

Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 01-048

A PTAS for the Grade of Service Steiner Minimum Tree Problem

Joon-mo Kim and Mihaela Cardei

December 26, 2001

A PTAS for the Grade of Service Steiner Minimum Tree Problem

J. Kim* M. Cardei*

Dec. 26, 2001

Abstract

In this paper, we present the design of a PTAS(Polynomial Time Approximation Scheme) for the Grade of Service Steiner Minimum Tree (GOSST) problem, which is known to be NP-Complete. The previous research has focused on geometric analyses and different approximation algorithms were proposed. But having the PTAS, which provides a near-optimal solution, would be the conclusion for an optimization problem. This problem has some important applications. In the Network Design, a fundamental issue for the physical construction of a network structure is the interconnection of many communication sites with the best choice of the connecting lines and the best allocation of the transmission capacities over these lines. Good solutions always provide paths with enough communication capacities between any two sites, with the least network construction costs. GOSST problem also has applications in transportation, for road constructions and some more potential uses in CAD in terms of interconnecting the elements on a plane such that to provide enough flux between any two elements.

1 Introduction

The Grade of Service Steiner Minimum Tree (GOSST) problem is a variation of the Euclidean Steiner Minimum Tree(ESMT)[9, 10], which is a problem of finding a minimum cost network that interconnects a set of given points in the Euclidean plane. Related

*Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA. E-mail: {jkim,mihaela}@cs.umn.edu

works for ESMT problem are found in [6, 7, 11, 12, 5, 13, 8, 10, 14, 15, 16, 17, 18, 19]. The reference, [4], shows its history and some major applications.

GOSST Problem Definition [4]: Let $P = \{p_1, p_2, \dots, p_n\}$ be a set of n terminal points in the Euclidean plane, where point p_i has a service request of grade $g(p_i) \in \{1, 2, \dots, n\}$. Let $0 < c(1) < c(2) < \dots < c(n)$ be n real numbers. The **Grade of Service Steiner Minimum Tree (GOSST)** problem asks for a minimum cost network interconnecting point set P and some Steiner points with service request of grade 0 such that (1) between each pair of terminal points p_i and p_j there is a path whose minimum grade of service is at least as large as $\min(g(p_i), g(p_j))$ and (2) the cost of the network is minimum among all interconnecting networks satisfying (1), where the cost of an edge with service of grade g is the product of the Euclidean length of the edge with $c(g)$. The *GOSST* problem is a generalization of the ESMT problem where all terminal points have the same grade of service request.

For the GOSST problem, the minimum cost interconnecting network is determined by the concurrent combinations of the two factors: *i*) the service of grade g for every edge, and *ii*) the choice of the Steiner points. That is, *i*) could not be determined before *ii*) is done, while the reverse is also true. Therefore, studies have done mostly for the simplified special cases of the GOSST problem: the cases that the grades of service request is either 2 or 3, or limited by some conditions. The general case approximation algorithm by Michandani [23] gives the performance ratio $r\rho + 1$, where ρ is the best performance ratio of a Steiner tree heuristic and r is the number of different grades of service request.

According to that property, the problem fundamentally requires a dynamic programming to reach the general case solutions: however, having a classic dynamic programming is impossible for it. In our approach, we give some adjustments and use *rectangular partitions* [1, 2] to the problem instance to facilitate the running of the dynamic programming with less geometric studies. Fortunately, the GOSST problem fits well to the PTAS[1, 2], so we may proceed straightforward to the $(1 + \epsilon)$ -approximation: the objective value that should be minimized is simply the total sum of the costs defined in the problem, differently from our related work [3].

The rest of this paper is organized as follow. Section 2 presents some definitions and the Structure Theorem. Then the technique to adjust the problem instance to the grids is shown in section 3. Section 4 continues with the PTAS and the dynamic programming algorithm analysis. Section 5 contains the proof for the Structure Theorem and section 6 concludes the paper.

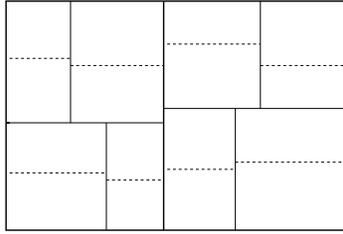


Figure 1: 1/3 : 2/3-tiling

2 Definitions and the Structure Theorem

Most of the names and definitions are from [1, 2] with some modifications for GOSST. The optimal structure is the optimal network that the GOSST problem asks. We mean a rectangle by an axis-aligned rectangle. The size of a rectangle is the length of its long side. The bounding-box of a set of terminal points is the smallest rectangle enclosing them.

