International Journal of Foundations of Computer Science
Vol. 22, No. 8 (2011) 1949–1969

DOI: 10.1142/S0129054111009148

World Scientific
www.worldscientific.com

APPROXIMATION ALGORITHMS FOR BUY-AT-BULK GEOMETRIC NETWORK DESIGN*,†

ARTUR CZUMAJ

Centre for Discrete Mathematics and its Applications (DIMAP) and Department of Computer Science
University of Warwick, UK
a.czumaj@warwick.ac.uk

JUREK CZYZOWICZ

Departement d'Informatique, Universite du Quebec en Outaouais, Gatineau, Quebec J8X 3X7, Canada
jurek.czyzowicz@uqo.ca

LESZEK GĄSIENIEC

Department of Computer Science, University of Liverpool, Peach Street, L69 7ZF, UK
l.a.gasieniec@liverpool.ac.uk

JESPER JANSSON

Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo-112-8610, Japan
jesper.jansson@ocha.ac.jp

ANDRZEJ LINGAS

Department of Computer Science, Lund University, 22100 Lund, Sweden
andrzej.lingas@cs.lth.se

PAWEL ZYLINSKI

Institute of Computer Science, University of Gdańsk, 80-952 Gdańsk, Poland
pawel.zylinski@inf.univ.gda.pl

Received 6 May 2011
Accepted 25 August 2011
Communicated by Subhash Suri

The buy-at-bulk network design problem has been extensively studied in the general graph model. In this paper, we consider *geometric* versions of the problem, where all points in a Euclidean space are candidates for network nodes, and present the first general approach for solving them. It enables us to

*A preliminary version of this article appeared in *Proceedings of the 11th International Symposium on Algorithms and Data Structures* (WADS 2009), Volume 5664 of *Lecture Notes in Computer Science*, pp. 168–180, Springer-Verlag, 2009.

†Research supported in part by VR grant 621-2008-4649, the Royal Society IJP - 2006/R2, the Centre for Discrete Mathematics and its Applications (DIMAP), EPSRC award EP/D063191/1, the Special Coordination Funds for Promoting Science and Technology (Japan), and the Visby Programme Scholarship 01224/2007.

obtain quasi-polynomial-time approximation schemes for basic variants of the buy-at-bulk geometric network design problem with polynomial total demand. Then, for instances with a single sink and low capacity links, we design fast polynomial-time, low-constant approximation algorithms.

Keywords: Geometric networks; buy-at-bulk; approximation algorithm; quasi-polynomial-time approximation scheme (QPTAS); Hanan grid; dynamic programming; belt decomposition.

1. Introduction

Consider a water heating company planning to construct a network of pipelines to carry warm water from a number of heating stations to a number of buildings. The company can install different types of pipes of various diameters and prices per unit of length. Typically, the prices grow with the diameter, while the ratio between the price per unit of length and the pipe throughput capacity decreases. The natural goal of the company is to minimize the total cost of pipes sufficient to construct a network that could carry the warm water to the buildings, assuming a fixed water supply at each source. Similar problems are faced by oil companies that need to transport oil to refineries, or telecommunication companies that need to buy capacities (in bulk) from a phone company.

The common difficulty of these problems is that only a limited set of link types (e.g., pipes) is available so that the price of installing a link to carry some volume of supply between its endpoints does not grow linearly with the volume, but has a discrete character. Even if there is only one type of link available, and that link type has enough capacity to carry the entire supply, the problem is NP-hard as it includes the minimum Steiner tree problem. Since the geometric versions of the latter problem are known to be strongly NP-complete [15], they cannot admit fully polynomial-time approximations schemes in the geometric setting [6, 15].

In operations research, these types of problems are often termed as *discrete cost network optimization* [8, 30], whereas in computer science as *minimum cost network* (or, *link/edge*) *installation problems* [34] or as *buy-at-bulk network design* [7]; we shall use the latter term. In computer science, the buy-at-bulk network design problem was introduced by Salman *et al.* [34], who also argued that a key special case is when the underlying graph is defined by points in the Euclidean plane because many optimization problems have a natural embedding in the plane. Since then, several variants of buy-at-bulk network design have been extensively studied in the *graph model* (as opposed to the geometric setting); see [2, 3, 7, 9, 10, 11, 14, 16, 17, 18, 19, 21, 22, 24, 26, 29, 35, 36]. However, the geometric versions of buy-at-bulk network design problems remain largely unexplored, despite their potential practical importance.

Depending on whether or not the entire supply at each source is required to follow a single path to a sink, the problem variants are further classified as *non-divisible* or *divisible* [34]. In terms of the warm water supply problem, the divisible graph model means that the possible locations of the pipes and their splits or joints are given a priori. In this paper, we consider the following divisible, geometric variants of the buy-at-bulk network design problem:

- ▷ *Buy-at-bulk geometric network design (**BGND**)*: For a given set of different link types and a given set of sources and sinks placed in a Euclidean space, construct a minimum cost geometric network sufficient to carry the integral supply at the sources to the sinks.

▷ *Buy-at-bulk single-sink geometric network design (**BSGND**)*: For a given set of different link types, a given single sink, and a given set of sources, construct a minimum cost geometric network sufficient to carry the integral supply at the sources to the sink.

Motivated by settings in which the underlying network must possess some basic structural properties, we also distinguish special versions of both problems where each edge of the network has to be parallel to one of the coordinate system axes, and term them as *buy-at-bulk rectilinear network design (**BRND**)* and *buy-at-bulk single-sink rectilinear network design (**BSRND**)*, respectively.

1.1. *Previous work*

Salman *et al.* [34] initiated the algorithmic study of the single-sink buy-at-bulk network design problem. They argued that the problem is especially relevant in the *geometric* case, and provided a polynomial-time approximation algorithm for the indivisible variant of BSGND on the input Euclidean graph with an approximation ratio of $O(\log(D/c_1))$, where D is the total supply and c_1 is the smallest link capacity. Their model differs somewhat from ours in that they only allowed some points of the plane to be used by the solution, whereas we allow the entire space to be used. To the best of our knowledge, the geometric versions of buy-at-bulk problems have only been studied previously by Salman *et al.* [34].

Salman *et al.* [34] also gave a constant-factor approximation for *general graphs* in the case where only one sink and one type of links is available; this approximation ratio was improved by Hassin *et al.* [24]. Mansour and Peleg [28] provided an $O(\log n)$-approximation for the multi-sink buy-at-bulk network design problem when only one type of link is available. Awerbuch and Azar [7] were the first to give a polylogarithmic approximation for the general graph case even when different sources have to communicate with different sinks. In the *single-sink buy-at-bulk* network design problem for general graphs, Garg *et al.* [16] designed an $O(K)$-approximation algorithm, where K is the number of link types, and Guha *et al.* [21] gave the first constant-factor approximation algorithm for the (non-divisible) variant of the problem. This constant has been reduced in a sequence of papers [19, 22, 20, 26, 35] to reach the approximation ratio 20.41 for the divisible variant, and 40.82 for the non-divisible variant (see [20]).

Further generalizations of the buy-at-bulk network design problem in the graph model have also been studied. For example, Meyerson *et al.* [29] introduced *non-uniform* buy-at-bulk network design, where each edge has its own cost function so that the price of installing a link also depends on the edge that it uses, and Chekuri *et al.* [10] later gave the first polynomial-time polylogarithmic approximation ratio algorithm for this problem. As another example, Goel and Estrin [17] and Goel and Post [18] considered *simultaneous* buy-at-bulk, where the costs per unit of length for links may be unknown or known to change over time and the objective is to find a solution that is expected to be a good approximation for all possible cost functions at the same time. For even more results on buy-at-bulk network design in the graph model, see, e.g., $[2, 3, 11, 14, 36]$.

