

Power Assignment Problems in Wireless Communication: Covering Points by Disks, Reaching few Receivers Quickly, and Energy-Efficient Travelling Salesman Tours

Stefan Funke¹, Sören Laue¹, Rouven Naujoks¹, and Zvi Lotker²

¹ Max-Planck-Institut für Informatik, Saarbrücken, Germany

² Ben Gurion University, Beer Sheva, Israel

Abstract. A fundamental class of problems in wireless communication is concerned with the assignment of suitable transmission powers to wireless devices/stations such that the resulting communication graph satisfies certain desired properties and the overall energy consumed is minimized. Many concrete communication tasks in a wireless network like broadcast, multicast, point-to-point routing, creation of a communication backbone, etc. can be regarded as such a power assignment problem.

This paper considers several problems of that kind; the first problem was studied before in [1,6] and aims to select and assign powers to k out of a total of n wireless network stations such that all stations are within reach of at least one of the selected stations. We show that the problem can be $(1+\epsilon)$ approximated by only looking at a small subset of the input, which is of size $O\left(\frac{k}{\epsilon^d} \frac{\alpha+1}{\epsilon^d}\right)$, i.e. independent of n and polynomial in k and $1/\epsilon$. Here d denotes the dimension of the space where the wireless devices are distributed, so typically $d \leq 3$ and α describes the relation between the Euclidean distance between two stations and the power consumption for establishing a wireless connection between them. Using this *coreset* we are able to improve considerably on the running time of $n^{((\alpha/\epsilon)^{O(d)})}$ for the algorithm by Bilo et al. at ESA'05 ([6]) actually obtaining a running time that is *linear* in n . Furthermore we sketch how outliers can be handled in our coreset construction.

The second problem deals with the energy-efficient, bounded-hop multicast operation: Given a subset C out of a set of n stations and a designated source node s we want to assign powers to the stations such that every node in C is reached by a transmission from s within k hops. Again we show that a coreset of size independent of n and polynomial in $k, |C|, 1/\epsilon$ exists, and use this to provide an algorithm which runs in time linear in n .

The last problem deals with a variant of non-metric TSP problem where the edge costs are the squared Euclidean distances; this problem is motivated by data aggregation schemes in wireless sensor networks. We show that a good TSP tour under Euclidean edge costs can be very bad in the squared distance measure and provide a simple constant approximation algorithm, partly improving upon previous results in [5], [4].

1 Introduction

Wireless network technology has gained tremendous importance in recent years. It not only opens new application areas with the availability of high-bandwidth connections for mobile devices, but also more and more replaces so far 'wired' network installations. While the spatial aspect was already of interest in the wired network world due to cable costs etc., it has far more influence on the design and operation of wireless networks. The power required to transmit information via radio waves is heavily correlated with the Euclidean distance of sender and receivers. Hence problems in this area are prime candidates for the use of techniques from computational geometry.

Wireless devices often have limited power supply, hence the energy consumption of communication is an important optimization criterion. In this paper we use the following simple geometric graph model: Given a set P of n points in \mathbb{R}^2 , we consider the complete graph $(P, P \times P)$ with edge weight $\omega(p, q) = |pq|^\alpha$ for some constant $\alpha > 1$ where $|pq|$ denotes the Euclidean distance between p and q . For $\alpha = 2$ the edge weights reflect the exact energy requirement for free space communication. For larger values of α (typically between 2 and 4), we get a popular heuristic model for absorption effects.

A fundamental class of problems in wireless communication is concerned with the assignment of suitable transmission powers to wireless devices/stations such that (1) the resulting communication graph satisfies a certain connectivity property Π , and (2) the overall energy assigned to all the network nodes is minimized. Many properties Π can be considered and have been treated in the literature before, see [8] for an overview. In this paper we consider several definitions of Π to solve the following problems:

k -Station Network/ k -disk Coverage: *Given a set S of stations and some constant k , we want to assign transmission powers to at most k stations (senders) such that every station in S can receive a signal from at least one sender.*

k -hop Multicast: *Given a set S of stations, a specific source station s , a set of clients/receivers $C \subseteq S$, and some constant k , we want the communication graph to contain a directed tree rooted at s spanning all nodes in C with depth at most k .*

TSP under squared Euclidean distance: *Given a set S of n stations, determine a permutation p_0, p_1, \dots, p_{n-1} of the nodes such that the total energy cost of the TSP tour, i.e. $\sum_{i=0}^{n-1} |p_i p_{(i+1) \bmod n}|^\alpha$ is minimized.*

1.1 Related Work

The k -Station Network Coverage problem was considered by Bilo et al. [6] as a k -disk cover, i.e. covering a set of n points in the plane using at most k disks such that the sum of the areas of the disks is minimized. They show that obtaining an exact solution is \mathcal{NP} -hard and provide a $(1 + \epsilon)$ approximation to this problem in time $n^{((\alpha/\epsilon)^{O(d)})}$ based on a plane subdivision and dynamic programming. Variants of the k -disk cover problem were also discussed in [1].