The cost of an interconnecting network is the sum of the products of the Euclidean edge length with the cost-per-unit-length (CPUL) of that edge. The CPUL for an edge with service grade g is $c(g)$. The minimum CPUL is $c(1)$ and the maximum is $c(n)$ ($= C \cdot c(1)$, C is a constant).

A line-separator of a rectangle R is a straight line segment parallel with R 's shorter edge that partitions R into two rectangles, each having area at least 1/3rd of R 's area. For example, if R 's width W is greater than the height, then a line-separator is any vertical line in the middle $W/3$ of R . Next we define a recursive partition of a rectangle, over which the dynamic program runs.

Definition 1 (*1/3 : 2/3-tiling*) A 1/3 : 2/3-tiling of a rectangle R is a binary tree (a hierarchy) of sub-rectangles of R . The rectangle R is at the root. If the size of R is ≤ 1 , then the hierarchy contains nothing else. Otherwise the root contains a line-separator for R , and has two subtrees that are 1/3 : 2/3-tilings of the two rectangles, into which the line-separator divides R . (See Figure 1).

The rectangles obtained in the tiling procedure at depth d form a partition of the root rectangle. Also, all rectangles at depth $d + 1$ are a refinement of the depth d partition, obtained by applying a line-separator to each depth d rectangle of size > 1 . The area of any depth d rectangle is at most $(2/3)^d$ times the total area, implying the following proposition.

Proposition 1 *If a rectangle has width W and height H , then its every $1/3 : 2/3$ tiling has depth at most $\log_{1.5}W + \log_{1.5}H + 2$*

Definition 2 (*portals*) *A portal in a $1/3 : 2/3$ -tiling is any point that lies on the edges of rectangles in the tiling. If m is any positive integer then a set of portals P is called m -regular for the tiling if there are exactly m equidistant portals on the line-separator of each rectangle of the tiling. (We assume that the end-points of the line-separator are also portals. In other words the line-separator is partitioned into exactly $m - 1$ equal parts by the portals on it)*

Definition 3 (*m -light structure*) *Let $m \in \mathbb{Z}^+$ and π be an interconnecting network instance for the GOSST problem that satisfies the conditions in the problem definition. Let S be a $1/3 : 2/3$ -tiling of the bounding_box and P be an m -regular set of portals on this tiling. Then π is a m -light structure with respect to S if the following are true : (i) in each rectangle of tiling S , the edges cross the line-separator of that rectangle at most m times (ii) the edges cross the line-separator only at portals in P . (See Figure 2).*

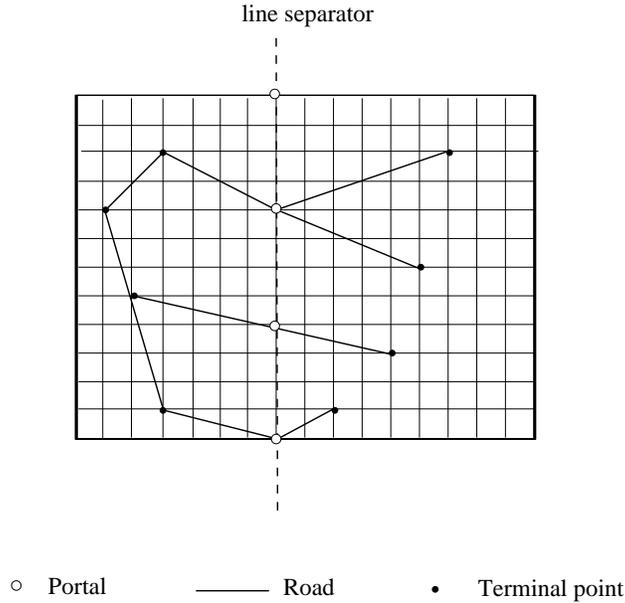
Theorem 1 (*Structure Theorem*) *The following is true for each $\epsilon > 0$. Every set of points in the problem has a $(1 + \epsilon)$ -approximate m -light structure where $m = O(\log n/\epsilon)$.*

In this paper we use the following abbreviations: *APX* for Approximation and *OPT* for Optimal.

3 Adjusting into Grids

In order to run the program, the instance of this problem should be adjusted into grids with integer coordinates. In fact, the adjustment will move each terminal point into the nearest grid point. In addition, we assume that the steiner points lie only at the grid points. We now show that the adjustment and assumption are acceptable and will study the problem instance over the grid.(See Figure 3).