A classical approach for approximation algorithms for geometric optimization problems builds on the techniques developed for polynomial-time approximation schemes (PTAS) for geometric optimization problems due to Arora $[4]$ and Mitchell $[31]$. In this approach, to obtain a $(1+\varepsilon)$-approximation in $n^{O(1/\varepsilon)}$ time, where n denotes the size of an input instance and ε is any given constant $0 < \varepsilon < 1$, one first transforms the problem to an integer grid of polynomial size and then recursively partitions the grid into *dissection squares* using a quadtree of logarithmic depth. The next step is to prove the so-called Structure Theorem, which guarantees that there exists an almost-optimal solution that for each dissection square crosses its boundary only a few times and only in a number (depending on $\frac{1}{\varepsilon}$) of prespecified portals. Finally, dynamic programming is employed over the recursive decomposition to find a solution satisfying the Structure Theorem. In particular, this method has been successfully applied to solve the minimum Euclidean Steiner tree problem $[4]$, which can be considered as a very restricted case of BGND. Therefore, an important question is whether these techniques can be applied to even more general buy-at-bulk geometric network design problems.

1.2. *Our contributions and techniques*

In this paper, we demonstrate how to take advantage of the structural properties of Euclidean space to obtain more efficient approximation algorithms for buy-at-bulk network design problems in the geometric setting than in the well-studied graph model.

Unfortunately, it is not possible to directly apply the techniques developed by Arora $[4]$ and Mitchell $[31]$ to create a PTAS for the BGND problem. The main difficulty with the application of Arora's $[4]$ and Mitchell's $[31]$ techniques to the general BGND problem lies in the reduction of the number of crossings on the boundaries of the dissection squares. This is because we cannot limit the number of crossings of a boundary of a dissection square below the integral amount of supply it carries into that square. On the other hand, we can significantly limit the number of crossing locations at the expense of a slight increase in the network cost. However, with this relaxed approach we can only achieve *quasi-polynomial* upper bounds (rather than polynomial upper bounds) on the number of subproblems on the dissection squares in the dynamic programming phase, except for in some very special cases (cf. $[5]$). Furthermore, the subproblems, in particular the leaf ones, become much more difficult. Nevertheless, we can solve them exactly in the case of BRND in super-polynomial time by using our exponential-time algorithm for this problem. As a result, we obtain a randomized *quasi-polynomial-time approximation scheme* (QPTAS) for the *divisible buy-at-bulk rectilinear network design problem* in the Euclidean plane with polynomially bounded total supply. Our result can be derandomized and generalized to include $O(1)$-dimensional Euclidean space. This implies that the aforementioned variant of buy-at-bulk geometric network design is not *APX*-hard, unless $SAT \in DTIME[n^{\log^{O(1)} n}]$.

Our result is then used to obtain fast low-constant-factor approximation algorithms for single-sink variants. We develop a new "belt decomposition technique" for the single-sink case, and then apply the QPTAS above. It yields a $(2+\varepsilon)$-approximation for the divisible buy-at-bulk rectilinear network design problem in the Euclidean plane that is fast if the capacities of links are small, and a $(\sqrt{8}+\varepsilon)$-approximation algorithm for the corresponding BGND problem in the Euclidean plane. In comparison, the best known approximation ratio achievable in polynomial time for single-sink divisible buy-at-bulk network design in the graph model is 20.41 [20].

1.3. *Paper organization*

In Section 2, we define the geometric variants of the buy-at-bulk network design problems studied in this paper, establish the NP-hardness of restricted versions of BSGND and BSRND, and provide a useful characterization of optimal solutions to BRND. In Section 3, we give an exact (exponential-time) algorithm for BRND. Next, in Section 4, we present a QPTAS for BRND in the Euclidean plane with polynomially bounded total demand, which generalizes to higher dimensions and yields an arbitrarily close to $\sqrt{2}$-approximation for the corresponding variants of BGND. In Section 5, we derive fast, low-constant approximations for cases of single sinks and low edge capacities. We conclude with some final remarks in Section 6.

2. Preliminaries

2.1. *Problem definitions*

Consider a Euclidean d-dimensional space $\mathbb{E}^d$. Let $\{s_1, \ldots, s_{n_s}\}$ be a given set of n_s points in $\mathbb{E}^d$ called *sources*, and let $\{t_1, \ldots, t_{n_t}\}$ be a given set of n_t points in $\mathbb{E}^d$ called *sinks*. Define $n = n_s + n_t$. Each source s_i supplies some integral *demand* $d(s_i)$ to the sinks, and each sink t_j is required to receive some integral *demand* $d(t_j)$ from the sources. The sums $\sum_i d(s_i)$ and $\sum_j d(t_j)$ are assumed to be equal, and their value is called the *total demand* D. Furthermore, there are K given types of links, each type with a fixed cost and capacity. The *capacity* of a link of type i is denoted by c_i and the *cost* of placing a link e of ith type and length $|e|$ is $|e| \cdot \delta_i$. We assume that the types of the links are ordered so that $c_1 < \cdots < c_K,\ \ \delta_1 < \cdots < \delta_K$, and $\frac{\delta_1}{c_1} > \cdots > \frac{\delta_K}{c_K}$, since otherwise we can eliminate some types of links [34].

A *geometric network* is a directed, weighted multigraph embedded in a d-dimensional Euclidean space $\mathbb{E}^d$. The objective of the *buy-at-bulk geometric network design problem* (**BGND**) is to construct a geometric network G satisfying:

- All of the sources $\{s_1, \ldots, s_{n_s}\}$ and sinks $\{t_1, \ldots, t_{n_t}\}$ belong to the set of vertices of G (the additional vertices of G are called *Steiner vertices*).
- Every copy of a multiedge in G is of one of the K available link types.
- For $\ell = \{1, \ldots, D\}$, there exists a supply-demand path (*sd-path* for short) P_ℓ from a source s_i to a sink t_j such that:
  - Each source s_i is a start point of $d(s_i)$ sd-paths;
  - Each sink t_j is an endpoint of $d(t_j)$ sd-paths;

- For each directed multiedge f of G, the number of sd-paths passing through f is less than or equal to the total capacity of all links in f.

that minimizes the total cost of all links.

If the set of sinks is a singleton then the problem is termed as the *buy-at-bulk single-sink geometric network design problem* (**BSGND** for short). If the multigraph is required to be rectilinear, i.e., only vertical and horizontal edges are allowed, then the problem is termed as the *buy-at-bulk rectilinear network design problem* (**BRND** for short), and its single-sink version is abbreviated as **BSRND**.

Example 1. *Suppose that we are given the instance of BRND shown in Fig. 1(a) consisting of two sources* $\{s_1, s_2\}$ *and two sinks* $\{t_1, t_2\}$ *with* $d(s_1) = 6,\ d(s_2) = 5,\ d(t_1) = 6$*, and* $d(t_2) = 5$*, and suppose that two types of links are available:*

- *Type 1 link: Capacity* $c_1 = 1,$ *cost per unit of length* $\delta_1 = 2,$
- *Type 2 link: Capacity* $c_2 = 10,$ *cost per unit of length* $\delta_2 = 7.$

Then, the geometric network in Fig. 1(b) has total cost $8 \cdot \delta_1 \cdot 6 + 8 \cdot \delta_1 \cdot 5 = 176$*, while the geometric network in Fig. 1(c) has total cost* $8 \cdot \delta_1 \cdot 1 + 1 \cdot \delta_2 \cdot 4 + 8 \cdot \delta_2 \cdot 1 = 100$. □

Note that we allow divisible (splittable) demands, i.e., the entire demand originating at a source s_i does not have to follow the same route. For simplicity, we assume in this paper that the Euclidean space under consideration is a Euclidean plane $\mathbb{E}^2$, although the majority of our results can be generalized to any Euclidean $O(1)$-dimensional space.

2.2. *Hardness of BSGND and BSRND*

The NP-hard problem of constructing a minimum (rectilinear) Steiner tree for a set of n points in the Euclidean plane (see [15]) can be regarded as a special case of BSGND (BSRND, respectively) with one link type. More precisely, let $n-1$ of the points be sources, each of demand 1, and let the remaining point be a sink with demand $n-1$. Allow only one type of link and set its capacity c_1 to n. This reduction yields:

Lemma 2. *BSGND and BSRND are NP-hard even if only one type of link is allowed and the total demand is polynomially bounded in the total number of sources and sinks.*

2.3. *Structural properties of solutions to rectilinear buy-at-bulk network design (BRND)*

Zachariasen [37] showed that several variants and generalizations of the minimum rectilinear Steiner problem in the Euclidean plane are solvable on the *Hanan grid* [23] of the input points, defined as the grid formed by the vertical and horizontal straight lines passing through these points. The next lemma extends this to BRND.