The general broadcast problem – assigning powers to stations such that the resulting communication graph contains a directed spanning tree and the total amount of energy used is minimized – has a long history. The problem is known to be \mathcal{NP} -hard ([7,8]), and for arbitrary, non-metric distance functions the problem can also not be approximated better than a log-factor unless $\mathcal{P} = \mathcal{NP}$ [14]. For the Euclidean setting in the plane, it is known ([2]) that the minimum spanning tree induces a power assignment for broadcast which is at most 6 times as costly as the optimum solution. This bound for a MST-based solution is tight ([7], [16]). There has also been work on restricted broadcast operations more in the spirit of the *k-hop multicast* problem we consider in this paper. In [3] the authors examine a *bounded-hop* broadcast operation where the resulting communication graph has to contain a spanning tree rooted at the source node s of depth at most k . They show how to compute an optimal k -hop broadcast range assignment for $k = 2$ in time $O(n^7)$. For $k > 2$ they show how to obtain a $(1 + \epsilon)$ -approximation in time $O(n^{O(\mu)})$ where $\mu = (k^2/\epsilon)^{2^k}$, that is, their running time is triply exponential in the number of hops k and this shows up in the exponent of n . In [10], Funke and Laue show how to obtain a $(1 + \epsilon)$ approximation for the k -hop broadcast problem in time doubly exponential in k based on a coreset which has size exponential in k , though.

The classical travelling salesperson problem is NPO-complete for arbitrary, non-metric distance functions (see [13]). However, if the metric satisfies the relaxed triangle inequality $\text{dist}(x, y) \leq \tau(\text{dist}(x, z) + \text{dist}(z, y))$ for $\tau \geq 1$ and every triple of points x, y and z as in our case, then a 4τ approximation exists [5].

General surveys of algorithmic range assignment problems can be found in [8,17,12].

1.2 Our Contribution

In Section 2 we show how to find a coreset of size **independent of n** and polynomial in k and $1/\epsilon$ for the k -Station Network Coverage/ k -Disk cover problem. This enables us improve the running time of the $(1 + \epsilon)$ approximation algorithm

by Bilo et al.[6] from $n^{((\alpha/\epsilon)^{O(d)})}$ to $O\left(n + \left(\frac{k \frac{d}{\alpha} + 1}{\epsilon^d}\right)^{\min\{2k, (\alpha/\epsilon)^{O(d)}\}}\right)$,

that is, we obtain a running time that is *linear* in n . We also present a variant that allows for the senders to be placed arbitrarily (not only within the given set of points) as well as a simple algorithm which is able to tolerate few outliers and runs in polynomial time for constant values of k and the number of outliers.

Also based on the construction of a (different) coreset of small size, we show in Section 3 how to obtain a $(1 + \epsilon)$ approximate solution to the k -hop multicast problem with respect to a constant-size set C of receivers/clients. Different from the solution for the k -hop broadcast problem presented in [10] we can exhibit a coreset of size *polynomial* in $k, 1/\epsilon$ and r . The approach in [10] requires a coreset of size exponential in k .

Finally, in Section 4 we consider the problem of finding energy-optimal TSP tours. The challenge here is that the edge weights induced by the energy costs

do not define a metric anymore; a simple example shows that an optimal solution to the Euclidean TSP can be a factor $\Omega(n)$ off the optimum solution. We present an $O(1)$ -approximation for the TSP problem with powers α of the Euclidean distance as edge weights. For small α we improve upon previous work by Andreae [4] and Bender and Chekuri [5].

2 Energy-Minimal Network Coverage or: "How to Cover Points by Disks"

Given a set S of points in \mathbb{R}^d and some constant k , we want to find at most k d -dimensional balls with radii r_i that cover all points in S while minimizing the objective function $\sum_{i=1}^k r_i^\alpha$ for some power gradient $\alpha \geq 1$. We distinguish two cases: the discrete case in which the ball centers have to be in S and the non-discrete case where the centers can be located arbitrarily in \mathbb{R}^d .

2.1 A Small Coreset for k -Disk Cover

In this section we describe how to find a coreset of size $O\left(\frac{k^{d/\alpha+1}}{\epsilon^d}\right)$, i.e. of size independent of n and polynomial in k and in $1/\epsilon$.

For now let us assume that we are given the cost of a λ -approximate solution S^λ for the point set S . We start by putting a regular d -dimensional grid on S with grid cell width δ depending on S^λ . For each cell C in the grid we choose an arbitrary representative point in $S \cap C$. We denote by R the set of these representatives. We say that C is *active* if $S \cap C \neq \emptyset$. Note that the distance between any point in $S \cap C$ and the representative point of C is at most $\sqrt{d} \cdot \delta$. In the following we write P^{OPT} for an optimal solution for a point set $P \subseteq S$. Now we obtain a solution R_S^{OPT} by increasing the disks in an optimal solution R^{OPT} by an additive term $\sqrt{d} \cdot \delta$. Since each point in S has a representative in R with distance at most $\sqrt{d} \cdot \delta$, R_S^{OPT} covers S . In the following we will show that (i) the cost of R_S^{OPT} is close to the cost of an optimal solution S^{OPT} to the original input set S and (ii) the size of the coreset R is small.