Proposition 2 *$(1+\epsilon)$ -approximation over the Grid instance implies $(1+\hat{\epsilon})$ -approximation over the Original instance.*



Roads are passing through only at Portals

Figure 2: m-light graph

Proof: By the adjustment and assumption,

$$|total_cost_{OPT}^{Original} - total_cost_{OPT}^{Grid}| \leq 2 \cdot (2n - 3) \cdot c(n)$$

because the number of edges of this tree structure is $2n - 3$. Note the maximum number of points for the problem instance is $2n - 2$ that is the sum of terminal points and steiner points. Both end points of an edge can move within the distance 1 each with the highest service request of grade, $c(n)$.

$$|total_cost_{APX}^{Original} - total_cost_{APX}^{Grid}| \leq 2 \cdot n^2 \cdot c(n)$$

where n^2 is the worst case number of edges when the approximation graph is a complete network.

This condition :

$$total_cost_{APX}^{Grid} \leq (1 + \epsilon) \cdot total_cost_{OPT}^{Grid}$$

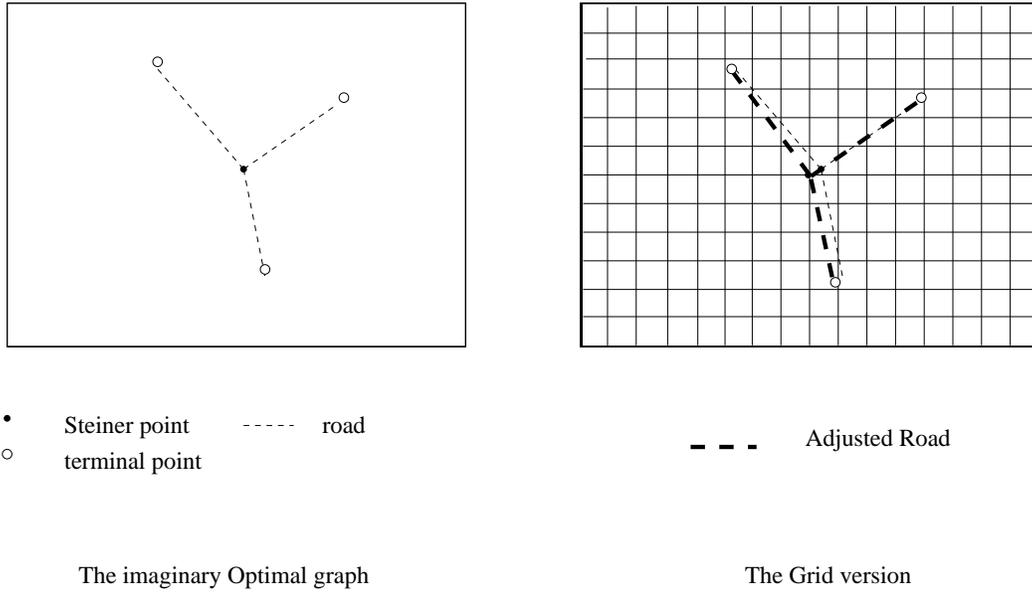


Figure 3: Difference between the Optimal and Grid graph

will be shown to be satisfied by the PTAS and then :

$$\begin{aligned}
 total_cost_{APX}^{Original} - 2n^2c(n) &\leq (1 + \epsilon) \cdot (total_cost_{OPT}^{Original} + 2(2n - 3)c(n)) \\
 total_cost_{APX}^{Original} &\leq (1 + \epsilon) \cdot total_cost_{OPT}^{Original} + 2(1 + \epsilon)(2n - 3)c(n) + 2n^2c(n) \\
 &= (1 + \epsilon + \frac{(1+\epsilon)2(2n-3)c(n)+2n^2c(n)}{total_cost_{OPT}^{Original}}) \cdot total_cost_{OPT}^{Original} \\
 &= (1 + \epsilon + \frac{(1+\epsilon)2(2n-3)c(n)+2n^2c(n)}{n^3}) \cdot total_cost_{OPT}^{Original} \\
 &\leq (1 + \hat{\epsilon}) \cdot total_cost_{OPT}^{Original}
 \end{aligned}$$

where $\hat{\epsilon}$ can be chosen accordingly. ■

The role of the term n^3 is the key of this proof and it is acquired by choosing the unit length of the grid short enough. We may set this value bigger than n^3 with a shorter unit length.