Lemma 3. *Any optimal solution to BRND in the plane can be converted into a planar multigraph in which sd-paths do not cross and where all vertices lie on the Hanan grid.*

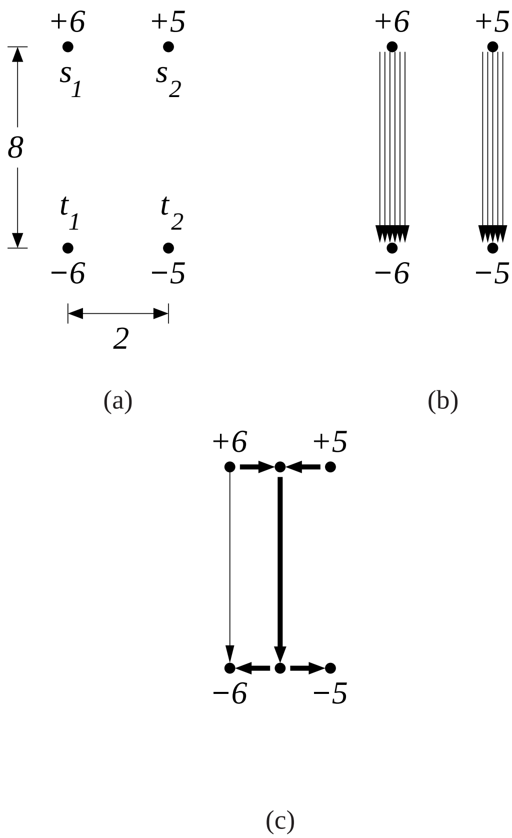

Fig. 1. An example of BRND.

Proof. Consider an optimal solution to any instance of BRND in the plane.

First, eliminate all crossings between sd-paths in the solution as follows. At each point p where two sd-paths P_1 and P_2 intersect, place a Steiner vertex. Let P_1^- and P_1^+ be the parts of P_1 leading to and from p, respectively, and similarly for P_2^- and P_2^+. In case the angle between P_1^- and P_1^+ is 180 degrees and the angle between P_2^- and P_2^+ is also 180 degrees, interchange P_1^+ and P_2^+ so that the two resulting sd-paths consist of P_1^- joined to P_2^+, and P_2^- joined to P_1^+. See Fig. 2 for an illustration.

Next, move the solution into the Hanan grid by repeating the following steps. While there exists a vertical path not lying on the Hanan grid, pick a vertical path P of maximum length that does not lie on the grid. Let cost_L be the total cost per unit of length of all horizontal links touching P from the left, and let cost_R be the total cost per unit of length of all horizontal links touching P from the right. If $\text{cost}_L \geq \text{cost}_R$ then move P to the left (otherwise, if $\text{cost}_L < \text{cost}_R$, move P to the right) until: P is no longer a vertical path of maximal length, P overlaps with another vertical path, or P reaches the Hanan grid. Clearly, this will not increase the total cost of the solution. Finally, eliminate horizontal paths not lying on the Hanan grid in the same way. □

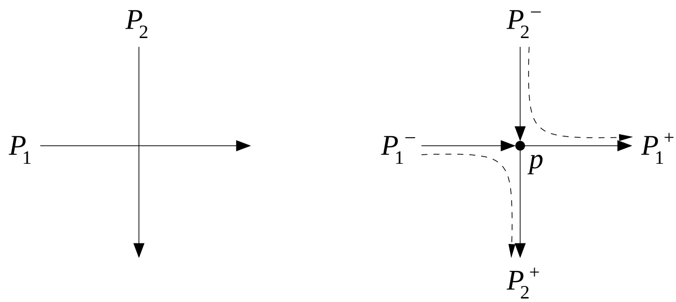

Fig. 2. In the proof of Lemma 3, every crossing between two sd-paths P_1 and P_2 of the form shown on the left is eliminated by replacing P_1 and P_2 by the two paths obtained by: (1) joining P_1^- (the part of P_1 leading to the intersection point p) to P_2^+ (the part of P_2 leading from p); and (2) joining P_2^- to P_1^+.

3. An Exact Algorithm for BRND

Here, we describe an exact, divide-and-conquer-based algorithm for BRND. It relies on Lemma 3, which guarantees the existence of an optimal solution to BRND lying on the Hanan grid. The main idea is to transform the instance into an exponential number of pairs of smaller instances which are solved recursively and then joined to find an optimal solution; the pairs of smaller instances are obtained by enumerating all valid combinations of sd-paths in the part of the grid that connects them.

Lemma 4. *Let H be a grid consisting of h horizontal lines and k vertical lines, where w.l.o.g. $h \leq k$. Consider an instance of BRND with total demand D, where each source or sink in the instance is placed in a distinct grid point of H. Suppose there is an oracle which, for any integer $q \leq D$, returns a specification of a cheapest set of links having total capacity at least q in constant time. Then an optimal solution to the BRND instance can be found in $D^{O(k)}$ time.*

Proof. The proof is by induction on k.

If $k \leq 2$ then the total number of sources and sinks is at most four. Since $h \leq k$, the grid H contains at most four edges between grid points. By Lemma 3, there is an optimal solution to the instance for which all links (i.e., copies of edges) lie on at most four edges of H. There are at most $(D+1)^4$ possible assignments of total capacities of the links on these (at most four) edges and at most 2^4 possible assignments of directions to the links. For each such pair of assignments, we can easily check whether it is compatible with a feasible solution to the instance of BRND. If yes, then for each of the edges, we query the oracle to find a cheapest set of links achieving at least the assigned capacity. Finally, we pick the cheapest among all such feasible solutions. Hence, we obtain an optimal solution in $D^{O(1)}$ time.

Next, suppose $k > 2$. Let M be the median vertical grid line in H. Denote the h positions where M intersects a horizontal line of the grid by $m_1, m_2, \ldots, m_h$ (ordered from top to bottom). Define a vector $(e_1, e_2, \ldots, e_h)$ that represents the demands of the sources

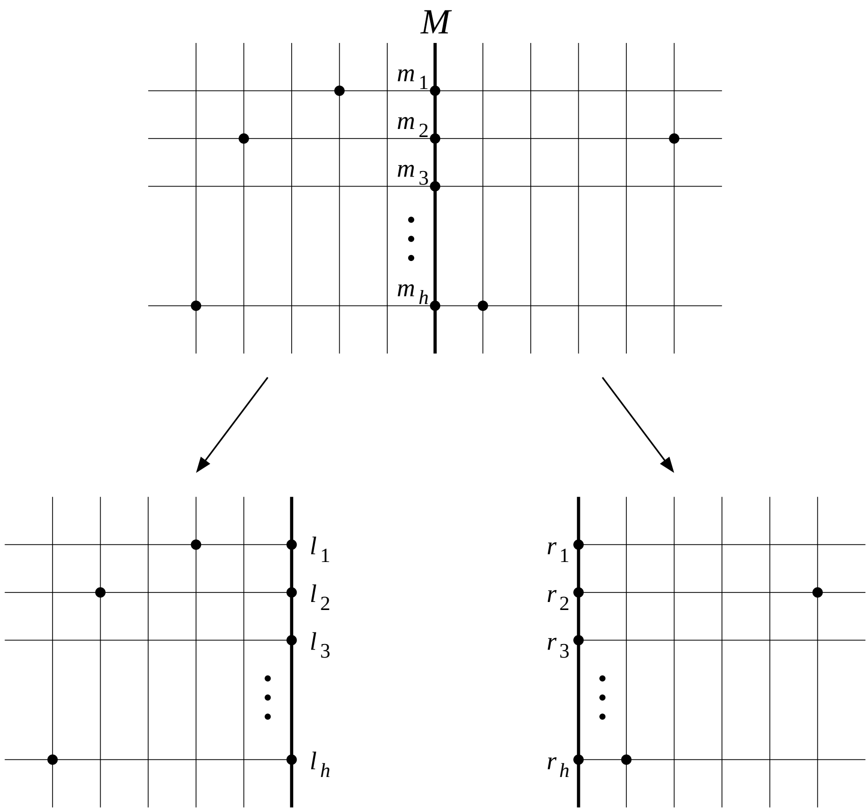

Fig. 3. Illustrating the proof of Lemma 4. The given instance of BRND is split into a *left part* and a *right part*, which are used to define pairs of smaller instances of BRND, denoted by $Left(f_1, f_2, \ldots, f_h)$ and $Right(f_1, f_2, \ldots, f_h)$, where each $f_i \in \{-D, -D+1, \ldots, -1, 0, 1, \ldots, D-1, D\}$.

and sinks lying on M as follows: For each $i \in \{1, 2, \ldots, h\}$, if m_i contains a source in the given instance of BRND then set $e_i = d(m_i)$; if m_i contains a sink then set $e_i = -d(m_i)$; otherwise (i.e., m_i contains neither a source nor a sink), set $e_i = 0$.

