지리정보시스템에서 고속도로 연결 문제의 가변적 근사기법

An Adaptive Approximation Method for the Interconnecting Highways Problem in Geographic Information Systems

김준모*, 황병연**

Joon-Mo Kim, Byung-Yeon Hwang

요 약 고속도로 연결문제(Interconnecting Highways problem)는 VLSI 설계, 광 또는 유선 네트워크의 설계, 도로 건설 계획 등의 분야에서 도출되는 여러 가지 배치문제들을 대표하는 추상화 된 문제이다. 도로 건설에 있어 기존의 지점들을 가장 짧은 거리로 상호 연결하는 도로망은 다른 도로망들에 비해 경제적인 면에서 많은 이익을 가져다준다. 즉, 기존의 도로나 도시들을 상호 연결하는 새로운 도로망을 찾는 문제는 중요한 이슈가 된다. 본 논문에서는 NP-hard 문제인 고속도로 연결문제에 대해 '최적에 접근하는 결과치'를 내는 근사방법을 제안한다. 이 방법은 컴퓨팅 자원이 지원되는 한 최적치에 접근하는 근사-결과치를 구할 수 있도록 한다. 따라서 실제 응용에서는 제안된 근사방법에서 산출되는 근사치를 사실상의 최적치로 간주할 수 있게 된다. 선행연구에서의 근사방법과 달리 본 논문에서 제안된 방법은 주어진 문제 인스턴스의 속성에 부합하는 알고리즘을 만들어 낼 수 있도록 하는 큰 장점을 가진다.

Abstract The Interconnecting Highways problem is an abstract of many practical Layout Design problems in the areas of VLSI design, the optical and wired network design, and the planning for the road constructions. For the road constructions, the shortest-length road layouts that interconnect existing positions will provide many more economic benefits than others. That is, finding new road layouts to interconnect existing roads and cities over a wide area is an important issue. This paper addresses an approximation scheme that finds near optimal road layouts for the Interconnecting Highways problem which is NP-hard. As long as computational resources are provided, the near optimality can be acquired asymptotically. This implies that the result of the scheme can be regarded as the optimal solution for the problem in practice. While other approximation schemes can be made for the problem, this proposed scheme provides a big merit that the algorithm designed by this scheme fits well to given problem instances.

주요어 : 근사 알고리즘, 근사 비율, 허용오차, 다항적-시간, 동적 프로그래밍, 다항적-시간 근사기법

KeyWords : Approximation algorithm, Approximation ratio, Error allowance, Polynomial time, Dynamic programming, PTAS(Polynomial Time Approximation Schemes)

1. Introduction

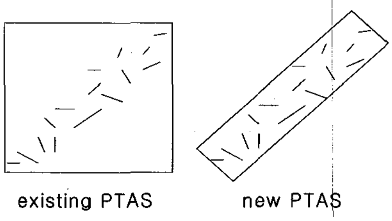
PTAS (Polynomial Time Approximation Scheme) [1],[2] includes the idea of dividing given problem instance to form a dynamic programming. Usually given problem instance is divided by rectangular partitions. During computations, approximate solutions are guided to pass through given points, named portals, on the perimeters of the rectangular partitions.

* 단국대학교 전기전자컴퓨터 공학부 q888@dankook.ac.kr

** 가톨릭대학교 컴퓨터정보공학부 byhwang@catholic.ac.kr

The PTAS for Interconnecting Highways Problem has been presented in[3], but its rectangular partitioning procedure requires a hard condition that the portals of a rectangular partition should be superimposed exactly with those of its upper level partitions. The condition makes the portal locations fixed and the density of portals high unnecessarily. This paper proposes a new PTAS for the same problem to introduce the idea of portal adjustment that makes it possible to design a versatile dynamic programming which dispenses with the condition.