4 PTAS for GOSST problem using the Polynomial Time Dynamic Programming(DP)

By Proposition 2 and assuming that the Structure Theorem is true, we can build the *m-light structure* up to the root of the tiling with the DP. The Structure Theorem

guarantees the existence of $(1 + \epsilon)$ -approximate *m-light structure* and its tilings S where $m = O(\frac{\log n}{\epsilon})$. By Proposition 1, the depth of any such tiling is at most $O(\log n)$. We describe next the DP that finds both S and the *m-light structure* in $\text{poly}(n) \cdot 2^{O(m)} = n^{O(1/\epsilon)}$ time. The analysis is presented later in this section.

The DP has a bottom-up approach, but it is easy to view the procedure from the final stage to the start, i.e., top-down way. The final result of the DP is a rectangle, which is the *bounding_box* for the given point set, and which contains the *m-light structure* that connects all the points with the minimum *cost*.

Right before the final rectangle is formed, many combinatorial cases must be checked. All the *line-separators* that could divide the final rectangle into two sub-rectangles according to the $1/3 : 2/3$ -tiling are considered one by one. Along such a *line-separator*, there are (1)choices of portals and each of them has the (2)choices of the grades of service request. Note that the grade of service request of a portal becomes that of the edge crossing it. (See Figure 4)

The combination of two choices produces many cases and each case has its own minimum cost *m-light structure*. Such a minimum cost *m-light structure* is the concatenation of the two smaller *m-light structures* from each of the sub-rectangles. That is, a rectangle has such minimum cost *m-light structures* for every case of the combinations.

The same observations hold repeatedly in each sub-rectangle for finding its *m-light structures* until the procedure reaches the bottom most rectangles. These bottom most rectangles have limited number of points, so the brute-force algorithm could be applied to get the minimum cost *m-light structures* for each of the combinatorial cases in the smallest rectangle.

However, note that the minimum cost *m-light structures* from the DP does not always satisfy the first condition of GOSST, i.e., (1) between each pair of terminal points p_i and p_j there is a path whose minimum grade of service is at least as large as $\min(g(p_i), g(p_j))$. Therefore, the DP should be followed by a procedure that checks if the minimum cost *m-light structure* in the *bounding_box* satisfies this condition, otherwise the checking goes on for the next minimum cost. The procedure goes on with the costs in the increasing order until the satisfying *m-light structure* is reached.

Now, we need to show that the number of entries in the lookup table for the DP is polynomial and that the run time to compute each entry is also polynomial. Because the DP should keep and reuse the minimum cost of each combinatorial case, an entry in the table is indexed by the triple : (a) a rectangle. (b) a multiset of k_1 ($\leq 4m$) portals along the perimeter of the rectangle. (c) a grade of service request.

For (a), the number of distinct rectangles is at most $\binom{n}{4}$ since the number of points