Split the given instance of BRND into two parts: the *left part*, consisting of all sources and sinks strictly to the left of M, and the *right part*, consisting of all sources and sinks strictly to the right of M. Augment the left part with h new points $l_1, l_2, \ldots, l_h$ at those h positions where M intersects a horizontal line of the grid, and augment the right part with new points $r_1, r_2, \ldots, r_h$ in the same way (note that each pair l_i and r_i would coincide if we merge the left and right parts). See Fig. 3. Next, for any vector $(f_1, f_2, \ldots, f_h)$, where each $f_i \in \{-D, -D+1, \ldots, -1, 0, 1, \ldots, D-1, D\}$, define two instances $Left(f_1, f_2, \ldots, f_h)$ and $Right(f_1, f_2, \ldots, f_h)$ of BRND by modifying the left and right parts as follows. For each $i \in \{1, 2, \ldots, h\}$:

- If $e_i + f_i > 0$ then let l_i be a source with demand $d(l_i) = e_i + f_i$. If $e_i + f_i < 0$ then let l_i be a sink with demand $d(l_i) = -e_i - f_i$. Otherwise (i.e., $e_i + f_i = 0$), delete point l_i.

- If $f_i > 0$ then let r_i be a sink with demand $d(r_i) = f_i$. If $f_i < 0$ then let r_i be a source with demand $d(r_i) = -f_i$. Otherwise (i.e., $f_i = 0$), delete point r_i.

Now, for any vector $(f_1, f_2, \ldots, f_h)$ such that the total demand of all sources and sinks in each of the two instances $\mathit{Left}(f_1, f_2, \ldots, f_h)$ and $\mathit{Right}(f_1, f_2, \ldots, f_h)$ does not exceed D, the two instances can be solved recursively and their solutions merged to give a candidate solution to the given instance of BRND. There are at most $(2D + 1)^h \leq (D+1)^h \cdot 2^h$ such vectors to try, and after solving all resulting pairs of instances, we select a candidate solution of minimum cost to find an optimal solution to the given instance of BRND.

In each of the instances $\mathit{Left}(f_1, f_2, \ldots, f_h)$ and $\mathit{Right}(f_1, f_2, \ldots, f_h)$, the number h of horizontal lines may exceed the number of vertical lines (which is at most $\lfloor k/2 \rfloor + 1$). Therefore, to obtain a simple recursive formula for the running time of our procedure, we further divide each of the left and right instances by a median horizontal grid line in the same way as above. Thus, the input instance is reduced to at most $(D+1)^{3k} 2^{3k}$ quadruples of BRND instances, each with at most $\lfloor k/2 \rfloor + 1$ vertical lines and at most $\lfloor k/2 \rfloor + 1$ horizontal lines. (The $3k$ in the exponent comes from the fact that three partitioning lines, each with at most k grid points, are needed to obtain such a quadruple of smaller instances). We obtain the following recursive bound on the time $T(k, D)$ required to obtain an optimal solution to the input instance:

$$
\begin{aligned}
T(D,k) &\leq D^{3k} \cdot 2^{3k+2} \cdot (T(D, \lfloor k/2 \rfloor + 1) + O(k)) \\
&\leq D^{3k} \cdot 2^{3k+2} \cdot (T(D, 2k/3) + O(k)) \\
&\leq D^{O\left(\sum_{i=0}^{\infty} k \cdot (2/3)^i\right)} \cdot 2^{O\left(\sum_{i=0}^{\infty} k \cdot (2/3)^i\right) + O(\log k)} \cdot D^{O(1)} \\
&\leq D^{O(k)}. \qquad \square
\end{aligned}
$$

Next, using the idea behind the pseudo-polynomial time algorithm for the integer knapsack problem [15] gives:

Lemma 5. *For all positive integers $q \leq D$, one can find the cheapest set of links having total capacity at least q in $O(DK)$ time, where K is the number of link types.*

Proof. We solve the problem for q in increasing order. To find an optimal solution for q, pick the cheapest solution among the solutions which can be decomposed into an optimal solution for $q - c_i$ and a single copy of the link of type i, where i ranges over all possible types of links. □

By using Lemma 5 as the oracle in Lemma 4, we obtain the following theorem.

Theorem 6. *There is an exact algorithm for BRND which runs in $D^{O(n)} \cdot K$ time, where n is the total number of sources and sinks, $D \geq 2$ is their total demand, and K is the number of link types.*

4. A QPTAS for BRND and a $(\sqrt{2}+\varepsilon)$-Approximation for BGND

In this section, we present our QPTAS for BRND and also show how to apply it to BGND. We begin with generalizations of several results from $[4, 33]$ concerning PTAS for TSP and the minimum Steiner tree in the plane.

We first state a generalization of the Perturbation Lemma from $[4, 33]$.

Lemma 7. *Let $G = (V, E)$ be a geometric multi-graph with vertices in $[0, 1]^2$, and edge costs of the form $\delta_i \cdot |e|$ for $e \in E$, where $i \in \{1, \ldots, K\}$ and $0 \le \delta_1 < \cdots < \delta_K$. Next, let $U \subseteq V$. Denote by $E(U)$ the set of edges incident to the vertices in U. One can perturb the vertices in U so that their coordinates become of the form $(\frac{i}{q}, \frac{j}{q})$, where $i,\ j$ are natural numbers not greater than a common natural denominator q, and the total cost of G increases or decreases by an additive term of at most $\sqrt{2} \cdot \delta_K \cdot |E(U)|/q$.*

Consider an instance of BGND or BRND with sources $s_1 \ldots s_{n_s}$ and sinks $t_1 \ldots t_{n_t}$. We may assume w.l.o.g. that the sources and the sinks are in $[0, 1)^2$.

Suppose that the total demand D is $n^{O(1)}$ where $n = n_s + n_t$. It follows that the maximum degree in a minimum cost multigraph solving BGND or BRND is $n^{O(1)}$. Hence, the total number of links incident to the sources and sinks in the multigraph is also $n^{O(1)} = n^{O(1)} \times n$. In the case of BRND, we conclude that the total number of links incident to all vertices, i.e., including the Steiner points, is, w.l.o.g., $n^{O(1)} = n^{O(1)} \times O(n^2)$ by Lemma 3.

Let $\Delta > 0$. Recall that δ_K denotes the maximum cost per unit of length, taken over all K types of links. By using a straightforward extension of Lemma 7 to include a geometric multigraph and rescaling the coordinates of the sources and sinks by a factor $L = \frac{n^{O(1)} \cdot \delta_K}{\Delta}$, we can alter the given BGND instance or BRND instance to a new instance so that:

- all sources and sinks of the new instance of BGND/BRND as well as all Steiner vertices of the new instance of BRND lie on the integer grid $[0, L)^2$, and
- given any solution to the original instance of BGND/BRND, it holds that for each type of link, the total length of all links of this type in the corresponding solution to the new instance is at most $L(1 + \Delta)$ times larger, and
- given any solution to the new instance of BGND/BRND, it holds that for each type of link, the total length of all links of this type in the corresponding solution to the original instance is at most $(1 + \Delta)/L$ times larger.