For a problem instance as in <Figure 1> where the highway segments are located alongside a diagonal of a square area, existing PTAS locates portals in terms of the square enclosing the problem instance, while new PTAS does in terms of the smaller rectangle tightly enclosing it. Inside the square of existing PTAS, the number of portals along a perimeter of partitions must be power of 2 to meet the condition mentioned above. This makes unnecessarily many portals that can be reduced by new PTAS, increasing computation time.



<Figure 1> Two Computation Scopes

For the worst-case time complexity, the two PTAS has roughly $O(n^{\frac{1}{\epsilon}})$ time. But the worst case happens rarely. Generally, for a given problem instance, versatile partitioning is needed to design a dynamic programming that fits well to given problem instances.

Interconnecting Highways problem is related to Euclidean Steiner minimum tree(ESMT)[4],[5] and defined as follows from[6]. Interconnecting Highways is to construct the roads of minimum length to interconnect n existing highways $H_1, H_2, \cdots, H_n$ under the constraint that the roads can intersect H_i only at one point, called an exit of H_i, in a designated interval I_i. To avoid unnecessary complexity, we assume that all I_i are disjoint. In this paper, we consider the case that I_i is a line segment, including the two extreme cases that I_i is a point or a line. The case that I_i is a point for all $i = 1, 2, \cdots, n$ is the Steiner minimum tree problem, which is NP-hard. Thus the problem is also NP-hard[7],[8]. Following sections introduce a new PTAS for the Interconnecting Highways Problem.

The remainder of this paper is organized as follows. Section2 mentions what is the subject of the minimization process in the approximation scheme. Section 3 shows that the approximation over the digitized problem instances gives the desired result. In Section 4, the tools that are required to perform the theorem proving are presented. Section 5 shows that the proposed scheme may result in a polynomial-time Dynamic Programming. Section 6 is the proof of the Theorem that shows that there exist the desired solutions that can be found in a polynomial time. Section 7 concludes the paper with the comment on the key point of the proposed idea.

2. The Objective Function

Our objective is to find the minimum-length road for the problem, but we can not design a PTAS in the way to minimize only the road_length. It will be explained later. We should take the way to minimize the total_length, which is the sum of the road_length and segment_length. The road_length is the sum of the lengths of all the roads and the segment_length is that of all the highway segments. APX is the abbreviation for Approximation and OPT for Optimal.

Proposition 1 Under the reasonable assumption that

$$\sum segment\_length \le c \cdot \sum road\_length_{OPT}$$

where c is a constant. The $(1+\varepsilon_1)$-approximation for the total_length implies $(1+\varepsilon_2)$-approximation for the road_length.

Proof :

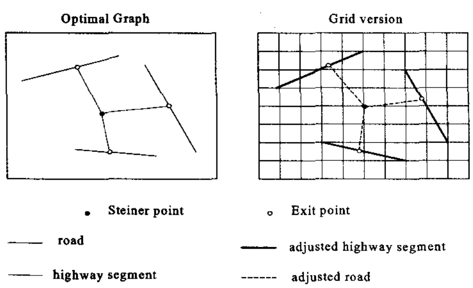
$$\begin{aligned} total\_length_{APX} &= \sum road\_length_{APX} + \sum segment\_length \\ &\le (1+\varepsilon_1)\cdot total\_length_{OPT} \\ &= (1+\varepsilon_1)\cdot(\sum road\_length_{OPT} + \sum segment\_length) \end{aligned}$$

$$\begin{aligned} \sum road\_length_{APX} &\le (1+\varepsilon_1)\cdot \sum road\_length_{OPT} + \varepsilon_1 \cdot \sum segment\_length \\ &\le (1+\varepsilon_1\cdot(1+c))\cdot \sum road\_length_{OPT} \\ &= (1+\varepsilon_2)\cdot \sum road\_length_{OPT} \end{aligned}$$ ■

3. Adjusting problem instances into grids

In order to run the program, the instance of our problem should be adjusted into grids, the integer Coordinates. In fact, the adjustment will move each end point into the nearest grid point. In addition, we assume that the crossing points between the roads, which are actually the steiner points, lie only at the grid points. We now show that the adjustment and the assumption are acceptable.