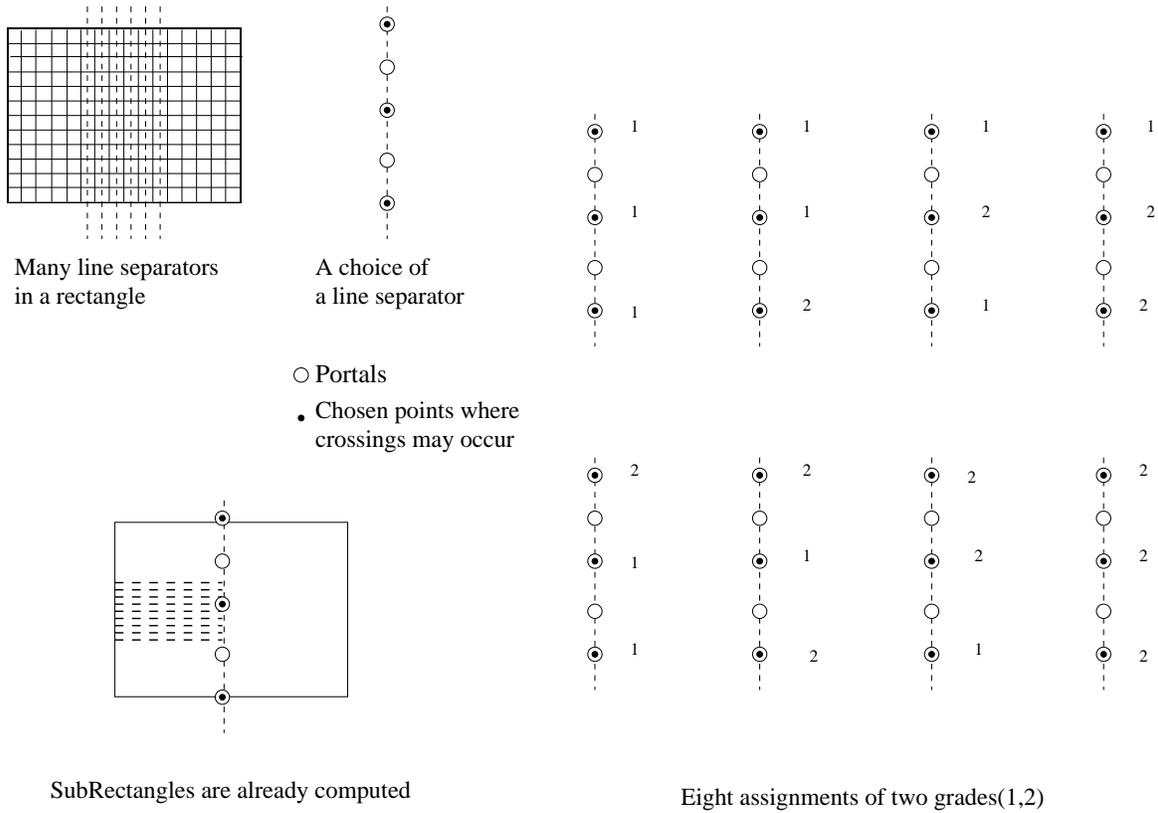


Figure 4: Combinational cases

is n and four points may make up a rectangle. For (b), each rectangle has four sides, and is part of the *line-separator* of some ancestor. The m portals on the *line-separator* are evenly spaced, so they are determined once we know the *line-separator*. But the number of choices of a *line-separator* is at most the number of pairs of points, which is $\binom{n}{2}$. This accounts for the factor $O((n^2)^4) = O(n^8)$. Once we have identified the set of $\leq 4m$ portals on the four sides, the number of ways to choose a multiset of portals is 2^{4m+k_1} . For (c), a portal may have one of the grades of service request and the edge that crosses the portal is assigned to have that value. The number of different grades of service request is r .

Hence we can upper bound the size of the lookup table by

$$\begin{aligned}
& n^4 \times n^8 \times \sum_{k_1=1}^{4m} 2^{4m+k_1} r^{k_1} \\
&= n^{12} \times 2^{4m} \sum_{k_1=1}^{4m} (2r)^{k_1} \\
&< n^{12} \times 2^{4m} \sum_{k_1=1}^{4m} (2r)^{4m} \\
&= n^{12} \times 4m(4r)^{4m} \\
&= n^{12} \times m4^{4m+1} r^{4m}
\end{aligned}$$

which is $n^{O(1/\epsilon)}$
since $m = O(\frac{\log n}{\epsilon})$ and $r^{4m} = r^{\frac{4}{\epsilon} \log_r n} = n^{\frac{4}{\epsilon}}$.

Now we consider the running time over the lookup table. The bottom level rectangles have limited number of points, so a brute-force algorithm runs in polynomial time. For each of the upper level rectangles, we compute the minimum values for each table entries by the concatenations and comparisons of the existing entries, so the time to make up for an entry is also polynomial. Therefore, the running time of the DP is upper bounded by

$$2^{O(m)} \times \text{poly}(n) \times \text{Size of the table}$$

, which is $n^{O(1/\epsilon)}$.

Note that, through the DP, all the possible *m-light structures* have been checked and the Structure Theorem ensures that the minimum cost *m-light structure* is within the expected approximation ratio.

5 Proof of the Structure Theorem

The Structure Theorem shows the existence of the *m-light structure*, which keeps itself within the small error allowance from the imaginary *optimal structure*. Once the existence of such a *m-light structure* is shown, the minimum cost *m-light structure* is also within the error allowance. Finding the minimum cost *m-light structure* is the share of the DP. The *optimal structure* mentioned in this section is, of course, the imaginary one.

The proof of the Structure Theorem can be stated as follows. For each rectangle from the $1/3 : 2/3$ -tiling of the problem instance, we choose the *line-separator* that crosses the edges of the *optimal structure* least number of times. Let's name the points at which the edges cross the chosen *line-separator*, as *target_points*. Then the *m-light structure* whose edges and points cross at the nearest portals to the *target_points* will be shown to be within the expected approximation bound.

