (see Figures 2.c and 3.c).

By summarizing, we can state that the "constants" hidden by the asymptotical analysis provided in [6] *strongly depend on* h: the asymptotical optimality of the Party heuristic comes out only when h is very small w.r.t. n, while Randomized Prim yield solutions of better cost for larger values of h. The latter fact leads us to conjecture that, for non constant values of h, the upper bound for the solution cost produced by Randomized Prim has the following form:

$$O\left(\frac{1}{f(h)} L n^{\alpha}\right)$$

where $f(h)$ is some "rapidly" increasing function of h.

On the other hand, it is important to emphasize that the Party heuristic is *extremely* fast (it works in $O(n)$ time) while the greedy heuristics works in $O(n^2)$.

Paper organization In Section 2, we describe the four heuristics and we briefly describe the asymptotical results for the 2-Dim Min h-Accumulation Range Assignment(α) problem for $\alpha = 1$ and show how to extend them to $\alpha = 2$. In section 3 we describe the experiments we have carried out and discuss their output. Finally, in Section 4 we draw some conclusions and state some open problems.

2 Practically Efficient Heuristics and their analysis

We first state the lower bounds from [6, 8] on the cost of any solution for the 2-Dim Min h-Accumulation Range Assignment(α) problem, for $\alpha = 1$ and $\alpha = 2$.

Theorem 2.1 *Let $h \geq 1$ be a constant value. Let S be a random set of n points in a square of side length L and let T be any h-tree spanning S. Then, the following results hold with high probability*[2]*:*

- *for* $\alpha = 1$, $\mathsf{cost}(T) = \Omega\left(L \cdot n^{\frac{1}{2}+\frac{1}{2^{h+1}-2}}\right)$ *[6]*
- *for* $\alpha = 2$, $\mathsf{cost}(T) = \Omega\left(L^2 \cdot n^{\frac{1}{h}}\right)$ *[8]*

We now introduce a simple *Divide et Impera* heuristic, α-PARTY. It takes in input the value h, a set of points V and a root $p \in V$ and makes a partition of the smallest square containing V into *cells* of suitable size; in each non-empty cell, it selects an *arbitrary* sub-root a and connects a to the root b; finally, it solves the non-empty cell sub-instances of the problem with $h-1$ hops, recursively. In both cases of $\alpha = 1$ and $\alpha = 2$, the cell size is a function of h, that is, it is equal to $\lfloor |V|^{\eta_\alpha(h)} \rfloor$, where $\eta_1(h) = \frac{1}{2} + \frac{1}{2^{h+1}-2}$ and $\eta_2(h) = \frac{1}{h}$.

```
procedure α-PARTY(h, V, p)
    if h = 1 then T ← {{x, p}|x ∈ V − {p}};
    else begin
        k ← ⌊|V|^{η_α(h)}⌋;        T ← ∅;
        Let l be the side length of the smallest square
              containing all points in V;
        Partition the square into a grid of square cells
              of side length l/⌊√k⌋;
        Let k′ be the number of cells and let V_i be the
              points of V in the i-th cell, 1 ≤ i ≤ k′;
        for i ← 1 to k′ do
            if |V_i| ≥ 1 then begin
                a ← a random point in V_i;
                T ← T ∪ {{a, p}};
                if |V_i| > 1 then
                    T ← T∪ α-PARTY(h − 1, V_i, a);
            end;
    end;
    output T
```

Of course, a solution for the original instance of 2-DIM MIN h-ACCUMULATION RANGE ASSIGNMENT(α) is given by α-PARTY(h, S, b). The bound on the solution cost yielded by α-PARTY is stated in Theorem 2.2, from [6] and [8]. Moreover, it is immediate to verify that, for any $h > 0$, the worst-case time complexity of α-PARTY is $O(n)$.

