An Approximation Algorithm for Partitioning a Hyperrectilinear Polytope with Holes

by

Teofilo Gonzalez and Mohammadreza Razzazi‡
Department of Computer Science
The University of California
Santa Barbara, CA 93106

EXTENDED ABSTRACT

Let RP be a hyperrectilinear polytope in E^d, for $d \geq 2$ (rectilinear polygon when $d = 2$) and let H be a set of disjoint hyperrectilinear polytopes in E^d defined inside RP. Polytope RP is referred to as the boundary and the set H is a set of holes. For $IP = (RP, H)$, we use $p(IP)$ to denote the $(d$-1)-volume of the hyperplane segments that define RP plus the sum of the $(d$-1)-volume of the hyperplane segments that define the holes in H. In this paper we consider the $RP-RP_d$ problem in which RP is partitioned into hyperrectangles (rectangles when $d = 2$) by introducing a set of orthogonal hyperplane segments (line segments when $d = 2$) whose total $(d$-1)-volume (length when $d = 2$) is least possible. We use $m(IP)$ to denote the total $(d$-1)-volume of the partitioning segments in an optimal solution to IP. The problem of finding $m(IP)$ given IP is NP-hard for all $d \geq 2$. In this paper we present an $O(dn \log n)$ approximation algorithm for the $RP-RP_d$ problem that generates solutions whose $(d$-1)-volume is at most $(2d-1.5)p(IP) + (4d-2)m(IP)$, where n is the total number of segments in RP and H.

The $RP-RP_2$ problem models the channel definition phase of a CAD system [R] and it was shown to be NP-hard by Lingas et. al [LPRS]. Several approximation algorithms for this problem have been developed. These algorithms are given in [R], [U], [L], [DC], [L1], and [L2]. The best of these algorithms are the ones reported in [L1] and [L2].

Levcopoulos' algorithm [L2] consists of two algorithms. The first algorithm has an approximation bound $12m(IP)+6p(IP)$, but Levcopoulos conjectures that it is bounded by $7m(IP) + 3.5p(IP)$. Obviously, this algorithm is advantageous when $p(IP) < m(IP)$. The second algorithm is advantageous when $p(IP) \geq m(IP)$. In this paper we develop for the $RP-RP_2$ problem an approximation algorithm that generates solutions whose total length is bounded by $6m(IP) + 2.5p(IP)$.

We also consider a more general version of the problem, i.e., when it is defined over E^d for $d > 2$, for which there are no previous approximation algorithms. Our $O(dn \log n)$ approximation algorithm

‡ Current address: Computer Engineering and Science Department, Case Western Reserve University, Cleveland, Ohio 44106.

that generates solutions whose $(d\text{-}1)$-volume is at most $(2d\text{-}1.5)p(IP) + (4d\text{-}2)m(IP)$.

APPROXIMATION ALGORITHM

Let us now discuss our approximation algorithm for the $RP-RP_d$ problem. First a hyperrectangle R (whose facets are orthogonal to axes) is placed to include the hyperrectilinear polytope, and each of the corners of RP and H is replaced by point. The set of points is referred to as set P. In other words, from an instance $IP = (RP, H)$ of the $RP-RP_d$ problem we construct an instance $I = (R, P)$ of the $RG-P_d$ problem. This problem is solved by the divide-and-conquer procedure defined in [GRZ] (other approximation algorithms appear in [GZ1], [GZ2], and [GRSZ]). All the parts of the hyperplane segments introduced by that algorithm inside RP but not inside the set of holes H are said to form the solution to the original problem, if it is the case that the segment includes a point in P or there is another segment introduced by the divide-and-conquer algorithm incident to it. We should point out that if we do not place these two restrictions, then there could be about n^2 segments in our solution. By placing these two restrictions we limit the number of segments to $O(n)$. The final segments are referred to as $E_{apx}(IP)$ and the segments introduced by the divide-and-conquer procedure that are inside RP but outside H are referred to as SET(IP). Note that $V_{d-1}(\text{SET}(IP)) \geq V_{d-1}(E_{apx}(IP))$, where $V_{d-1}(A)$ denotes the sum of the d-1 volume of the elements in set A.

Before we explain this in more detail, let us explain the divide-and-conquer procedure given in [GRZ]. The $RG-P_d$ problem is formally defined by $I = (R = (O, X), P)$, where O and X define a hyperrectangle or boundary R ($O = (o_1, o_2, \ldots, o_d)$ is the "lower-left" corner of the hyperrectangle (*origin* of I), and $X = (x_1, x_2, \ldots, x_d)$ are the dimensions of the boundary) in d-dimensional Euclidean space (E^d), and $P = \{ p_1, p_2, \ldots, p_n \}$ is a set of points (degenerate holes) inside hyperrectangle R. We shall refer to the d dimensions (or axes) of E^d by the integers 1,2, ..., d.

Procedure PARTITION begins by relabeling the dimensions so that $x_1 \geq x_2 \geq \ldots \geq x_d$. Then it checks if $P(I)$ is empty and if so, it returns. Otherwise, it introduces a *mid–cut* or an *end–cut*. A *mid-cut* is a hyperplane segment orthogonal to the 1-axis that intersects the center of the hyperrectangle (i.e., it includes point $(o_1+x_1/2, o_2+x_2/2, \ldots, o_d+x_d/2)$) and an *end-cut* is a hyperplane segment orthogonal to the 1-axis that contains either the "leftmost" or the "rightmost" point in $P(I)$. A *mid–cut* is introduced when the two resulting subproblems have at least one point each. Otherwise, an *end–cut* is introduced. The *end–cut* intersects the leftmost point if such a point is not located to the left of the center of the hyperrectangle, otherwise the *end–cut* intersects the rightmost point.