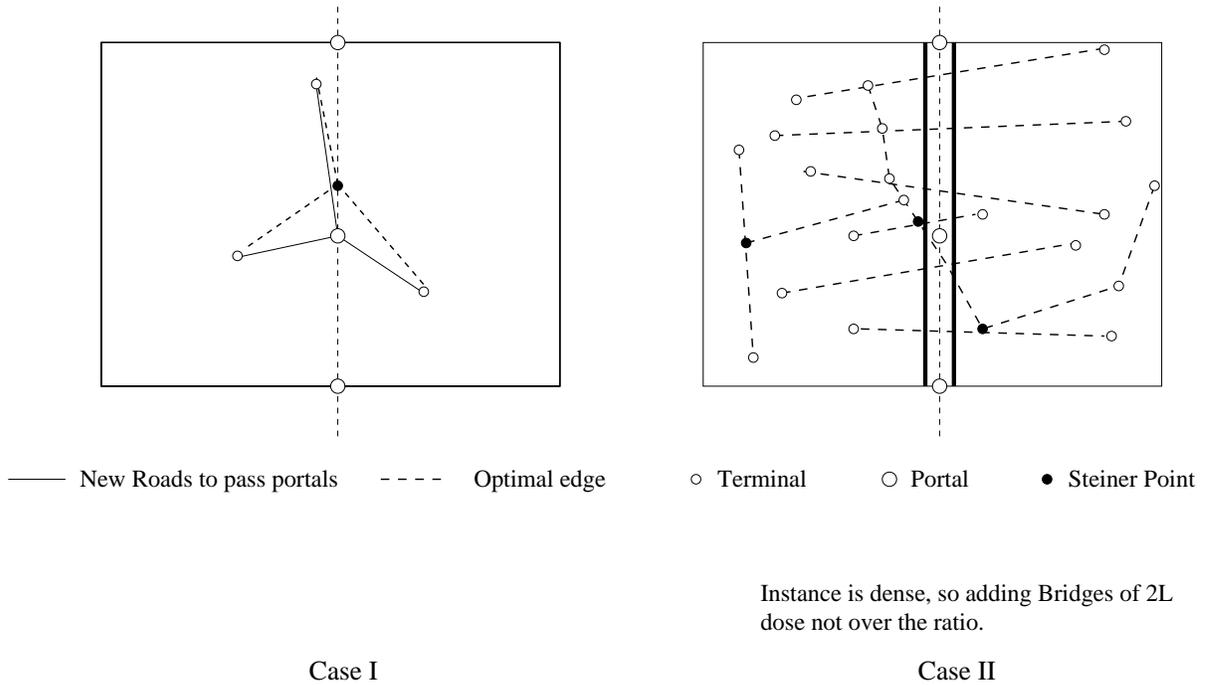


Figure 6: Two cases

of $(1+\epsilon)$ -approximation. Name such a *m-light structure* as a *close structure*. The minimum cost *m-light structure* which has a *cost* less or equal than the *cost* of the *close structure* will be found by the DP.

Proof:(Structure Theorem)

We are going to figure out how much more *cost* the *close structure* has than the *optimal structure*. There are two cases to be considered because the number of crossings between the *optimal structure* and the *line-separator* could be more than m or not. (See Figure 6).

Case I For a rectangle, there is a *line-separator* which is crossed by the *optimal structure* k ($\leq m$) times, where k is the minimum number of the crossings that a *line-separator* in the rectangle may have.

Case II For a rectangle, all the *line-separators* cross the *optimal structure* more than m times.

For the Case I, the maximum *cost* difference between the *optimal structure* and the *close structure* in a rectangle is $3k \cdot c(n) \cdot \frac{L}{m}$ because all the crossing points of the *close structure* at the portals can be Steiner points of degree 3 and each of them may

have a distance of up to L/m to the nearest *target_point*, with the maximum CPUL on the edges.

So, the ratio of the *cost* difference to the *cost* of the *optimal structure* in the rectangle is : $\frac{3k \cdot \frac{L}{m} \cdot c(n)}{\frac{k}{2} \cdot \frac{1}{3} L \cdot c(1)} = \frac{18 \cdot C}{m}$. We may let $m = O(\frac{\log n}{\epsilon})$ so that the ratio of the *close structure's cost* to the *optimal structure's cost* is $\leq (1 + \frac{18 \cdot C}{m}) = (1 + \hat{\epsilon})$.