Theorem 1. *We have in the*

$$\begin{aligned} \text{non-discrete case:} \quad & \text{cost}(R_S^{OPT}) \leq (1 + \epsilon)^\alpha \cdot \text{cost}(S^{OPT}) \\ \text{discrete case:} \quad & \text{cost}(R_S^{OPT}) \leq (1 + \epsilon)^{2\alpha^2} \cdot \text{cost}(S^{OPT}) \end{aligned}$$

$$\text{for } \delta := \frac{1}{\sqrt{d}} \cdot \delta' \quad \text{where } \delta' := \frac{\epsilon}{k^{1/\alpha}} \left(\frac{\text{cost}(S^\lambda)}{\lambda} \right)^{1/\alpha}$$

Proof. Suppose R^{OPT} is given by k balls $(C_i)_{i \in \{1, \dots, k\}}$ with radii $(r_i)_{i \in \{1, \dots, k\}}$. Then

$$\frac{\text{cost}(R_S^{OPT})}{\text{cost}(S^{OPT})} = \frac{\sum_{i=1}^k (r_i + \sqrt{d} \cdot \delta)^\alpha}{\text{cost}(S^{OPT})} =: (I)$$

One can easily show that this term is maximized when $r_1 = r_i \forall i \in \{2, \dots, k\}$ (simply by differentiation). Thus we have

$$\begin{aligned} \frac{\text{cost}(R_S^{OPT})}{\text{cost}(S^{OPT})} &\leq k \cdot \frac{\left(\frac{\text{cost}(R^{OPT})^{1/\alpha}}{k^{1/\alpha}} + \frac{\epsilon}{k^{1/\alpha}} \left(\frac{\text{cost}(S^\lambda)}{\lambda}\right)^{1/\alpha}\right)^\alpha}{\text{cost}(S^{OPT})} \\ &= \frac{\left(\text{cost}(R^{OPT})^{1/\alpha} + \epsilon \left(\frac{\text{cost}(S^\lambda)}{\lambda}\right)^{1/\alpha}\right)^\alpha}{\text{cost}(S^{OPT})} \\ &\leq \left(\frac{\text{cost}(R^{OPT})^{1/\alpha}}{\text{cost}(S^{OPT})^{1/\alpha}} + \epsilon\right)^\alpha := (II) \end{aligned}$$

Now let us distinguish between the non-discrete and the discrete case. In the non-discrete case $(II) \leq (1 + \epsilon)$ following from the monotonicity of the problem, i.e. that for all subsets $P \subseteq S : \text{cost}(P^{OPT}) \leq \text{cost}(S^{OPT})$. This can easily be seen, since each feasible solution for S is also a feasible solution for P . In the discrete case $\text{cost}(R^{OPT})$ can be bigger than $\text{cost}(S^{OPT})$ - but not much as we will see: Given an optimal solution S^{OPT} . We transform S^{OPT} into a feasible solution S_P^{OPT} for R by shifting the ball centers to their corresponding representative point in R . Since some points can be uncovered now we have to increase the ball radii by an additional $\sqrt{d} \cdot \delta$. Following exactly the same analysis as above, in this case $\text{cost}(S_R^{OPT}) \leq (1 + \epsilon)^\alpha \cdot \text{cost}(S^{OPT})$. Hence $\text{cost}(R_S^{OPT}) \leq ((1 + \epsilon)^\alpha + \epsilon)^\alpha \leq (1 + \epsilon)^{2\alpha^2} \cdot \text{cost}(S^{OPT})$.

Knowing that the coresets is a good representation of the original input set S we will show now that R is also small.

Theorem 2. *The size of the computed coresets R is bounded by*

$$O\left(\frac{k^{\frac{d}{\alpha}+1} \cdot \lambda^{d/\alpha}}{\epsilon^d}\right)$$

Proof. Observe that the size of R is exactly given by the number of cells that contain a point in S . The idea is now to use an optimal solution S^{OPT} to bound the number of active cells. We can do so because any feasible solution for S covers all points in S and thus a cell C can only be active if such a solution covers fully or partially C . Thus the number of active cells $\#cc$ cannot be bigger than the volume of such a solution divided by the volume of a grid cell. To ensure that also the partially covered cells are taken into account we increase the radii by an additional term $\sqrt{d} \cdot \delta$. Thus consider S^{OPT} given by k balls $(C_i)_{i \in \{1, \dots, k\}}$ with radii $(r_i)_{i \in \{1, \dots, k\}}$. Then