The second and third properties imply that we may assume that any input instance of BGND has all its sources and sinks on the integer grid in $[0, L)^2$, since this assumption introduces at most an additional $(1 + \Delta)$ factor to the final approximation ratio. We shall call it *the rounding assumption*. In the case of BRND, we may further assume not only that our input instance has its sources and sinks on the integer grid, but also that all Steiner vertices may be located only on this grid. We term this assumption *the strong rounding assumption*. By the second and third properties, the strong rounding assumption in the case of BRND also introduces (at most) an additional $(1 + \Delta)$ factor to the final approximation ratio.

Now, we pick two integers a and b uniformly at random from $[0, L)$ and extend the grid by a vertical grid lines to the left and $L - a$ vertical grid lines to the right. We similarly increase the height of the grid by using b, and denote the grid obtained this way by $L(a, b)$. Next, recursively decompose $L(a, b)$ into dissection squares by using a 4-ary tree $Q(a, b)$ called a *quadtree* so that:

- The root of $Q(a, b)$ corresponds to the square $L(a, b)$.
- Each square S of area > 1 and containing > 1 point is recursively partitioned into four equal-sized subsquares S_1, S_2, S_3, S_4 called *sibling dissection squares*. In $Q(a, b)$, the internal node representing S has four children representing S_1, S_2, S_3, S_4.
- $Q(a, b)$ has height $O(\log n)$ and $n^{O(1)}$ nodes.

We say a graph G is r-*light* if it crosses each boundary between two sibling dissection squares of $Q(a, b)$ at most r times. A multigraph H is r-*fine* if it crosses each boundary between two sibling dissection squares of $Q(a, b)$ in at most r places. For any line segment ℓ and positive integer r, the r-*portals* of ℓ are defined as the r endpoints of the $r - 1$ equal-length subsegments of ℓ into which ℓ can be partitioned. See Fig. 4.

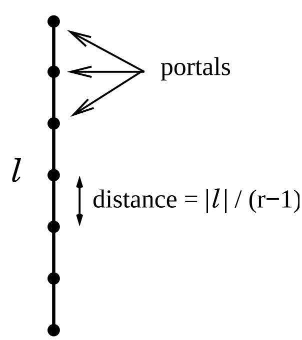

Fig. 4. A line ℓ and the r-portals of ℓ.

Next, we derive the following new theorem which can be seen as a generalization of the structure theorem of Arora [4] to include geometric multigraphs where the guarantee of r-lightness is replaced by the weaker guarantee of r-fineness.

Theorem 8. *For any $\varepsilon > 0$ and any BRND (or BGND, respectively) on the grid $L(a, b)$, there is a multigraph on $L(a, b)$ crossing each boundary between two sibling dissection squares of $Q(a, b)$ only at $O(\log L/\varepsilon)$-portals which is a feasible solution of BRND (BGND, respectively) and has expected length at most $(1 + \varepsilon)$ times larger than the minimum.*

Proof. We will prove the result for BRND. Identical arguments work for BGND.

The proof follows the approach proposed by Arora in [4]. Let $r = \frac{2(1+\log L)}{\varepsilon}$. Let G be the multigraph which is an optimal solution for BRND. We will modify the edges of G to obtain a multigraph H on $L(a,b)$ crossing each boundary between two sibling dissection squares of $Q(a,b)$ only at r-portals that is a feasible solution of BRND and that has expected length at most $(1+\varepsilon)$ times larger than that of G.

The dissection squares in $Q(a,b)$ are defined hierarchically and one can assign *levels* ranging from 0 to $\log L$ to them in the natural way: the root is at level 0, and for every $i \geq 0$, the children of a dissection square at level i are at level $i+1$. Let ℓ be any grid line in $L(a,b)$. We define the *level of* ℓ as the minimum level of all dissection squares that have one side contained in ℓ. Observe that if ℓ is of level i then any dissection square in $Q(a,b)$ with a side lying on ℓ has side length at most $L/2^i$. Also note that since the shift a and the shift b are chosen at random, for each vertical or horizontal line ℓ in the grid and for each i with $0 \leq i \leq \log L$, we have $\mathbf{Pr}[\ell \text{ is at level } i] \leq 2^i/L$.

For a geometric multigraph $\mathcal{Q}$ (which will be either G or H in our case), for $i = 1,\ldots,K$, let $\mathcal{Q}_i$ be the sub-multigraph of $\mathcal{Q}$ induced by links of type i. For any sub-multigraph $\mathcal{Q}'$ of $\mathcal{Q}$, we denote by $cost(\mathcal{Q}')$ the total cost of all its links, whereas $|\mathcal{Q}'|$ stands for the total length of all its links.

Consider a line ℓ and let level(ℓ) be its level. Each edge e crossing ℓ will be deflected in the new multigraph H to ensure that it crosses ℓ in the nearest portal to the original crossing. Since the portals are placed in ℓ at distance $\frac{L}{r\cdot 2^{\text{level}(\ell)}}$ from each other, this modification of edge e will increase the length of e in H by at most an additive term of $\frac{L}{r\cdot 2^{\text{level}(\ell)}}$. Therefore, if we denote by $\tau(\ell, G_i)$ the number of edges in G_i crossing ℓ, the total increase in the length of all edges in G_i crossing ℓ will be at most $\frac{\tau(\ell,G_i)\cdot L}{r\cdot 2^{\text{level}(\ell)}}$.

Next, we observe that level(ℓ) is a random variable that depends on the random choice of the shift in $Q(a,b)$. Therefore, the expected increase of the length of all edges in G_i crossing ℓ will be at most:

$$\begin{aligned}\sum_{i=0}^{\log L} \mathbf{Pr}[\text{level}(\ell) = i] \cdot \frac{\tau(\ell,G_i)\cdot L}{r\cdot 2^i} &\leq \sum_{i=0}^{\log L} \frac{2^i}{L}\cdot\frac{\tau(\ell,G_i)\cdot L}{r\cdot 2^i} \\ &= \frac{(1+\log L)\cdot\tau(\ell,G_i)}{r}.\end{aligned}$$

Therefore, if we sum this over all $G_0,\ldots G_K$ and horizontal and vertical grid lines ℓ, then we see that the expected total cost of the new multigraph H is upper bounded by:

$$\begin{aligned}\mathbf{E}\big[\text{cost}(H)\big] &= \mathbf{E}\big[\sum_{i=0}^{K}\text{cost}(H_i)\big] \\ &\leq \sum_{i=0}^{K}\text{cost}(G_i) + \delta_i\cdot\frac{(1+\log L)\cdot\sum_{\ell\text{: grid line in } L(a,b)}\tau(\ell,G_i)}{r} \\ &= \text{cost}(G) + \frac{1+\log L}{r}\cdot\sum_{i=0}^{K}\delta_i\cdot\sum_{\ell\text{: grid line in } L(a,b)}\tau(\ell,G_i)\end{aligned}$$

where δ_i is the cost per unit of length of a link of type i, as in Section 2.1.

Now, we use Lemma 4 from [4] which bounds (under the assumption that all edges in G_i are of length at least 4, which can be assumed w.l.o.g. by rescaling the original grid) the sum: $\sum_{\ell\text{: grid line (either horizontal or vertical) in } L(a,b)} \tau(\ell, G_i) \leq 2 \cdot |G_i|$.

Finally, we set $r = \frac{2(1+\log L)}{\varepsilon}$ to obtain:

$$\mathbf{E}\big[\text{cost}(H)\big] \leq \text{cost}(G) + \frac{1+\log L}{r} \cdot \sum_{i=0}^{K} \delta_i \cdot 2 \cdot |G_i| \;=\; (1+\varepsilon) \cdot \text{cost}(G) \;. \qquad \square$$

To obtain a QPTAS for BRND in the Euclidean plane with polynomial total demand, we employ a bottom-up dynamic programming approach to find a minimum cost multigraph for BRND on $L(a,b)$ that crosses each boundary between two sibling dissection squares of $Q(a,b)$ only at r-portals, where $r = O(\log n/\varepsilon)$. This means that we need to solve a number of subproblems, each consisting of finding a minimum cost r-fine rectilinear multigraph for BRND within the square, where the sources are the original sources within the square and the crossing points expected to supply some demand, whereas the sinks are the original sinks within the square and the crossing points expected to receive some demand. More precisely, every subproblem is specified by:

(1) A node in the quadtree $Q(a,b)$, corresponding to a dissection square.
(2) A choice of crossing points, i.e., a subset of that square's $O(r)$-portals.
(3) For each chosen crossing point p, an integral demand $d(p)$ that it should either supply or receive (instead of the pairing of the distinguished portals as in [4]).