For the Case II, the estimated minimum *cost* of the *optimal structure* in the middle 1/3 area of a rectangle is $\frac{m}{2} \cdot \frac{1}{3} L \cdot c(1)$ since $k = m$.

Because the *cost* is high in this rectangle, we may add two *bridges* of length L and CPUL $c(n)$ to form the *close structure* instead of considering the details of the crossings, which might cause exponential time complexity. A *bridge* is a straight line, that may be touched by the edges, terminal points and Steiner points of the *close structure*, and which lies parallelly and tightly close to but not touching the *line-separator* on both of its sides. The two *bridges* may be connected through the portal by infinitesimally short line segments whose *costs* are ignored.

So, the ratio of the *cost* difference to the *cost* of the *optimal structure* in the rectangle is : $\frac{2L \cdot c(n)}{\frac{m}{2} \cdot \frac{1}{3} L \cdot c(1)} = \frac{12 \cdot C}{m}$.

As a result, the *cost* difference from the two cases is at most $\frac{18 \cdot C}{m}$. Since the tiling has depth of $O(\log n)$, the approximation ratio is $(1 + \frac{18 \cdot C}{m})^{O(\log n)} = (1 + \frac{18 \cdot C}{m})^{m(\frac{1}{m} O(\log n))} = e^{O(\epsilon)} \leq (1 + \epsilon)$ ■

We describe next the work of the DP in the context of Case I and II as stated above in order to see the coherence between the proof and the DP algorithm.

Case I is usually for the bottom and lower level rectangles because the numbers of points in them are small, so the number of crossings through a *line-separator* would usually be $\leq m$. But m should be a large number to take all the cases of the *optimal structure* passing the *line-separator*. Case II would apply for the upper level rectangles.

For the Case I, where two rectangles are combined and the *line-separator* between them is crossed no more than m times, generally, the crossings at the portal of the small rectangles are moved to the nearest portals of the bigger rectangles. In this way, more than one crossing from the small rectangle may move to the nearest portal of the bigger rectangle if the inter-portal distance of the bigger rectangle is more than twice longer than that of the smaller one. Actually, the portals(crossings) of the smaller rectangle do not move the distance literally, but a short-*bridge*(a line segment) is added to connect the crossings(portals) to the nearest portal of the bigger rectangle. As many short-*bridges* as the number of crossings are added tightly close to the *line-separator* from the locations of the small rectangle's portals(crossings) to the bigger

rectangle's ones. At the same time, the portals of the small rectangle are removed and the crossings are changed to touchings with the corresponding short-*bridges*. Because the short-*bridges* and portals are mathematical lines and points, we may put them tightly close to each other, but some adjustment at the final stage to separate them reasonably within the error allowance could be done if needed. The Structure Theorem is valid in this view of the DP's work, which is clearly polynomial time.

Case II happens when two rectangles are combined, the *line-separator* between them is the concatenation of the two short-*line-separators* (or two sides) of the lower level rectangles and the sum of the numbers of crossings from the two short-*line-separators* is more than m . Then *bridges* are used to deal with the excessive crossings. As explained in the proof of the Structure Theorem, the *bridges* are straight lines, which lie parallelly and tightly close to but not touching the *line-separator* on both of its sides. All the crossings at the two short-*line-separators* are changed to touchings with the *bridges*, removing themselves. Then the *structure* in the upper rectangle is likely to be a network. The edges that crossed the portals in the smaller rectangles are not the crossings at the upper rectangle anymore. Instead, they remain touched with the *bridge* and the *bridge* will have the crossings through the portals no more than m times.

The two *bridges* on both sides of the *line-separator* are connected by infinitesimally small line segments through the portals, and actually in that case only one portal could be used for the crossing. This is how the DP forms the *m-light structures*. A procedure that trims the *approximated structure* may follow the DP to provide a more refined result.

6 Conclusions

In this paper we study the Grade of Service Steiner Minimum Tree problem. This is a recent problem in literature, with some important applications in network design and road interconnection areas. The previous research has focused on geometric analyses and different approximation algorithms were proposed. In this paper we present a PTAS for the GOSST problem. We give some adjustments and use the rectangular partitions technique to the problem instance in order to facilitate the running of the dynamic programming algorithm with less geometric studies. In this way we proceeded straightforward to the $(1 + \epsilon)$ -approximation solution in a polynomial running time.