$$\begin{aligned}
 \#cc &\leq \sum_{i=1}^k \frac{2^d \cdot (r_i + \sqrt{d} \cdot \delta)^d}{\delta^d} = (2\sqrt{d})^d \cdot \sum_{i=1}^k \left(1 + \frac{r_i}{\delta'}\right)^d \\
 &\stackrel{(*)}{\leq} (2\sqrt{d})^d \cdot \sum_{i=1}^k \left(1 + \frac{(k \cdot \lambda)^{1/\alpha}}{\epsilon} \cdot \left(\frac{\text{cost}(S^{OPT})}{\text{cost}(S^\lambda)}\right)^{1/\alpha}\right)^d \\
 &\leq (2\sqrt{d})^d \cdot k \cdot \left(1 + \frac{(k \cdot \lambda)^{1/\alpha}}{\epsilon}\right)^d \leq (4\sqrt{d})^d \cdot \left(\frac{k^{\frac{1}{\alpha} + \frac{1}{d}} \cdot \lambda^{1/\alpha}}{\epsilon}\right)^d \\
 &\in O\left(\frac{k^{\frac{d}{\alpha} + 1} \cdot \lambda^{d/\alpha}}{\epsilon^d}\right)
 \end{aligned}$$

where inequality (*) follows from the fact that $r_i \leq \text{cost}(S^{OPT})^{1/\alpha}$.

Note that it is easy to see that any $(1 + \omega)^\alpha$ -approximation algorithm yields a $(1 + \epsilon)$ -approximation algorithm by setting $\epsilon := \frac{\omega}{2^\alpha}$. Doing so increases our coreset by another *constant factor* of $(2^\alpha)^d$, i.e. the size of R does not change asymptotically.

2.2 Algorithms

Still we have to show how to approximate $\text{cost}(S^{OPT})$ for the construction of the grid. In [15] Feder et al. show how to compute deterministically a 2-approximate solution for the so called k -center problem in $O(n \log k)$ time. Furthermore Har-Peled shows in [11] how to obtain such an approximation in $O(n)$ expected time for $k = O(n^{1/3}/\log n)$. The k -center problem differs from the k -disk coverage problem just in the objective function which is given by $\max_{i=1..k} r_i^\alpha$ where the discs have radii r_i . Since $\max_{i=1..k} r_i^\alpha \leq \frac{1}{k} \cdot \sum_{i=1}^k r_i^\alpha$ a 2-approximation for the k -center problem is a $2k$ -approximation for the k -disk coverage problem. Using such an approximation the size of our coreset becomes $O\left(k^{\frac{2d}{\alpha} + 1}/\epsilon^d\right)$. Now we will show how to solve the coreset.

Discrete Version

Via Bilo et al.: Note that the discrete version of the k -disc cover problem can be solved by the approach of Bilo et. al. [6]. Recall that their algorithm runs $n^{((\alpha/\epsilon)^{O(d)})}$ time.

Via Exhaustive Search: Alternatively we can find an optimal solution in the following way. We consider all k -subsets of the points in the coreset R as the possible centers of the balls. Note that at least one point in R has to lie on the boundary of each ball in an optimal solution (otherwise you could create a better solution by shrinking a ball). Thus the number of possible radii for each ball is bounded by $n - k$. In total there are $(n - k)^k \cdot \binom{n}{k} \leq n^{2k}$ possible solutions. Hence

Corollary 1. *The running time of our approximation algorithm in the discrete is*

$$O \left(n + \left(\frac{k^{\frac{2d}{\alpha}+1}}{\epsilon^d} \right)^{\min \{ 2k, (\alpha/\epsilon)^{O(d)} \}} \right)$$

Non-Discrete Version Via Exhaustive Search: Note that on each ball D of an optimal solution there must be at least three points (or two points in diametrical position) that define D - otherwise it would be possible to obtain a smaller solution by shrinking D . Thus for obtaining an optimal solution via exhaustive search it is only necessary to check all k -sets of 3- respectively 2-subsets of S which yields a running time of $O(n^{3k})$. Hence

Theorem 3. *A $(1 + \epsilon)$ -approximate solution of the non-discrete k -disk coverage problem can be found in*

$$O \left(n + \left(\frac{k^{\frac{2d}{\alpha}+1}}{\epsilon^d} \right)^{3k} \right)$$

2.3 k -Disk Cover with Few Outliers

Assume we want to cover not all points by disks but we relax this constraint and allow a few points not to be covered, i.e. we allow let's say c outliers. This way, the optimal cover might have a considerably lower power consumption/cost.

Conceptually, we think of a k -disk cover with c outliers as a $(k + c)$ -disk cover with c disks having radius 0. Doing so, we can use the same coresets construction as above, replacing k by $k + c$. Obviously, the cost of an optimal solution to the $(k + c)$ -disk cover problem is a lower bound for the k -disk cover with c outliers. Hence, the imposed grid might be finer than actually needed. So snapping each point to its closest representative still ensures a $(1 + \epsilon)$ -approximation.

Constructed as above, the coresets has size $O(\frac{(k+c)^{\frac{2d}{\alpha}+1}}{\epsilon^d})$.

Again, there are two ways to solve this reduced instance, first by a slightly modified version of the algorithm proposed by Bilo et al. [6] and second by exhaustive search.