By the upper bound $D \leq n^{O(1)}$, we may assume that $d(p) = n^{O(1)}$ for every crossing point p. Thus, the total number of different subproblem specifications must be $n^{O(r)}$. Note that using a smaller ε implies a better approximation ratio but more subproblems.

We first solve all leaf subproblems, i.e., subproblems represented by leaves in the quadtree $Q(a,b)$, and then all non-leaf subproblem, i.e., subproblems represented by internal nodes in $Q(a,b)$, in bottom-up order as follows:

- Each leaf subproblem, where the dissection square is a cell of $L(a,b)$ and the original sources and sinks may be placed only at the corners of the dissection square and the remaining $O(r)$ ones at r-portals along the boundary, is solved in $n^{O(r)}$ time according to Theorem 6.
- Each non-leaf subproblem P is solved by combining optimal solutions to quadruples of compatible subproblems P_1, P_2, P_3, P_4 corresponding to the four dissection squares that are children of the dissection square for P, and selecting the best such quadruple. (Here, P_1, P_2, P_3, P_4 are *compatible* if the locations of the crossing points and their specified demands are consistent among P_1, P_2, P_3, P_4.) There are $n^{O(r)} \cdot n^{O(r)} \cdot n^{O(r)} \cdot n^{O(r)} = n^{O(r)}$ quadruples of subproblems of P to consider, and to check if any quadruple is compatible takes $n^{O(1)}$ time, so solving a single subproblem P takes $n^{O(r)} \cdot n^{O(1)} = n^{O(r)}$ time.

This gives:

Lemma 9. *A feasible r-fine multigraph for BRND on $L(a,b)$ with polynomially bounded total demand and total cost within $1+\varepsilon$ from the optimum is computable in $n^{O(r)}$ time.*

By combining Theorem 8 with Lemma 9 for $r = O(\frac{\log n}{\varepsilon})$ and the fact that the rounding assumption introduces only an additional $(1+O(\varepsilon))$ factor to the approximation ratio, we obtain our first result.

Theorem 10. *For any $\varepsilon > 0$, there is a randomized $n^{O(\log n/\varepsilon)}$-time algorithm for BRND in the Euclidean plane, with a total of n sources and sinks and total demand polynomial in n, that yields a solution whose expected cost is within $(1+\varepsilon)$ of the optimum.*

Theorem 10 directly implies the following result for BGND.

Corollary 11. *For any $\varepsilon > 0$, there is a randomized $n^{O(\log n/\varepsilon)}$-time algorithm for BGND in the Euclidean plane, with a total of n sources and sinks and total demand polynomial in n, that yields a solution whose expected cost is within $(\sqrt{2}+\varepsilon)$ of the optimum.*

5. A Fast, Low-Constant Approximation Scheme for BSRND and BSGND

In this section, we present another method for BRND and BGND with single sinks (BSRND and BSGND), based on a new "belt decomposition technique". The method runs in polynomial time, gives a low-constant approximation guarantee, and does not require a polynomial bound on the total demand. It is especially efficient if the link capacities are small.

We start with two useful lemmas. First, by taking the shortest path from each source to a sink and considering the case when the most economical cost per capacity over all link types could always be employed, we obtain the next lemma, analogous to the so-called routing lower bound in $[28, 34]$:

Lemma 12. *Let S be the set of sources in an instance of BRND (BGND), and for each $s \in S$, let $t(s)$ be the closest sink in this instance. The cost of an optimal solution to BRND (BGND, respectively) is at least $\sum_{s \in S} dist(s,t(s)) \cdot \frac{\delta_K}{c_K} \cdot d(s)$, where $dist(s,t(s))$ is the L_1 distance (the Euclidean distance, respectively).*

Next, we adapt the proof of Lemma 2 on p. 184 in $[27]$ for upper bounding the length of a closed walk through a set of points in a d-dimensional cube to the problem of computing a minimum Steiner tree in the plane to obtain:

Lemma 13. *Let S be a set of k points within a square of side length ℓ. One can find a Steiner tree of S with length $O(\ell\sqrt{k})$ in $O(k)$ time.*

Proof. Divide the square into ℓ/w vertical strips of width w. There is a trivial Steiner tree of S including the vertical and the top boundaries of the strips whose total length is at most $(\ell/w+2)\ell + kw/2$. See Fig. 5.

By setting w to $\ell/\sqrt{k}$, we obtain the lemma. □

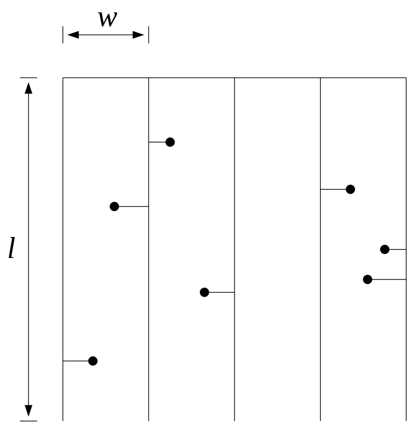

Fig. 5. The trivial Steiner tree.

The following key lemma describes a procedure which yields an *almost* feasible solution to BSRND or BSGND with cost arbitrarily close to the optimum.

Lemma 14. *For any $\varepsilon > 0$, there is a reduction procedure for BSRND (or BSGND, respectively) with one sink, $n-1$ sources, and the ratio between the maximum and minimum distances of a source from the sink equal to m, which returns a multigraph yielding a partial solution to the given instance of BSRND (or BSGND, respectively) satisfying the following conditions:*

- *All but $O((\frac{1}{\varepsilon})^2 c_K^2 \log m)$ sources can ship their whole demand to the sink;*
- *For each source s there are at most $c_K - 1$ units of its whole demand $d(s)$ which cannot be shipped to the sink.*

The procedure runs in $O(\frac{c_K}{\varepsilon^2} \log m \log n + c_K n)$ time, which is $O(n/\varepsilon^2)$ if $c_K = O(1)$.

Proof. Form a rectilinear $2\lceil m \rceil \times 2\lceil m \rceil$ grid F with unit distance equal to the minimum distance between the only sink t and a source, centered around t. Let μ be a positive constant whose value will be set later.

Partition F into a square R with side length $2\lceil \mu\sqrt{c_K} \rceil$ centered in t, and for $i \in \{0, 1, \ldots, \log m\}$, the *belts* B_i of squares of side length 2^i within L_∞ distance at least $2^i \lceil \mu\sqrt{c_K} \rceil$ and at most $2^{i+1} \lceil \mu\sqrt{c_K} \rceil$ from t. See Fig. 6 for an illustration. Note that the number of squares in any belt B_i is at most $(4\lceil \mu\sqrt{c_K} \rceil)^2 = O(\mu^2 c_K)$ by the definition of the grid; hence, the total number of squares in all the belts is $O(\mu^2 c_K \log m)$.

The reduction procedure consists of two phases, described below. Importantly, after the first phase, the remaining demand for each source is at most $c_K - 1$ units. After the second phase, the total remaining demand of *all* sources inside each square is at most $c_K - 1$ units.