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



<Figure 2> Difference between the Optimal graph and the Grid version

Proof : By the adjustment and assumption,

$$\left| total\_length^{Original}_{OPT} - total\_length^{Grid}_{OPT} \right| \le 2(3n-3)$$

because the number of edges of this tree structure is $3n-3$. Note the maximum number of points for the problem instance is $3n-2$ that is the sum of endpoints and steiner points and both the end points of an edge can move within the distance 1 each.

Likewise,

$$\left| total\_length^{Original}_{APX} - total\_length^{Grid}_{APX} \right| \le 2n^2$$

where n^2 is the worst case number of edges when the approximated-graph is a complete network. This condition:

$$total\_length^{Grid}_{APX} \le (1+\varepsilon)\cdot total\_length^{Grid}_{OPT}$$

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

$$total\_length^{Original}_{APX} - 2n^2 \le (1+\varepsilon)\cdot(total\_length^{Original}_{OPT} + 2(3n-3))$$

$$\begin{aligned} total\_length^{Original}_{APX} &\le (1+\varepsilon)\cdot total\_length^{Original}_{OPT} + (1+\varepsilon)\cdot 2(3n-3) + 2n^2 \\ &= (1+\varepsilon+\frac{(1+\varepsilon)\cdot 2(3n-3)+2n^2}{total\_length^{Original}_{OPT}})\cdot total\_length^{Original}_{OPT} \\ &= (1+\varepsilon+\frac{(1+\varepsilon)\cdot 2(3n-3)+2n^2}{n^3})\cdot total\_length^{Original}_{OPT} \\ &\le (1+\hat{\varepsilon})\cdot total\_length^{Original}_{OPT} \end{aligned}$$

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

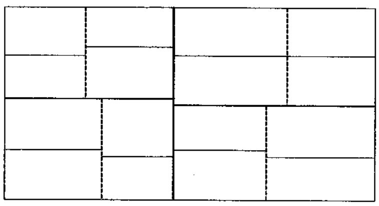
The role of the term n^3 is the key of this proof and it is acquired by choosing the unit length of the grid short enough. We may set is bigger than n^3 with the shorter unit length.

4. 1/3:2/3-tiling, portals and the Structure Theorem

We call the tree that is composed of the segments and roads together and satisfies the problem to be total_tree. The network of total_tree plus the additional roads for the approximation purpose is named a graph. Their length is called total_length.

In case that the total_tree inside the rectangle is $(1+\varepsilon)$-approximation, we call the tree objective total_tree and the objective_graph is defined analogously. A Steiner point is a crossing point of the roads that holds the Steiner point properties. We call highway segments as segments. We mean a rectangle as an axis-aligned rectangle. The size of the rectangle is the length of its long edge. The bounding box of the problem instance is the smallest rectangle enclosing them. A line-separator of a rectangle is a straight line segment parallel to its shorter edge that partitions it into two rectangles of at least $\frac{1}{3}$rd the area. For example, if the rectangle's width W is greater than its height, then a line-separator is any vertical line in the middle $\frac{W}{3}$ of the rectangle. The character m stands for the number of portals on a line-separator. Now 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 sub-trees that are *1/3:2/3*-tiling of the two rectangles, into which the line-separator divides R.



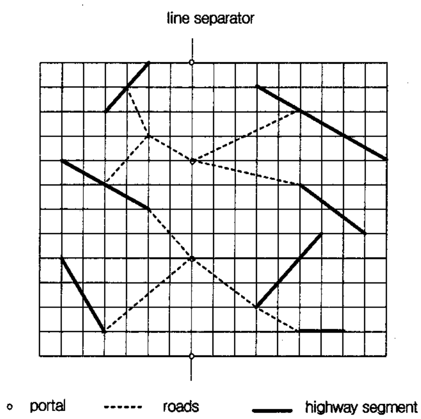
<Figure 3> 1/3:2/3-tiling

Note that rectangles at depth d in the tiling form a partition of the root rectangle. The set of all rectangles at depth d+1 is a refinement of this partition obtained by putting a line-separator through each depth d rectangle of size>1 . The area of any depth d rectangle is at most $\frac{2^d}{3}$ times the total area. The following proposition is therefore immediate.