We will shortly sketch the algorithm by Bilo et al. [6] which is based on a hierarchical subdivision scheme proposed by Erlebach et al. in [9]. Each subdivision is assigned a level and they together form a hierarchy. All possible balls are also assigned levels depending on their size. Each ball of a specific level has about the size of an ϵ -fraction of the size of the cells of the subdivision of same level. Now, a cell in the subdivision of a fixed level is called relevant if at least one input point is covered by one ball of the same level. If a relevant cell S' is included in a relevant cell S and no larger cell S'' exists that would satisfy $S' \subseteq S'' \subseteq S$, then S' is called a child cell of S and S is called the parent of S' . This naturally defines a tree. It can be shown that a relevant cell has at most a constant number of child cells (the constant only depending on ϵ , α and d). The key ingredient

for the algorithm to run in polynomial time is the fact that there exists a nearly optimal solution where a relevant cell can be covered by only a constant number of balls of larger radius. The algorithm then processes all relevant cells of the hierarchical subdivision in a bottom-up way using dynamic programming. A table is constructed that for a given cell S , a given configuration P of balls having higher level than S (i.e. large balls) and an integer $i \leq k$ stores the balls of level at most the level of S (i.e. small balls) such that all input points in S are covered and the total number of balls is at most i . This is done for a cell S by looking up the entries of the child cells and iterating over all possible ways to distribute the i balls among them.

The k -disk cover problem with c outliers exhibits the same structural properties as the k -disk cover problem without outliers. Especially, the local optimality of the global optimal solution is preserved. Hence, we can adapt the dynamic programming approach of the original algorithm. In order for the algorithm to cope with c outliers we store not only one table for each cell but $c+1$ such tables. Each such table corresponds to the table for a cell S where $0, 1, \dots, c$ points are not covered. Now, we do not only iterate over all possible ways to distribute the i balls among its child cells but also all ways to distribute $l \leq c$ outliers. This increases the running time to $n^{((\alpha/\epsilon)^{O(d)})} \cdot c^{((\alpha/\epsilon)^{O(d)})} = n^{((\alpha/\epsilon)^{O(d)})}$. Hence running the algorithm on the coresets yields the following result:

Corollary 2. *We can compute a minimum k -disk cover with c outliers $(1 + \epsilon)$ approximately in time*

$$O \left(n + \left(\frac{(k+c)^{\frac{2d}{\alpha}+1}}{\epsilon^d} \right)^{(\alpha/\epsilon)^{O(d)}} \right).$$

For the exhaustive search approach we consider all assignments of k disks each having a representative as its center and one lying on its boundary. For each such assignment we check in time $O(kn)$ whether the number of uncovered points is at most c . We output the solution with minimal cost.

Corollary 3. *We can compute a minimum k -disk cover with c outliers $(1 + \epsilon)$ approximately in time*

$$O \left(n + k \left(\frac{(k+c)^{\frac{2d}{\alpha}+1}}{\epsilon^d} \right)^{2k+1} \right).$$

3 Bounded-Hop Multicast or: "Reaching Few Receivers Quickly"

Given a set S of points (stations) in \mathbb{R}^d , a distinguished source point $s \in S$ (sender), and a set $C \subset S$ of client points (receivers) we want to assign distances/ranges $r : S \rightarrow \mathbb{R}_{\geq 0}$ to the elements in S such that the resulting communication graph contains a tree rooted at s spanning all elements in C and

with depth at most k (an edge (p, q) is present in the communication graph iff $r(p) \geq |pq|$). The goal is to minimize the total assigned energy $\sum_{p \in S} r(p)^\alpha$. This can be thought of as the problem of determining an energy efficient way to quickly (i.e. within few transmissions) disseminate a message or a data stream to a set of few receivers in a wireless network.

As in the previous Section we will solve this problem by first deriving a coresets R of size independent of $|S| = n$ and then invoking an exhaustive search algorithm. We assume both k and $|C| = c$ to be (small) constants. The resulting coresets will have size *polynomial* in $1/\epsilon$, c and k . For few receivers this is a considerable improvement over the exponential-sized coresets that was used in [10] for the k -hop broadcast.

3.1 A Small Coresets for k -Hop Multicast

In the following we will restrict to the planar case in \mathbb{R}^2 , the approach extends in the obvious way to higher (but fixed) dimensions. Assume w.l.o.g. that the maximum distance of a point $p \in S$ from s is exactly 1. We place a square grid of cell width $\delta = \frac{1}{\sqrt{2}} \frac{\epsilon}{kc}$ on $[-1, 1] \times [-1, 1] \subset \mathbb{R}^2$. The size of this grid is $O(\frac{(kc)^2}{\epsilon^2})$. Now we assign each point in P to its closest grid point. Let R be the set of grid points that had at least one point from S snapped to it, C' the set of grid points that have at least one point from C snapped to it.