Theorem 2.2 *Let $h \geq 1$ be a constant value. Let S be a random set of n points in a square of side length L and let $b \in S$. For any h-tree T returned by α-PARTY on input (h, S, b), with high probability it holds that*

- *for* $\alpha = 1$, $\mathsf{cost}(T) = \mathrm{O}\left(L \cdot n^{\frac{1}{2}+\frac{1}{2^{h+1}-2}}\right)$ *[6]*
- *for* $\alpha = 2$, $\mathsf{cost}(T) = \mathrm{O}\left(L^2 \cdot n^{\frac{1}{h}}\right)$ *[8]*

Theorems 2.1 and 2.2 imply that, for any fixed h, α-PARTY returns a solution which is, with high probability, a constant factor approximation of the optimum.

The problem of finding an MST can be efficiently solved by classical greedy algorithms like Prim's [18] and Kruskal's [15] ones. By adapting them for the hop constraint, it is possible to derive fast heuristics for the 2-DIM MIN h-ACCUMULATION RANGE ASSIGNMENT(α) problem. We analyze three heuristics. The first two were originally formulated for the slightly different version of the problem in which the hop constraint is on the tree diameter. The first Prim-based heuristic was introduced by Abdalla et al. [1]: it starts from the root point and at each step it chooses the minimum weight edge connecting a new point and satisfying the hop constraint. We call this heuristics PRIM. Another Prim-based heuristic, denoted here as RANDOMIZED PRIM, was presented in [19]: at each step it picks at random a point, not yet connected, and it connects the point to the tree by the minimum weight edge satisfying the hop constraint.

```
procedure PRIM(h, S, b)
    U ← S − {b};    T ← ∅;
    while U ≠ ∅ do begin
        Let {a, c} be the pair of points of minimum
              distance with a ∈ S − U and c ∈ U and
              such that T ∪ {{a, c}} is an h-tree;
        T ← T ∪ {{a, c}};        U ← U − {c};
    end;
    output T
```

```
procedure RANDOMIZED PRIM(h, S, b)
    T ← ∅;
    U ← S − {b};
    while U ≠ ∅ do begin
        c ← a random point from U;
        a ← the point in S − U nearest to c such that
              T ∪ {{a, c}} is an h-tree;
        T ← T ∪ {{a, c}};
        U ← U − {c};
    end;
    output T
```

We introduce a third heuristic KRUSKAL, inspired by Kruskal algorithm: initially each point forms a component; then, at each step, it chooses a minimum weight edge such that it merges two components and the resulting set of components can still be connected in a tree satisfying the hop constraint.

We recall that Kruskal's algorithm starts from the forest of n disjoint trees (i.e. *components*) and, at

[2] Here and in the sequel the term *with high probability* (in short, *w.h.p.*) means that the event holds with probability at least $1 - e^{-c \cdot n}$, for some constant $c > 0$.

every stage, it considers the edge e of minimal weight connecting two disjoint trees T_1 and T_2. Then, it replaces these two trees with the tree $T = T_1 \cup \{e\} \cup T_2$. Our heuristic needs the following notions. A component is *feasible* if either it contains the root and its height is at most h or it does not contain the root and its diameter is at most $2h - 2$. An edge is *feasible* if it connects two feasible components yielding a feasible component.

procedure KRUSKAL(h, S, b)
  $T \leftarrow \{C_1, C_2, \ldots, C_n\}$ where C_i is the component consisting of the ith point of S;
  while $|T| > 1$ **do begin**
    Let $\{p, q\}$ be the minimum weight feasible edge w.r.t. T;
    Let C and C' be the feasible components connected by $\{p, q\}$;
    Let C'' be the feasible component yielded by connecting C and C' by $\{p, q\}$;
    $T \leftarrow T \cup \{C''\} - \{C, C'\}$;
  end;
  output T

The time complexity of PRIM and of RANDOMIZED PRIM is $O(n^2)$. The time complexity of KRUSKAL is $O(n^2 \log n)$.
Next theorem shows that the asymptotic behavior of the above three heuristics is the same on random instances. The case $\alpha = 1$ follows by [6], while the case $\alpha = 2$ follows by a simple adaptation of the proofs in [6].