Proposition 3 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) Let $m \in Z^+$ and π be the roads for the problem instance. 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 the total_tree with π is m-light with respect to S if the followings are true: (i) in each rectangle of tiling S, the roads crosses the line-separator of that rectangle at most m times, (ii) the roads crosses the line-separator only at portals in P.



<Figure 4> m-light graph

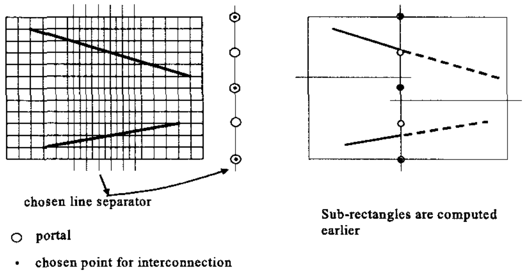
Theorem (Structure Theorem) The following is true for each $\varepsilon > 0$. Every set of highways in the problem has a $(1+\varepsilon)$-approximate total_tree and an associated *1/3:2/3*-tiling of the bounding box such that the roads are m-light for this tiling where $m = O(\frac{\log n}{\varepsilon})$.

5. The Polynomial Time Dynamic Programming (DP)

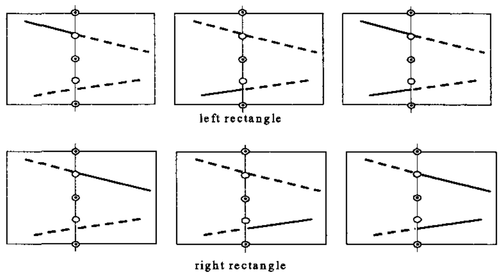
By Proposition 2 and assuming that the Structure Theorem is true, we can build the m-light objective_graph up to the root of the tiling with the DP. The Structure Theorem guarantees the existence of $(1+\varepsilon)$-approximate m-light graphs and tiling S, with respect to which the graphs are m-light, where $m = O(\frac{\log n}{\varepsilon})$. By Proposition 3, the depth of any such tiling is at most $O(\log n)$. We describe a simple DP that finds both S and π in $poly(n)2^{O(m)} = n^{O(\frac{1}{\varepsilon})}$ time, for which the analysis comes later.

The work of the DP is 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 the rectangle, which is the bounding box for the given problem instance, and which contains the m-light objective_graph that satisfies the problem.

Right before the final rectangle is formed, many combinational cases must have been 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 choices of portals <Figure 5>. Then, for each choice of a line-separator and a multi-set of portals, there are choices of the inclusion/exclusion of the segments for the two sub-rectangles <Figure 6>. Thus there are many combinational cases as above and each of them has its own minimum cost m-light graph. Such a minimum cost m-light graph is the concatenation of the two smaller m-light graphs from each of the sub-rectangles. So a rectangle holds such minimum cost m-light graphs as many as the number of the combinations.



<Figure 5> Partitioning with a line-separation



<Figure 6> Combinations in terms of keeping or deleting segment

The same observations hold repeatedly in each of the sub-rectangle for finding its m-light graphs until the work reaches the bottom most rectangles where there are limited number of segments and so the brute-force algorithm could get the minimum cost m-light graphs for each of the combinational cases in the smallest rectangle.

Out of all the minimum cost m-light graphs from all the cases of combinations in the final rectangle, we choose the most minimum cost one(s), which is the m-light objective_graphs, and call it the approximated solution for the problem.

Now, we show the number of entries of the lookup table for this DP is polynomial and the run time for each of the entries is poly time. An entry is indexed by the triple: (a) a rectangle, (b) a multi-set of k_1 $(\le 4m)$ portals along the perimeter of the rectangle, and (c) a set of the segments inside the rectangle.