References

- [1] S. Arora, Polynomial-time approximation schemes for Euclidean TSP and other geometric problems, *Proc. 37th IEEE Symp. on Foundations of Computer Science*, (1996) pp. 2-12.
- [2] S. Arora, Nearly linear time approximation schemes for Euclidean TSP and other geometric problems, *Proc. 38th IEEE Symp. on Foundations of Computer Science*, (1997) pp. 554-563.
- [3] X. Cheng, J.-M. Kim and B. Lu, A polynomial time approximation scheme for the problem of interconnecting highways, *Journal of Combinatorial Optimization*, vol. 5(3), pp.327-343, Feb. 2001.
- [4] G.L. Xue, G.H. Lin and D.Z. Du, Grade of Service Steiner Minimum Trees in the Euclidean Plane, *Algorithmica*, 31(4)(2001), 479–500.
- [5] Ding-Zhu Du, Bing Lu, Hung Ngo and P.M. Pardalos. Steiner Tree Problems, *manuscript (2000)*
- [6] F.K. Hwang, D.S. Richards and P. Winter. The Steiner Tree Problem, *Annals of Discrete Mathematics*, Vol.53, North-Holland (1992).
- [7] F.K. Hwang, A primer of the Euclidean Steiner problem, *Annals of Operations Research*, Vol.33(1991),pp.73-84.
- [8] M.R. Garey, R.L. Graham and D.S. Johnson, The complexity of Computing Steiner Minimal Trees, *SIAM J. Appl. Math.*, 32(1977) 835-859.
- [9] Z.A. Melzak, On the problem of Steiner, *Canadian Mathematics Bulletin*, Vol.4(1961),pp143-148.
- [10] E.N. Gilbert and H.O. Pollak, Steiner minimal trees, *SIAM Journal on Applied Mathematics*, Vol. 16(1968),pp.1-29.
- [11] E.J. Cockayne and D.E. Hewgill, Exact Computation of Steiner minimal trees in the plane, *Information Processing Letters*, Vol. 22(1986),pp.151-156.
- [12] E.J. Cockayne and D.E. Hewgill, Improved computation of plane Steiner minimal trees, *Algorithmica*, Vol.7(1992),pp219-229.
- [13] D.Z. Du and F.K. Hwang, An approach for proving lower bounds: solution of Gilbert-Pollak conjecture on Steiner ratio, *Proceedings of IEEE 31st FOCS*,(1990),pp.76-85.

- [14] MacGregor J. Smith and B. Toppur, Euclidean Steiner minimal trees, minimum energy configurations, and the embedding problem of weighted graphs in E^3 , *Discrete Applied Mathematics*, Vol.71(1997),pp.187-215.
- [15] M. Sarrafzadeh, W.-L. Lin and C.K. Wong, *Floating Steiner Trees*, *IEEE Transactions on Computers*, Vol.47(1998),pp.197-211.
- [16] P. Winter, An algorithm for the Steiner problem in the Euclidean plane, *Networks*, Vol.15(1985),pp.323-345.
- [17] P. Winter and M. Zachariasen, Large Euclidean Steiner minimum trees in an hour, *Technical report 96/34*, <http://www.diku.dk/pawel/publications.html>.
- [18] P. Winter and M. Zachariasen, Large Euclidean Steiner minimum trees: an improved exact algorithm, *Networks*, Vol.30(1998),pp.149-66.
- [19] A. Zelikovsky, An $\frac{11}{6}$ -approximation algorithm for the network Steiner problem, *Algorithmica*, Vol.9(1993),pp.463-470.
- [20] C. Duin and A. Volgenant, The multi-weighted Steiner tree problem, *Annals of Operations Research*, Vol.33(1991),pp.451-469.
- [21] J.R. Current, C.S. Revelle and J.L. Cohon, The hierarchical network design problem, *European Journal of Operational Research*, Vol. 27(1986),pp.57-66.
- [22] A. Balakrishnan, T.L. Magnanti and P. Mirchandani, Modeling and heuristic worst-case performance analysis of the two-level network design problem, *Management Science*, Vol.40(1994),pp.846-867.
- [23] P. Mirchandani, The multi-tier tree problem, *INFORMS Journal on Computing*, Vol.8(1996),pp.202-218.
- [24] C.J. Colbourn and G. Xue, Grade of service Steiner trees on a series-parallel network, to appear in D.-Z. Du, J.M. Smith and J.H. Rubinstein (Eds.), *Advances in Steiner Trees*, Kluwer Academic Publishers, 1998.