Theorem 2.3 *Let $h \geq 1$ be a constant value. Let S be a random set of n points on a square of side length L and let T be any h-tree spanning S returned by either* PRIM *or* RANDOMIZED PRIM *or* KRUSKAL. *Then, with high probability it holds that*

- *for $\alpha = 1$, $\mathsf{cost}(T) = \Omega(L \cdot n)$ [6]*
- *for $\alpha = 2$, $\mathsf{cost}(T) = \Omega(L^2 \cdot n)$*

The lower bound is clearly asymptotically tight, since the heuristics cannot do worse than that. Rather surprisingly, they thus do not do better, up to some constant factor, than the trivial algorithm that connects each point directly to the root.
The proof of Theorem 2.3 implicitly gives reasons for the bad behavior of each of the three heuristics. Given an h-tree T, a point is said to be a *bridge* if it is not a leaf and its father is at distance not less than a suitable constant fraction of L. Informally speaking, the bad behavior of the greedy heuristics can be explained in terms of the number and the distribution of bridges. As for PRIM, it is shown that there are no bridges, w.h.p.. Thus, the points, "far" from the root, must directly connect to some points "close" to the root: in a random instance, the number of such points is a large fraction of n (see Figures 1.b and 1.f). In the solutions yielded by RANDOMIZED PRIM, it is firstly proved that there are only a constant number of bridges, w.h.p.. Then, it is shown that the cost yielded by the points to reach such bridges is high (see Figures 1.c and 1.g). To sum up, the bad behavior of the Prim-based heuristics is due to a "too small" number of bridges. On the other hand, it is proved that KRUSKAL yields "too many" bridges, w.h.p. (see Figures 1.d and 1.h).

3 Experimental analysis

In this section, we show the experimental results of the studied heuristic and a modification of α-PARTY called α-PARTYEX. This heuristic differs from α-PARTY in that, after choosing the point a in V_i, instead of connecting a to p, it choose a connection point c in T whose level is at most the level of p and the distance from a is minimal. In particular in Section 3.1 we show the results of the experimentation of the proposed heuristics for $\alpha = 1$. The experiments described in Section 3.2 are executed in the "wireless" case $\alpha = 2$. For this case we perform experiments with the classical uniform distribution and also with the normal distribution. The latter distribution is useful to model the behavior of the heuristics in the case of networks implemented in urban conglomerates whose density is high in the center and decreases towards the periphery. The normal distribution is considered only for the case $\alpha = 2$ due to the major relevance of this case for wireless networks.

All the experiments are carried out for several dimensions n of the random instances (between 100 and 80,000) and for $h = 3, 5, 8$. For each n and h, the number of runs decreases from 1000 to 50 as n grows. Notice that, in each round associated to any fixed n and h, we run the heuristics on the same instance generated at random.

First of all, we observe that for $h = 3$ and $h = 5$ α-PARTY (and α-PARTYEX) is always better than the other heuristics independently from the chosen distribution and from the value of α. Hence, for small values of h the experimental behavior reflects the theoretical asymptotical results, even for small values of n. The influence of the value of α for the case $h = 8$ is stronger. A more detailed analysis of these fact will be given in the appropriate sections.

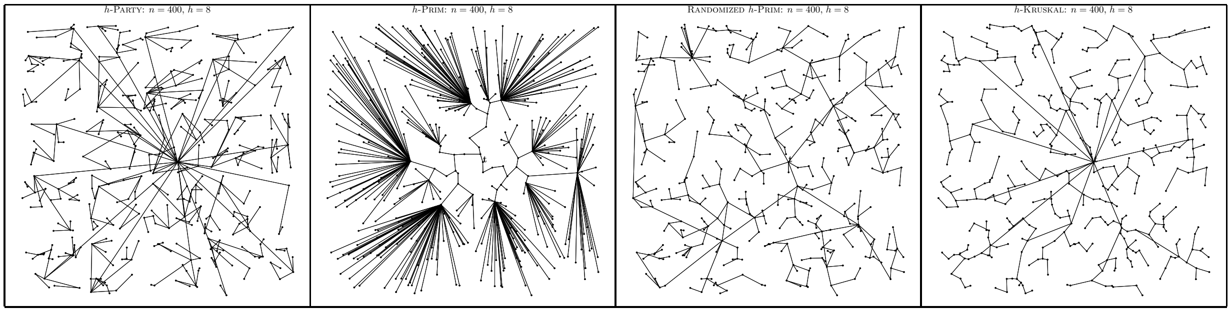

Figure 1: The trees yielded by the heuristics on the same random instance with 400 points and $h = 8$.