For (a), the number of distinct rectangles is at most $\binom{2(3n-3)}{4}$ since the maximum number of points are $2(3n-3)$ from the Proof of Proposition 2. For (b), Each rectangle has 4 sides, and is part of the line-separator of some ancestor of the rectangle. The m portals on the line-separator are evenly spaced, so they are completely determined once we know the line-separator. But the number of choices for 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 $\le 4m$ portals on the four sides, the number of ways of choosing a multi-set of k_1 out of them is at most 2^{4m+k_1}. For (c), the number of the sets of the segments in the rectangle is limited to polynomial by the bundling of segments into one when more than m segments pass a line-separator of the rectangle. So the maximum number of segments passing the perimeter is $< 4m$ and the number of the sets is $< 2^{4m}$.

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

$$\sum_{k_1=1}^{4m} n^4 \times n^8 \times 2^{4m+k_1} \times 2^{4m}$$

, which is $O(n^{12}\, 2^{12m}) = n^{O(\frac{1}{\varepsilon})}$.

Now we consider the running time over the lookup table. The bottom level rectangles has limited number of segments of $O(m)$ and so run the brute-force algorithm for each in $2^{O(m)}$ time. For each of the upper level rectangles, we compute the minimum value by the comparition between the sub-rectangles, so the time is polynomial. Therefore, the running time of the DP is upper bounded by $2^{O(m)} \times poly(n) \times \text{size of the table}$, which is $n^{O(1/\varepsilon)}$.

In this DP, we can check all the m-light graphs so that the $(1+\varepsilon)$-approximate graph shall be found. The existence of $(1+\varepsilon)$-approximate graph in any chosen rectangle is ensured by the Structure Theorem.

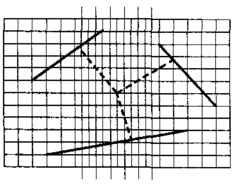
6. Proof of the Structure Theorem

The Structure Theorem shows the existence of the m-light objective_graph, which keeps itself within the small error allowance from the imaginary optimal total_tree. Once the existence of such a m-light objective_graph is shown, the minimum cost m-light graph is also within the error bound. Finding the minimum cost m-light graph is the DP's share. The optimal total_tree mentioned in this paper is the imaginary one.

Note that the m-light graphs does not have to satisfy the properties that the optimal total_tree should have. That is, the optimal total_tree should be a Steiner tree, but the m-light graph does not have to be. The final approximated-graph is normally picked out from the m-light objective_graph in polynomial time by the Spanning tree routine.

The proof of the Structure Theorem can be stated as follows. For each rectangle over the *1/3:2/3*-tiling of the problem instance, we continue to choose a line-separator that crosses with the edges of the optimal total_tree less times than the others. Name the points, at which the edges cross with the chosen line-separator to be target_points. Then the m-light graph whose edges and points cross at the nearest portals from the target_points will be shown to be within the expected approximation bound.

The analytic sum of the lengths between the target_points and the nearest portals through which the parts of the m-light structure pass is used for the estimation of the length difference between the optimal structure and the m-light structure. The length difference will be compared with the minimum estimation of the optimal structure's length, which is measured with the use of Lemma 1. Let a unit_band in a rectangle be the sub-rectangle with the unit length of width and the height of length up to L, a side of the rectangle.q



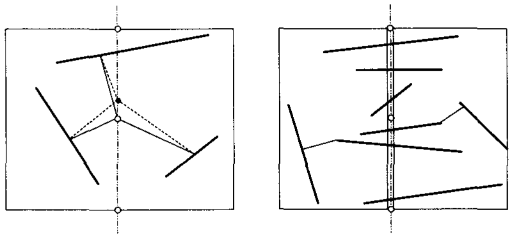
<Figure 7> Six unit bands in the middle

Lemma 1 When a line-separator inside the unit_band crosses with the roads and segments $k(\in N \cup \{0\})$ times, the $total\_length^{Grid}_{OPT}$ minimum in the unit_band is $\frac{k}{2}$.