It remains to show that R is indeed a coresets. We can transform any given valid range assignment r for S (wrt receiver set C) into a valid range assignment r' for R (wrt receiver set C'). We define the range assignment r' for S as

$$r'(p') = \max_{p \text{ was snapped to } p'} r(p) + \sqrt{2}\delta.$$

Since each point p is at most $\frac{1}{\sqrt{2}}\delta$ away from its closest grid point p' we certainly have a valid range assignment for R . It is easy to see that the cost of r' for R is not much larger than the cost of r for S . We have:

$$\begin{aligned} \sum_{p' \in R} (r'(p'))^\alpha &= \sum_{p \in S} \left(\max_{p \text{ was snapped to } p'} r(p) + \sqrt{2}\delta \right)^\alpha \\ &\leq \sum_{p \in S} \left(\max_{p \text{ was snapped to } p'} r(p) + \frac{\epsilon}{kc} \right)^\alpha \\ &\leq \sum_{p \in S} \left(r(p) + \frac{\epsilon}{kc} \right)^\alpha. \end{aligned}$$

The relative error satisfies

$$\frac{\text{cost}(r')}{\text{cost}(r)} \leq \frac{\sum_{p \in S} (r(p) + \frac{\epsilon}{kc})^\alpha}{\sum_{p \in S} (r(p))^\alpha}.$$

Notice, that $\sum_{p \in P} r(p) \geq 1$ and r is positive for at most kc points p (each of the c receivers must be reached within k hops). Hence, the above expression is

maximized when $r(p) = \frac{1}{kc}$ (which can be seen simply by differentiation) for all points p that are assigned a positive value. Thus

$$\frac{\text{cost}(r')}{\text{cost}(r)} \leq \frac{(kc) \cdot \left(\frac{1}{kc} + \frac{\epsilon}{kc}\right)^\alpha}{(kc) \cdot \left(\frac{1}{kc}\right)^\alpha} = (1 + \epsilon)^\alpha.$$

On the other hand we can transform any given valid range assignment r' for R into a valid range assignment r for S as follows. We select for each grid point $g \in R$ one representative g_S from S that was snapped to it. For the grid point to which s (the source) was snapped we select s as the representative. If we define the range assignment r for S as $r(g_S) = r'(g) + \sqrt{2}\delta$ and $r(p) = 0$ if p does not belong to the chosen representatives, then r is a valid range assignment for S because every point is moved by at most $\delta/\sqrt{2}$. Using the same reasoning as above we can show that $\text{cost}(r) \leq (1 + \epsilon)^\alpha \text{cost}(r')$. In summary we obtain the following theorem:

Theorem 4. *For the k -hop multicast problem with c receivers there exists a coreset of size $O\left(\frac{(kc)^2}{\epsilon^2}\right)$.*

3.2 Solution Via a Naive Algorithm

As we are not aware of any algorithm to solve the k -hop multicast problem we employ a naive brute-force strategy, which we can afford since after the coreset computation we are left with a 'constant' problem size. Essentially we consider every kc -subset of R as potential set of senders and try out the $|R|$ potential ranges for each of the senders. Hence, naively there are at most $\left(\frac{kc^2}{\epsilon^2}\right) \cdot \left(\frac{kc^2}{\epsilon^2}\right)^{kc}$ different range assignments to consider at all. We enumerate all these assignments and for each of them check whether the range assignment is valid wrt \mathcal{C}' ; this can be done in time $|R|$. Of all the valid range assignments we return the one of minimal cost.

Assuming the floor function a coreset R for an instance of the k -hop multicast problem can be constructed in linear time. Hence we obtain the following corollary:

Corollary 4. *A $(1 + \epsilon)$ -approximate solution to the k -hop multicast problem on n points in the plane can be computed in time $O\left(n + \left(\frac{kc}{\epsilon}\right)^{4kc}\right)$.*

As we are only after an approximate solution, we do not have to consider all $|R|$ potential ranges but can restrict to essentially $O\left(\log_{1+\epsilon} \frac{kc}{\epsilon}\right)$ many, the running time of the algorithm improves accordingly:

Corollary 5. *A $(1 + \epsilon)$ -approximate solution to the k -hop multicast problem on n points in the plane can be computed in time $O\left(n + \left(\frac{(kc)^2 \log \frac{kc}{\epsilon}}{\epsilon^3}\right)^{kc}\right)$.*

4 Information Aggregation Via Energy-Minimal TSP Tours

While early wireless sensor networks (WSNs) were primarily data collection systems where sensor readings within the network are all transferred to a central computing device for evaluation, current WSNs perform a lot of the data processing *in-network*. For this purpose some nodes in the network might be interested in periodically *collecting* information from certain other nodes, some nodes might want to *disseminate* information to certain groups of other nodes. A typical approach for data collection and dissemination as well as for data aggregation purposes are tree-like subnetwork topologies, they incur certain disadvantages with respect to load-imbalance as well as non-obliviousness to varying initiators of the data collection or dissemination operation, though. Another, very simple approach could be to have a *virtual token* floating through the network (or part thereof). Sensor nodes can attach data to the token or read data from the token and then hand it over to the next node. Preferably the token should not visit a node again before all other nodes have been visited and this should happen in an energy-optimal fashion, i.e. the sum of the energies to hand over the token to the respective next node should be minimized. Such a scheme has some advantages: first of all none of sensor nodes plays a distinguished role – something that is desirable for a system of homogeneous sensor nodes – furthermore every sensor node can use the same token to initiate his data collection/dissemination operation. Abstractly speaking we are interested in finding a *Travelling Salesperson tour* (TSP) of minimum energy cost for (part of) the network nodes. Unfortunately, the classical TSP with non-metric distance function is very hard to solve (see [13]). However, for metrics satisfying the relaxed triangle inequality $\text{dist}(x, y) \leq \tau(\text{dist}(x, z) + \text{dist}(z, y))$ for $\tau \geq 1$ and every triple of points x, y and z a 4τ approximation exists [5].