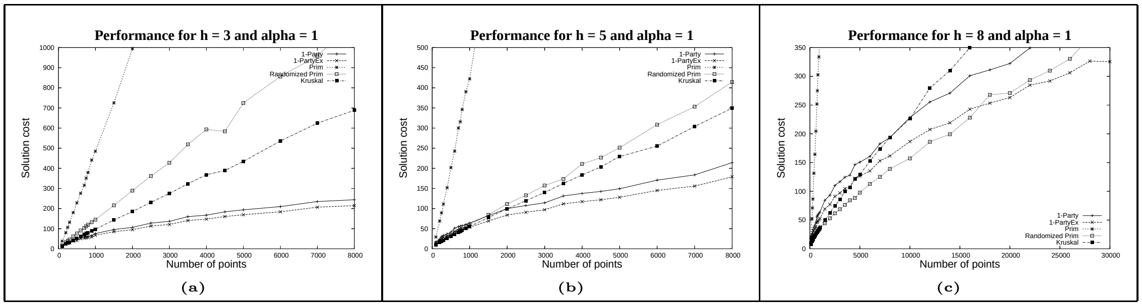

Figure 2: Performance of the heuristics for $\alpha = 1$, in the uniform distribution case.

3.1 The Euclidean Case ($\alpha = 1$)

In Figures 2.a-c we show the average costs of the heuristics for $\alpha = 1$. In particular, Figure 2.a and Figure 2.b show the results for $h = 3$ and $h = 5$, respectively. Here the dimensions n of the random instances varies between 100 and 8000. For these two values of h the performance of both 1-Party and 1-PartyEx are considerably better than those of the greedy heuristics, even for small values of n.

Figure 2.c shows the results for $h = 8$. Differently from the previous cases, here the dimension of the network varies till to $n = 30,000$. This is necessary to discover the value of n since when 1-Party behaves better than Randomized Prim. It follows that only when n is at least 18,000 1-PartyEx behaves better than Randomized Prim.

Basically, Randomized Prim has the best performance as h grows. This fact can be explained in the following way: The next point v to be connected to the tree by the Randomized Prim heuristic is chosen at random from all the points that are still not connected. So, this point can be also very far from its father in the tree even if its distance (in terms of number of hops) from the target can be very small. This implies that all the points close to v can be connected to v by small edges. On the contrary, if h is very small (3 or 5) it is still true that v can be very far from its father in the tree but the distance in terms of hops from v to the target can be h with a non negligible probability. Then, the points close to v cannot be connected to v by small edges.

On the contrary, in the Prim heuristic the next point to be connected to the graph is the closest one to the connected part. This implies that all the h available hops are consumed by small edges connecting points close to the target. Hence, the points distant from the target must be connected to the tree by longe edges, since they have to reach in one hop points belonging to a small region around the target.

The Kruskal heuristic gives good performance for relatively small values of n, but its behavior rapidly worsens as n grows. The reason of this behavior is due to the fact that this heuristic creates a number of connected components by using short edges. Such connected components are linked to the root by long edges. If n is sufficiently large with respect to h, the number of connected components (and of long edges) is very large.

The behavior of 1-Party strongly depends on the number of square cells produced. Each square cell induces an (eventually long) edge from some point inside the square to the target. The more square we have the more long edges are added in the solution. Since the number of square cells depends on h, this explains the "bad" behavior of 1-Party for large h. Notice that, with respect to the same number of points n, the number of square cells for $h = 8$ is almost twice the number of square cells for $h = 3$.

3.2 Case $\alpha = 2$

In this subsection we show the results of our experimental analysis that compares the performance of the previously described heuristics in the case $\alpha = 2$. The random points in these experiments are chosen according to both the uniform and the normal distribution.

As previously stated, the normal distribution simulates networks implemented in urban conglomerates. Since we are interested in describing situations in which several conglomerates simultaneously exist, we divide the region in equal-size squares each corresponding to a conglomerate. Then, inside each square we choose a set of points according to a normal distribution.