Phase 1: Connect each source s by a multi-path composed of $\lfloor d(s)/c_K \rfloor$ copies of a shortest path from s to t, implemented with the K-th type of links. Observe that the average cost

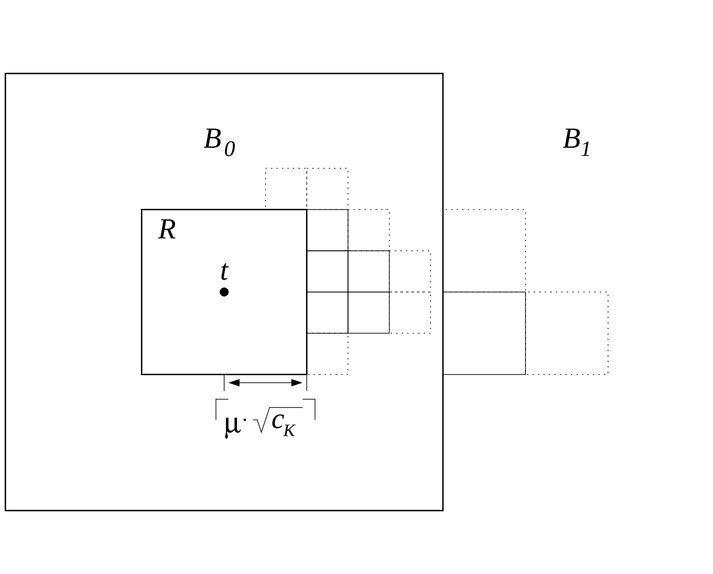

Fig. 6. An example of the decomposition into the square R and the belts B_i. The side length of a square in belt B_{i+1} is exactly twice that of a square in B_i for $i \geq 0$, and the number of squares in each belt is $O(\mu^2 c_K)$.

of such a connection per each of the $c_K \lfloor d(s)/c_K \rfloor$ demand units u shipped from s to t is $dist(s,t) \cdot \frac{\delta_K}{c_K}$, which is optimal by Lemma 12.

Phase 2: For each square Q in each of the belts B_i, locate all sources contained in Q. Then, while the total remaining demand of all sources in Q is at least c_K, arbitrarily select c_K units of remaining demand, find a minimum Steiner multitree of their sources and connect its vertex v closest to t by a shortest path to t. The total length of the resulting multitree is $dist(v,t) + O(2^i \sqrt{c_K}) \leq (1 + O(\frac{1}{\mu})) \cdot dist(v,t)$ by the definition of the squares and Lemma 13. Hence, for each demand unit u in the c_K-tuple originating from its source $s(u)$, we can assign the average cost of connection to t by the multitree implemented with the K-th type of links not greater than $(1+O(\frac{1}{\mu}))\cdot dist(s(u),t)\cdot \frac{\delta_K}{c_K}$. It follows from Lemma 12 that the total cost of the constructed network is within $(1 + O(\frac{1}{\mu}))$ from the minimum cost of a multigraph for the input BSRND. By choosing μ appropriately large, we obtain the required $(1 + \varepsilon)$-approximation.

After the second phase, each square outside R contains at most $c_K - 1$ sources with non-zero remaining demand. Since the number of squares is $O(\mu^2 c_K \log m)$, the total number of their sources with a non-zero remaining demand (i.e., at most $c_K - 1$ units) to ship is $O(\mu^2 c_K^2 \log m)$ after the second phase. Furthermore, since the square R contains at most $O(\mu^2 c_K)$ sources, the number of sources in R with a non-zero remaining demand is only $O(\mu^2 c_K)$.

Finally, we analyze the running time. The first phase can be implemented in time linear in the number of sources. The second phase first needs to find all sources located inside each square Q. This can be done by using a standard 2D range query data structure for reporting all points in a query rectangle [32] with the query rectangle set to Q; in total, $O(\mu^2 c_K \log m)$ 2D range queries for (disjoint) squares are required, which takes $O(\mu^2 c_K \log m \log n)$ time by [32]. Next, the second phase constructs $O(c_K n/c_K) = O(n)$ Steiner trees on at most c_K vertices each, using the method of Lemma 13 above, which takes $O(c_K n)$ time. W.l.o.g., we let $\mu = O(\frac{1}{\varepsilon})$ and conclude that the whole procedure takes $O(\frac{c_K}{\varepsilon^2} \log m \log n + c_K n)$ time. □

Now, we are ready to derive our main result in this section. Basically, we can solve any given instance of BSRND approximately by first running the reduction procedure of Lemma 14, and then applying the more sophisticated approximation algorithm from Theorem 10 to handle the leftover demands.

Theorem 15. *For any $\varepsilon > 0$, there is a $(2 + \varepsilon)$-approximation algorithm for BSRND with one sink and $n - 1$ sources in the Euclidean plane, running in $O((\frac{1}{\varepsilon})^2 c_K \log^2 n + c_K n) + (c_K^2 \log n)^{O(\frac{\log c_K}{\varepsilon^2})}$ time; in particular, in $n(\log n)^{O(1)} + (\log n)^{O(\frac{\log \log n}{\varepsilon^2})}$ time if $c_K = (\log n)^{O(1)}$.*

Proof. By the rounding assumption discussed in Section 4, we can perturb the sinks and the sources so they lie on an integer grid of polynomial size introducing only an additional $(1+O(\varepsilon))$ factor to the final approximation ratio. The perturbation can easily be done in linear time. Next, we apply the reduction procedure from Lemma 14 to obtain an almost feasible solution of total cost not exceeding $(1+O(\varepsilon))$ of that for the optimal solution to the BSRND on the grid. Note that $m \le n^{O(1)}$ and hence $\log m = O(\log n)$ in this application of the reduction by the polynomiality of the grid. It remains to solve the BRND subproblem for the $O((\frac{1}{\varepsilon})^2 c_K^2 \log n)$ remaining sources with total remaining demand polynomial in their number. This subproblem can be solved with the randomized $(1 + O(\varepsilon))$-approximation algorithm of Theorem 10. We may also use its derandomized version here, which will run in $(c_K^2 \log n)^{O(\frac{\log c_K}{\varepsilon^2})}$ time. □

As an immediate corollary to Theorem 15, we obtain:

Corollary 16. *For any $\varepsilon > 0$, there is a $(\sqrt{8} + \varepsilon)$-approximation algorithm for BSGND with one sink and $n - 1$ sources in the Euclidean plane, running in $O((\frac{1}{\varepsilon})^2 c_K \log^2 n + c_K n) + (c_K^2 \log n)^{O(\frac{\log c_K}{\varepsilon^2})}$ time.*

Remark 1. For the special case of $c_K = O(1)$, one can obtain a simpler (and still polynomial-time) approximation algorithm for BSRND by plugging in Theorem 6 instead of Theorem 10 to handle the leftover demands in the proof of Theorem 15. The reason is that when $c_K = O(1)$, the number of sources with a non-zero remaining demand is $O((\frac{1}{\varepsilon})^2 c_K^2 \log m) = O((\frac{1}{\varepsilon})^2 \log m) = O((\frac{1}{\varepsilon})^2 \log n)$, i.e., logarithmic in n.

Remark 2. Our new technique in this section also extends to a *relaxed* variant of BRND in which the demands of the sinks are unspecified, meaning that every sink can accept any amount of supply from the sources. We can generalize the reduction of Lemma 14 to include n_t sinks by first computing the Voronoi diagram of the sinks in the L_1-metric on the grid, then associating each source to its closest sink by checking which region of the Voronoi diagram it belongs to, and finally running the reduction procedure of Lemma 14 separately on each set of sources contained in a single Voronoi region. The construction of the Voronoi diagram and the location of the sources take $O(n \log n)$ time (see $[25, 32]$). It follows by straightforward calculations that for any $\varepsilon > 0$, there is a $(2+\varepsilon)$-approximation algorithm for the relaxed BRND with n_t sinks and $n - n_t$ sources in the Euclidean plane, running in $O((\frac{1}{\varepsilon})^2 n_t c_K \log^2 n + n(\log n + c_K)) + (n_t c_K^2 \log n)^{O(\frac{\log n_t + \log c_K}{\varepsilon^2})}$ time. This also implies a $(\sqrt{8}+\varepsilon)$-approximation algorithm for the analogously relaxed BGND with n_t sinks and $n - n_t$ sources in the Euclidean plane, running in time $O((\frac{1}{\varepsilon})^2 n_t c_K \log^2 n + n(\log n + c_K)) + (n_t c_K^2 \log n)^{O(\frac{\log n_t + \log c_K}{\varepsilon^2})}$.