In this section we show that the 'normal' Euclidean TSP is not suitable for obtaining an energy-efficient tour and devise a 6-approximation algorithm for TSP under squared Euclidean metric.

4.1 Why Euclidean TSP Does Not Work

Simply computing an optimal tour for the underlying Euclidean instance does not work. The cost for such a tour can be a factor $\Omega(n)$ off from the optimal solution for the energy-minimal tour. Consider the example where n points lie on a slightly bent line and each point having distance 1 to its right and left neighbor. An optimal Euclidean tour would visit the points in their linear order and then go back to the first point. Omitting the fact that the line is slightly bent this tour would have a cost of $(n-1) \cdot 1^2 + (n-1)^2 = n(n-1)$ if the edge weights are squared Euclidean distances. However, an optimal energy-minimal tour would have a cost of $(n-2) \cdot 2^2 + 2 \cdot 1^2 = 4(n-1) + 2$. This tour would first

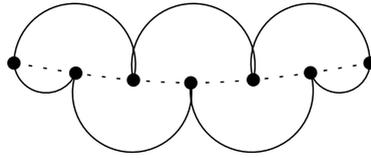


Fig. 1. An optimal energy-minimal tour for points on a line

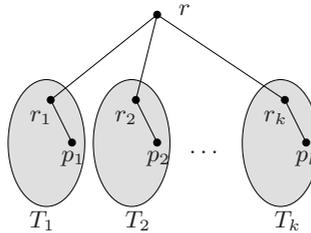


Fig. 2. Tree T and its children trees T_1, T_2, \dots, T_k

visit every second point on the line and on the way back all remaining points as in figure 1.

4.2 A 6-Approximation Algorithm

In this section we will describe an algorithm which computes a 6-approximation for the TSP under squared Euclidean distance. Obviously, the cost of a minimum spanning tree is a lower bound for the optimal value OPT of the tour. Consider a non-trivial minimum spanning tree T for a graph with node set V and squared Euclidean edge weights. We denote the cost of such a tree by $MST(T)$. Let r be the root of T and p be one child of T .

We define two Hamiltonian paths $\pi^a(T)$ and $\pi^b(T)$ as follows. Let $\pi^a(T)$ be a path starting at r , finishing at p that visits all nodes of T and the cost of this path is at most $6 MST(T) - 3\|rp\|^2$. Let $\pi^b(T)$ be defined in the same way but in opposite direction, i.e. it starts at p and finishes at r .

Now, if we have such a tour $\pi^a(T)$ for the original vertex set V we can construct a Hamilton tour by connecting r with p . The cost of this tour is clearly at most $6 MST(T) - 3\|rp\|^2 + \|rp\|^2 \leq 6 MST(T) \leq 6 OPT$. It remains to show how to construct such tours π^a and π^b . We will do this recursively.

For a tree T of height 1, i.e. a single node r , $\pi^a(T)$ and $\pi^b(T)$ both consist of just the single node. Conceptually, we identify p with r in this case. Obviously, the cost of both paths is trivially at most $6 MST(T) - 3\|rp\|^2$.

Now, let T be of height larger than 1 and let T_1, \dots, T_k be its children trees. Let r denote the root of T and r_i the root of T_i and p_i be a child of T_i as in figure 2. Then we set $\pi^a(T) = (r, \pi^b(T_1), \pi^b(T_2), \dots, \pi^b(T_k))$.

The cost of the path $\pi^a(T)$ satisfies

$$\begin{aligned}
 \text{cost}(\pi^a(T)) &= \|r p_1\|^2 + \text{cost}(\pi^b(T_1)) + \|r_1 p_2\|^2 + \text{cost}(\pi^b(T_2)) + \dots \\
 &\quad + \|r_{k-1} p_k\|^2 + \text{cost}(\pi^b(T_k)) \\
 &\leq (\|r r_1\| + \|r_1 p_1\|)^2 + \text{cost}(\pi^b(T_1)) \\
 &\quad + (\|r_1 r\| + \|r r_2\| + \|r_2 p_2\|)^2 + \text{cost}(\pi^b(T_2)) \\
 &\quad \vdots \\
 &\quad + (\|r_{k-1} r\| + \|r r_k\| + \|r_k p_k\|)^2 + \text{cost}(\pi^b(T_k)) \\
 &\leq 2\|r r_1\|^2 + 2\|r_1 p_1\|^2 + \text{cost}(\pi^b(T_1)) \\
 &\quad + 3\|r_1 r\|^2 + 3\|r r_2\|^2 + 3\|r_2 p_2\|^2 + \text{cost}(\pi^b(T_2)) \\
 &\quad \vdots \\
 &\quad + 3\|r_{k-1} r\|^2 + 3\|r r_k\|^2 + 3\|r_k p_k\|^2 + \text{cost}(\pi^b(T_k)) \\
 &\leq 6 \sum_{i=1}^k \|r r_i\|^2 + 3 \sum_{i=1}^k \|r_i p_i\|^2 + \sum_{i=1}^k \text{cost}(\pi^b(T_i)) - 3\|r r_k\|^2 \\
 &\leq 6 \sum_{i=1}^k \|r r_i\|^2 + 6 \sum_{i=1}^k \text{MST}(T_i) - 3\|r r_k\|^2 \\
 &= 6 \text{MST}(T) - 3\|r r_k\|^2.
 \end{aligned}$$