3.2.1 Uniform Distribution

In Figures 3.a-c we show the average performance of the heuristics for $\alpha = 2$. The points are chosen inside a region of size 1000×1000 according to the uniform distribution. In order to overcome the chance of selecting pairs of points at distance smaller than 1 from each other, we perform a normalization on the distances.

The results are summarized in Figure 3. They show that the asymptotic behavior of the heuristics emerges for larger values of n than those appeared in the case $\alpha = 1$. In particular, Figure 3.c shows that for $h = 8$ the RANDOMIZED PRIM behaves seriously better than all the other heuristics for $n < 30,000$, since when 2-PARTY becomes the best one.

This behavior is due to the same reasons that motivate the trend of the three heuristic for the Euclidean case with the difference that, when $\alpha = 2$, the presence of several long edges is even more crucial than in the case $\alpha = 1$.

3.2.2 Multi-Normal Distribution

In this subsection we describe the results of our tests obtained by choosing the position of the points in the plane according to a normal distribution. As already explained, in order to model the presence of several conglomerates we divide the plane into square cells. Inside each cell, we choose the points according to the normal distribution centered in the middle of the square. The variance of the distribution is $\ell/5$, where ℓ is the length of the squares. This choice guarantees that the points are not too concentrated in the center of the squares. The number of squares is chosen at random in every run. In order to guarantee a sufficient number of points inside each square, this number varies between 1 and $n/225$.

The obtained results show that the trend for the cases $h = 3, 5$ and 8 very similar to the case of uniform distribution shown in Figure 3.

3.3 Considerations about the running times

In Figure 4 we compare the running times[3] of the greedy heuristics for $n = 100, \ldots, 8000$. For all the greedy heuristic (and for all values of n) the best running time over all the instances with n points is displayed. Conversely, for α-PARTY and α-PARTYEX the worst running time is displayed. Notice that, since α-PARTY is orders of magnitude faster than the greedy heuristics, it cannot be displayed in Figure 4. In order to figure out the performance of α-PARTY we want to emphasize that the worst running time that we obtain for $n = 30,000$ was 0.13 seconds.

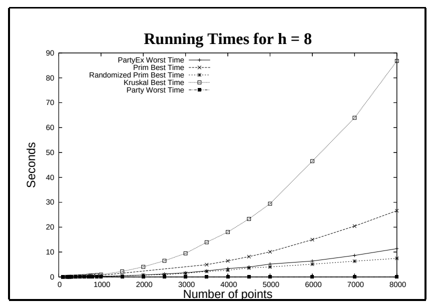

Figure 4: Heuristics running times.

4 Conclusions and open problems

The main open question is to refine the asymptotical analysis in order to obtain good bounds on the constant factors. In particular, it will be interesting to understand how the constant factors depend on h: as suggested by the experimental results, they can even exponentially depend on h.

From the experimental analysis, it seems that the performance ratio of the α-PARTY heuristic gets worse as h grows (see Figures 2 and 3). Is it possible to give a formal proof of that?

Finally, it will be interesting to extend our asymptotical analysis to non constant h (e.g. $h = \Omega(\log n)$).

[3]The heuristic has been implemented in C. The experiments has been executed on a Pentium® IV, 1.7 GHz with 256 MB of RAM, the operating system was Linux and the compiler was gcc.

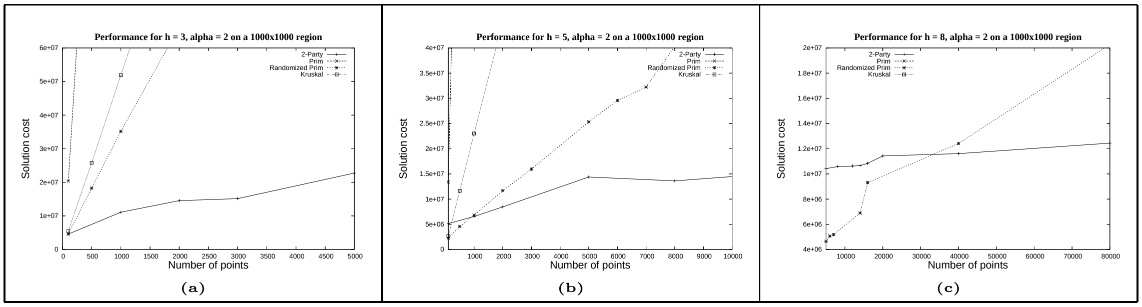

Figure 3: Performance of the heuristics for $\alpha = 2$, in the uniform distribution case.