6. Conclusion

In this paper, we have shown:

- BGND, BRND, BSGND, and BSRND are NP-hard, even when $K = 1$ and D is polynomially bounded in n (Lemma 2 in Section 2.2).
- An exact algorithm for divisible BRND that runs in $D^{O(n)} \cdot K$ time (Theorem 6 in Section 3).
- A QPTAS for divisible BRND in the Euclidean plane with $D \leq n^{O(1)}$ (Theorem 10 in Section 4).
- A $(2+\varepsilon)$-approximation algorithm for divisible BSRND in the Euclidean plane that runs in $n(\log n)^{O(1)} + (\log n)^{O(\frac{\log\log n}{\varepsilon^2})}$ time if $c_K = (\log n)^{O(1)}$ (Theorem 15 in Section 5).
- A $(\sqrt{8}+\varepsilon)$-approximation algorithm for divisible BSGND in the Euclidean plane that runs in $O((\frac{1}{\varepsilon})^2 c_K \log^2 n + n(\log n + c_K)) + (c_K^2 \log n)^{O(\frac{\log c_K}{\varepsilon^2})}$ time (Corollary 16 in Section 5).

To summarize, our results suggest that certain geometric variants of minimum total edge cost problems are easier to closely approximate than their graph counterparts. For example, the best known approximation ratios achievable in polynomial time for divisible buy-at-bulk network design in the graph model are polylogarithmic in the multi-sink case [7] and 20.41 in the single-sink case [20], while we have obtained an $(\sqrt{8}+\varepsilon)$-approximation algorithm for BSGND in the Euclidean plane whose running time is polynomial when c_K (the maximum capacity of a link) is constant, according to Corollary 16.

All our approximation results for different variants of BRND in the Euclidean plane derived in this paper can be generalized to include the corresponding variants of BRND in a Euclidean space of fixed dimension. Our approximation schemes are randomized but they can be derandomized in the same way as those in $[4, 12, 13, 33]$. An open problem is how to design faster approximation algorithms for the problems considered in this paper desirably under less restrictive assumptions on the number of sinks, the capacities, and the total demand. It is worth mentioning that recently a polynomial-time approximation scheme for the restriction of BSGND in the Euclidean plane allowing for only one type of link with not too large capacity and polynomial total demand has been presented in [1].

Acknowledgment

We thank Anna Adamaszek for pointing out a flaw in Subsection 3.2 of the preliminary conference version of this paper; the subsection has been omitted in this final version.

References

[1] A. Adamaszek, A. Czumaj, A. Lingas, and J.O. Wojtaszczyk. Approximation Schemes for Capacitated Geometric Network Design. *Proc. 38th ICALP*, pp. 25–36, 2012.

[2] M. Andrews. Hardness of buy-at-bulk network design. *Proc. 45th IEEE FOCS*, pp. 115–124, 2004.

[3] S. Antonakopoulos, C. Chekuri, B. Shepherd, and L. Zhang. Buy-at-bulk network design with protection. *Proc. 48th IEEE FOCS*, pp. 634–644, 2007.

[4] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, September 1998.

[5] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-medians and related problems. *Proc. 30th ACM STOC*, pp. 106–113, 1998.

[6] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag, Berlin, 1999.

[7] B. Awerbuch and Y. Azar. Buy-at-bulk network design. *Proc. 38th IEEE FOCS*, pp. 542–547, 1997.

[8] D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming*, 81(2):177–199, 1998.

[9] M. Charikar and A. Karagiozova. On non-uniform multicommodity buy-at-bulk network design. *Proc. 37th ACM STOC*, pp. 176–182, 2005.

[10] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximation algorithms for nonuniform buy-at-bulk network design. *SIAM Journal on Computing*, 39(5):1772–1798, 2010.

[11] S. Chopra, I. Gilboa, and S. T. Sastry. Source sink flows with capacity installation in batches. *Discrete Applied Mathematics*, 85(3):165–192, 1998.

[12] A. Czumaj and A. Lingas. On approximability of the minimum-cost k-connected spanning subgraph problem. *Proc. 10th ACM-SIAM SODA*, pp. 281–290, 1999.

[13] A. Czumaj and A. Lingas. Fast approximation schemes for Euclidean multi-connectivity problems. *Proc. 27th ICALP*, LNCS Vol. 1853, pp. 856–868, 2000.

[14] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Connected facility location via random facility sampling and core detouring. *Journal of Computer and System Sciences*, 76(8): 709–726, 2010.

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

[16] N. Garg., R. Khandekar, G. Konjevod, R. Ravi, F. S. Salman, and A. Sinha. On the integrality gap of a natural formulation of the single-sink buy-at-bulk network design problem. *Proc. 8th IPCO*, LNCS Vol. 2081, pp. 170–184, 2001.

[17] A. Goel and D. Estrin. Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk. *Proc. 14th ACM-SIAM SODA*, pp. 499–505, 2003.

[18] A. Goel and I. Post. One tree suffices: A simultaneous $O(1)$-approximation for single-sink buy-at-bulk. *Proc. 51st IEEE FOCS*, pp. 593–600, 2010.

[19] F. Grandoni and G. F. Italiano. Improved approximation for single-sink buy-at-bulk. *Proc. 17th ISAAC*, LNCS Vol. 4288, pp. 111–120, 2006.

[20] F. Grandoni and T. Rothvoß. Network design via core detouring for problems without a core. *Proc. 37th ICALP*, LNCS Vol. 6198, pp. 490–502, 2010.

[21] S. Guha, A. Meyerson, and K. Munagala. A constant factor approximation for the single sink edge installation problem. *Proc. 33rd ACM STOC*, pp. 383–399, 2001.

[22] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. *Proc. 35th ACM STOC*, pp. 365–372, 2003.

[23] M. Hanan. On Steiner's problem with rectilinear distance. *SIAM Journal on Applied Mathematics*, 14(2):255–265, 1966.

[24] R. Hassin, R. Ravi, and F. S. Salman. Approximation algorithms for a capacitated network design problem. *Proc. 3rd APPROX*, LNCS Vol. 1913, pp. 167–176, 2000.

[25] F. K. Hwang. An $O(n \log n)$ algorithm for rectilinear minimal spanning trees. *Journal of the ACM*, 26(2):177–182, 1979.

[26] R. Jothi and B. Raghavachari. Improved approximation algorithms for the single-sink buy-at-bulk network design problems. *Proc. 9th SWAT*, LNCS Vol. 3111, pp. 336–348, 2004.

[27] R. M. Karp and J. M. Steele. Probabilistic analysis of heuristics. Chapter 6 in *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, pp. 181–205 (edited by E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys). John Wiley & Sons Ltd., 1985.

[28] Y. Mansour and D. Peleg. An approximation algorithm for minimum-cost network design. Technical report, The Weizmann Institute of Science, Rehovot, Israel, CS94-22, 1994.

[29] A. Meyerson, K. Munagala, and S. Plotkin. Cost-distance: Two metric network design. *Proc. 41st IEEE FOCS*, pp. 624–630, 2000.

[30] M. Minoux. Discrete cost multicommodity network optimization problems and exact solution methods. *Annals of Operations Research*, 106:19–46, 2001.

[31] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999.

[32] F. Preparata and M. Shamos. *Computational Geometry*. Springer Verlag, New York, 1985.

[33] S. B. Rao and W. D. Smith. Approximating geometrical graphs via "spanners" and "banyans". *Proc. ACM STOC*, pp. 540–550, 1998. Full version appeared as TR, NEC, 1998.

[34] F. S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian. Approximating the single-sink link-installation problem in network design. *SIAM Journal on Optimization*, 11(3):595–610, 2000.

[35] K. Talwar. The single-sink buy-at-bulk LP has constant integrality gap. *Proc. 9th IPCO*, LNCS Vol. 2337, pp. 475–486, 2002.

[36] D. P. Williamson and A. van Zuylen. A simpler and better derandomization of an approximation algorithm for single source rent-or-buy. *Operations Research Letters*, 35(6):707–712, 2007.

[37] M. Zachariasen. A catalog of Hanan grid problems. *Networks*, 38(2):76–83, 2001.