Proof : For a crossing, the length of the corresponding edge or segment is the shortest when the edge stops at the line-separator and it is $\geq \frac{k}{2}$ of the unit length. So, the minimum cost in the unit_band is $\frac{k}{2}$.

For each rectangle over the *1/3:2/3*-titling, we choose only one line-separator, in the middle $\frac{1}{3}$ area of it, that crosses with the optimal total_tree k times. The other line-separators in the area cross at least k times. If $k = 0$, the case will turn out to be simple. As a result, the minimum $total\_length^{Grid}_{OPT}$ in the middle $\frac{1}{3}$ area could be estimated as $\frac{k}{2} \cdot \frac{1}{3} L$. Over the chosen line-separators above, we may build up a m-light graph, which will be shown to have the approximation ratio of $(1+\varepsilon)$. Name such a m-light graph as a close graph. The minimum length m-light graphs which has less or equal length than the close graph will be found by the DP.

Proof: (Structure Theorem) We are going to figure out how much more length the close graph has than the optimal total_tree does. There are two cases to be considered because the number of the crossings between the optimal total_tree and the line-separator could be more than m or not, and they should be observed separately.



<Figure 8> Two case

Case I For a rectangle, there is a line-separator, which is crossed by the optimal total_tree $k\,(\leq m)$ 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 with the optimal total_tree more than m times.

For Case I, the maximum length difference between the optimal total_tree and the close graph in a rectangle is $3k \cdot (\frac{L}{m})$ because all the crossing points of the close graph at the portals can be Steiner points of degree 3 and each of them may have a distance of up to $\frac{L}{m}$ to the nearest target_points.

So, the ratio of the length difference to $total\_length^{Grid}_{OPT}$ in the rectangle is:

$\frac{\frac{3k}{m}L}{\frac{k}{2}\cdot\frac{1}{3}L} = \frac{18}{m}$. We may let $m = O(\frac{\log n}{\varepsilon})$ so that the ratio of the close graph's length to the optimal total_tree's is $\le (1+\frac{18}{m}) = (1+\hat{\varepsilon})$.

For Case II, the estimated minimum $total\_length^{Grid}_{OPT}$ in the $\frac{1}{3}$ area of a rectangle is $\frac{m}{2}\cdot\frac{1}{3}L$ since $k = m$.

Because the density of the edges in this case is high, we may add two bridges of length L to form the close graph instead of considering the details of the crossings, which might cause exponential time complexity.

The bridge is a part of the roads and a straight line, with which the other parts of the roads, segments and Steiner points of the close graph may touch, and which lies parallely and tightly close to —but not touching — the line-separator on both of its sides. But the two bridges may be connected each other through the portal by infinitesimally short line segments whose lengths are ignored. So, the ratio of the length difference to the $total\_length^{Grid}_{OPT}$ in the rectangle is: $\frac{2L}{\frac{m}{2}\cdot\frac{1}{3}L} = \frac{12}{m}$. As a result, the length difference from the two cases is at most $\frac{18}{m}$. Since the tiling has depth of $O(\log n)$, the approximation ratio is

$$(1+\frac{18}{m})^{O(\log n)} = (1+\frac{18}{m})^{m(\frac{1}{m}O(\log n))} = e^{O(\varepsilon)} \le (1+\varepsilon). \quad ■$$

We may describe the works of the DP in view of Case I and Case II stated above so as to see the relationship between the proof and the DP's works. Case I is usually for the bottom and lower level rectangles because the numbers of the segments in them are small and so the number of crossings through a line-separator would usually be $\le m$. But m should be a large number to take all the cases of the optimal structure passing the line-separator. Case II would be for the upper level rectangles.

For 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, and further, more than one crossings from the small rectangle may move to one 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.