References

[1] A. Abdalla, N. Deo, and P. Gupta. Random Tree Diameter and the Diameter Constrained MST. *Congressus Numerantium*, 144:161–182, 2000.

[2] L. Alfandari and V.T. Paschos. Approximating minimum spanning tree of depth 2. *Intl. Trans. In Op. Res.* 6:607-622, 1999.

[3] E. Althaus, S. Funke, S. Har-Peled, J. Koenemann, E. A. Ramos, M. Skutella, Approximation k-hop minimum-spanning trees. *Operations Research Letters*, 33 (2005) 115-120.

[4] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-medians and related problems. *Proc. 30-th ACM Symposium on Theory of Computing* 106-113, 1998.

[5] F.A. Chudak. Improved approximation algorithms for uncapacitated facility location problem. *Proc. of the 6-th Conference on Integer Programming and Combinatorial Optimization* 1998.

[6] A.E.F. Clementi, M. Di Ianni, A. Monti, G. Rossi, and R. Silvestri, Analysis of Fast Heuristics for the Minimum h-Hops Spanning Tree Problem on Random Instances of the Plane (draft version).

[7] A.E.F. Clementi, G. Huiban, P. Penna, G. Rossi, and Y.C. Verhoeven. Some Recent Theoretical Advances and Open Questions on Energy Consumption in Ad-Hoc Wireless Networks. In *Proc. of the 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks*, pages 23–38, 2002.

[8] A. Clementi, P. Penna and R. Silvestri. On The Power Range Assignment Problem on Radio Networks. Mobile Networks and Applications, 9(2): 125-140 (2004).

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

[10] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms* 31:228-248, 1999.

[11] L. Gouveia and C. Requejo. A new relaxation approach for the hop-constrain minimum spanning tree problem. *European Journal of Operational Research* 132:539-552, 2001.

[12] M. Haenggi, "Twelve Reasons not to Route over Many Short Hops", in IEEE Vehicular Technology Conference (VTC'04 Fall), (Los Angeles, CA), Sept. 2004.

[13] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power Consumption in Packet Radio Networks. *Theoretical Computer Science*, 243:289–305, 2000.

[14] M.R. Korupolu, C.G. Plaxton and R. Rajaraman. Analysis of a Local Search Heuristic for Facility Location Problems. *Proc. of the 9-th annual ACM-SIAM symposium on Discrete algorithms* 1-10, 1998.

[15] J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. of the American Mathematics Society*, vol. 7(1):48–50, 1956.

[16] M. Mahdian, Y. Ye and J. Zhang, A 1.52-approximation algorithm for the uncapacitated facility location problem. *Proc. of APPROX 2002* LNCS 2462: 229-242, 2002.

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

[18] R. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, vol. 36:1389–1401, 1957.

[19] G. R. Raidl and B. A. Julstrom, Greedy Heuristics and an Evolutionary Algorithm for the Bounded-Diameter Minimum Spanning Tree Problem. *Proc. of the 2003 ACM symposium on Applied computing* 747-752, 2003.

[20] D.B. Shmoys, E. Tardos and K. Aardal, Approximation algorithms for facility location problems. *Proc. of the 29-th Annual ACM Symposium on Theory of Computing* 265-274, 1997.

[21] A. Ephremides, G.D. Nguyen, and J.E. Wieselthier. On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. In *Proc. of the 19th INFOCOM*, 585–594, 2000.

[22] R. Bar-Yehuda, A. Israeli, and A. Itai. Multiple Communication in Multi-Hop Radio Networks. *SIAM Journal on Computing*, 22:875–887, 1993.