In the above calculation we used the fact that $(\sum_{i=1}^n a_i)^\alpha \leq n^{\alpha-1} \cdot \sum_{i=1}^n a_i^\alpha$, for $a_i \geq 0$ and $\alpha \geq 1$, which follows from Jensen’s inequality and the fact that the function $f : x \mapsto x^\alpha$ is convex. The path $\pi^b(T)$ is constructed analogously.

In fact, the very same construction and reasoning can be generalized to the following corollary.

Corollary 6. *There exists a $2 \cdot 3^{\alpha-1}$ -approximation algorithm for the TSP if the edge weights are Euclidean edge weights to the power α .*

The metric with Euclidean edge weights to the power α satisfies the relaxed triangle inequality with $\tau = 2^{\alpha-1}$. A short computation shows that our algorithm is better than previous algorithms [5,4] for small α , i.e. for $2 \leq \alpha \leq 2.7$.

References

1. Alt, H., Arkin, E.M., Brönnimann, H., Erickson, J., Fekete, S.P., Knauer, C., Lenchner, J., Mitchell, J.S.B., Whittlesey, K.: Minimum-cost coverage of point sets by disks. In: SCG 2006: Proc. 22nd Ann. Symp. on Computational Geometry, pp. 449–458. ACM Press, New York (2006)
2. Ambühl, C.: An optimal bound for the mst algorithm to compute energy-efficient broadcast trees in wireless networks. In: ICALP, pp. 1139–1150 (2005)
3. Ambühl, C., Clementi, A.E.F., Ianni, M.D., Lev-Tov, N., Monti, A., Peleg, D., Rossi, G., Silvestri, R.: Efficient algorithms for low-energy bounded-hop broadcast in ad-hoc wireless networks. In: STACS, pp. 418–427 (2004)

4. Andreae, T.: On the traveling salesman problem restricted to inputs satisfying a relaxed triangle inequality. *Networks* 38(2), 59–67 (2001)
5. Bender, M.A., Chekuri, C.: Performance guarantees for the TSP with a parameterized triangle inequality. *Information Processing Letters* 73(1–2), 17–21 (2000)
6. Bilò, V., Caragiannis, I., Kaklamanis, C., Kanellopoulos, P.: Geometric clustering to minimize the sum of cluster sizes. In: *European Symposium on Algorithms (ESA)*, pp. 460–471 (2005)
7. Clementi, A.E.F., Crescenzi, P., Penna, P., Rossi, G., Vocca, P.: On the complexity of computing minimum energy consumption broadcast subgraphs. In: *STACS*, pp. 121–131 (2001)
8. Clementi, A.E.F., Huiban, G., Penna, P., Rossi, G., Verhoeven, Y.C.: Some recent theoretical advances and open questions on energy consumption in ad-hoc wireless networks. In: *Proc. 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks (ARACNE)*, pp. 23–38 (2002)
9. Erlebach, T., Jansen, K., Seidel, E.: Polynomial-time approximation schemes for geometric graphs. In: *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA 2001)*, Washington, DC, pp. 671–679 (2001)
10. Funke, S., Laue, S.: Bounded-Hop Energy-Efficient Broadcast in Low-Dimensional Metrics Via Coresets. In: Thomas, W., Weil, P. (eds.) *STACS 2007*. LNCS, vol. 4393, pp. 272–283. Springer, Heidelberg (2007)
11. Har-Peled, S.: Clustering motion. *Discrete Comput. Geom.* 31(4), 545–565 (2004)
12. Kirousis, L.M., Kranakis, E., Krizanc, D., Pelc, A.: Power consumption in packet radio networks. *Theor. Comput. Sci.* 243(1-2), 289–305 (2000)
13. Orponen, P., Mannila, H.: On approximation preserving reductions: Complete problems and robust measures. Technical Report, University of Helsinki (1990)
14. Guha, S., Khuller, S.: Improved methods for approximating node weighted steiner trees and connected dominating sets. *Information and Computation* 150, 57–74 (1999)
15. Feder, T., Greene, D.: Optimal algorithms for approximate clustering. In: *Proc. 20th ACM Symp. on Theory of Computing* (1988)
16. Wan, P.-J., Calinescu, G., Li, X., Frieder, O.: Minimum-energy broadcast routing in static ad hoc wireless networks. In: *INFOCOM*, pp. 1162–1171 (2001)
17. Wieselthier, J.E., Nguyen, G.D., Ephremides, A.: On the construction of energy-efficient broadcast and multicast trees in wireless networks. In: *INFOCOM*, pp. 585–594 (2000)