However, 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-bridge as the number of the crossings are added tightly close to the line-separator from the locations of the small rectangle's portals (crossings) to the bigger rectangle's. At the same time, the portals of the small rectangle are removed and the crossings are changed to touching with the corresponding short-bridge. Because the short-bridge and portals are mathematical lines and points, we may put them tightly close 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 proved in such a 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 parallely 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 touching with the bridges, removing themselves. Then the graph in the upper rectangle is likely to be a network.

The roads 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. But the bridges and the segments are just connected if they cross, probably violating the constraint that the roads can intersect a segment only at one point where bridges are parts of the roads. The violation will be recovered by the Spanning Tree procedure that follows the DP.

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 graphs.

It is mentioned in the introduction that we can not design a PTAS in the way to minimize only the road_length and now it can be explained. We noted that the minimum $total\_length_{OPT}$ is $\frac{k}{2}\cdot\frac{1}{3}L$ and it is needed to show that the ratio of the length increase is relatively small. That is, the minimum $total\_length_{OPT}^{Grid}$ should be big enough relative to the length increase. But, if we consider only the road_length then the minimum $total\_length_{OPT}^{Grid}$ can be almost zero, resulting that the ratio gets big unboundedly. Note that the number of portals, m, on a line-separator is supposed to be reasonably big enough to cover every possible cases of crossings. If m is a small number and the intersections on the line-separator is too many, we can not apply the Structure Theorem, i. e. many steiner points can not cross neighboring segments.

7. Conclusion

In the proof of Structure Theorem, it is shown that the amount of added lengths of roads at each partition is less than or equal to an allowance so that the total sum of the difference to the optimal solution can not be greater than the small amount, ε, resulting in $(1+\varepsilon)$-approximation. In new PTAS, the locations of portals neither affect nor depend on the way of partitioning the problem instance. Just choosing the proper number of portals along the perimeter of a partition according to the problem instance satisfies the requirement.

This paper mentions the importance of locating portals by showing a new PTAS. It shows that the difference to the optimal solution incurred by passing through a portal at a level of partitioning is within the allowance as well as the differences at the upper levels are. This property can be applied to the design of approximation algorithms for similar problems whose solutions may have an allowance, like ε, for the difference to the optimal solutions. The point is that the allowance dispenses with the conditions to fix the locations of portals.

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, issue 3, 2001, pp. 327-343.

[4] Z.A. Melzak, "On the problem of Steiner", Canadian Mathematics Bulletin, Vol. 4, 1961, pp. 143-148.

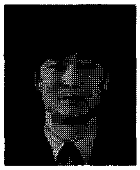
[5] E.N. Gilbert and H.O. Pollak, "Steiner minimal

trees", SIAM Journal on Applied Mathematics, Vol. 16, 1968, pp.1-29.

[6] DING-ZHU DU, FRANK K. HWANG and GUOLIANG XUE. "Interconnecting Highways", SIAM Discrete Mathematics, Vol. 12, 1999, pp. 252-261.

[7] Ding-Zhu Du, Ker-I Ko, Theory of Computational Complexity, Wiley Inter-Science 2000.

[8] Michael R. Garey, David S. Johnson, Computers and Intractability A Guide to the Theory of NP-Completeness, Freeman, 1978



김준모
1989년 서울대학교 컴퓨터공학과 (공학사)
2001년 Computer Science at the University of Minnesota (공학박사)
2002년 ~ 2004년 한국정보보호진흥원 연구원
2004년 ~ 현재 단국대학교 전기전자컴퓨터공학부 전임강사
관심분야 : 계산이론(Approximations for NP-hard problems, the Church-Turing thesis)

황병연
1986년 서울대학교 컴퓨터공학과 (공학사)
1989년 한국과학기술원 전산학과 (공학석사)
1994년 한국과학기술원 전산학과(공학박사)
1999년 ~ 2000년 University of Minnesota Visiting Scholar
1994년 ~ 현재 가톨릭대학교 컴퓨터정보공학부 교수
관심분야 : 지리정보시스템, XML 데이타베이스, 데이타 마이닝, 정보검색