ANALYSIS

Now let us analyze the performance of our algorithm. To establish the time complexity bound is simple since it follows from the fact that the number of points is n; the time complexity bound in [GRZ];

and the fact that $O(n)$ segments are introduced. Now let us concentrate on the approximation bound. Let $E_{opt}(IP)$ be any optimal hyperrectangular partition for IP. Let R' be any hyperrectangle in it. Note that all the d-volume in R' must also be in RP. When running the algorithm we may think of the problem instance as being formed by $I' = (R, R', P)$, even though the algorithm does not know R'. Let BDRY($R'(I)$) be the (d-1)-volume of the facets in $R'(I)$ located inside $R(I)$, and let OV($R'(I)$) be (d-1)-volume of the facets of $R'(I)$ inside $R(I)$ that overlap with the hyperplane segments introduced by PARTITION. Later on we establish that the set of segments introduced by the algorithm inside or on R', which we denote by SET($R'(I)$) is such that

$$V_{d-1}(\text{SET}(R'(I))) \leq (2d\text{-}1.5)\ \text{BDRY}(R'(I)) + \text{OV}(R'(I)).$$

Summing over all R', $\sum V_{d-1}(\text{SET}(R'(I))) \leq \sum$ ($(2d\text{-}1.5)$ BDRY($R'(I)$) + OV($R'(I)$)). Since $\sum$ BDRY($R'(I)$) $= 2m(IP) + p(IP)$, and $\sum$ OV($R'(I)$) is $m(IP)$ (note that the only new segments that overlap with the boundary are those in $E_{opt}(IP)$), we know that

$$V_{d-1}(E_{apx}(IP)) \leq (4d\text{-}2)m(IP) + (2d\text{-}1.5)p(IP).$$

Theorem 1: $V_{d-1}(E_{apx}(IP)) \leq (4d\text{-}2)m(IP) + (2d\text{-}1.5)p(IP)$.
Proof: By the above discussion. □

BOUND FOR V_{d-1}(SET($R'(I)$))

Let us now show that for every problem instance I defined above, algorithm PARTITION introduces a set of hyperplane segments inside $R'(I)$ whose total (d-1)-volume is at most ($2d$-1.5) BDRY($R'(I)$) + OV($R'(I)$). In lemma 1 we prove a stronger result (which is easier to prove) that uses the CARRY function. One may visualize our proof as follows. Every time a hyperplane segment is introduced inside R' by the algorithm it is colored red. The segments in R' that overlap with a cut (OV) or segments in BDRY belong to an instance without points, are colored blue. Our approach is to bound the (d-1)-volume of the red segments by that of the blue segments. The segments in SET represent the segments introduced in R' and CARRY represents some previously introduced red segments that have not yet been accounted for by blue segments. The proof consists of showing that at all time the (d-1)-volume of the red segments can be accounted by that of the blue segments. There are many technical details that for brevity we cannot include. The result in this section, whose proof is complex, is given by the following lemma.

Lemma 1: For any problem instance I, SET($R'(I)$) + CARRY(I) $\leq$ ($2d$-1.5) BDRY($R'(I)$) + OV($R'(I)$).
Proof: The proof, which for brevity will be omitted, is by induction on the number of points inside I, i.e., $|P(I)|$. It is interesting to note that besides the generalization of our arguments to d dimensions, there are only a couple of cases more in the general proof than in the one for the case when $d = 2$. □

We have developed an $O(dn\ log\ n)$ approximation algorithm for the $RP-RP_d$ problem that generates solutions whose (d-1)-volume is at most $(2d-1.5)p(IP) + (4d-2)m(IP)$. Obviously, when $p(IP) \leq m(IP)$, it generates reasonably good solutions. Our approximation bound degrades when $p(IP)$ is very large compared to $m(IP)$. Developing an efficient approximation algorithm for this other case seems difficult. The approach used for the case when $d = 2$, cannot be generalized to arbitrary d.

REFERENCES

[DC] Du, D. Z. and Chen Y. M., "On Fast Heuristics for Minimum Edge Length Rectangular Partition," Technical Report, MSRI 03618-86, Feb. 1986.

[DPS] Du, D. Z., L. Q. Pan and M. T. Shing, "Minimum Edge Length Guillotine Rectangular Partition," Technical Report, MSRI 02418-86, Jan. 1986.

[GRZ] Gonzalez, T., M. Razzazi, and S. Zheng, "An Efficient Divide-and-Conquer Approximation Algorithm for Hyperrectangular Partitions," Technical Report TR-89-24, CS Dept., UCSB, October 1989 (Also in proceedings of the Second Canadian Computational Geometry Conference, August 1990, 214 - 217.

[GZ1] Gonzalez, T. and S.-Q. Zheng, "Bounds for Partitioning Rectilinear Polygons", Proc. Symp. Computational Geometry, June 1985, pp. 281-287, (also appears as Technical Report #85-22, CS Dept., UCSB, Dec. 1985).

[GZ2] Gonzalez, T. and S.-Q. Zheng, "Approximation Algorithms for Partitioning Rectilinear Polygons", Technical Report # 85-21, CS Dept., UCSB, Dec. 1985, *Algorithmica*, 5, January 1990, 11 - 42.

[GZ3] Gonzalez, T and S. Q. Zheng, "Improved Bounds for Rectangular and Guillotine Partitions," *Journal of Symbolic Computation*, 7, 1989, 591 - 610.

[L] Lingas, A., "Heuristics For Minimum Edge Length Rectangular Partitions of Rectilinear Figures," 6th GI-Conference, Dortmund, 1983, Lecture Notes in Computer Science. 195 (Springer-Verlag).

[LPRS] Lingas, A., R. Y. Pinter, R. L. Rivest, and A. Shamir, "Minimum Edge Length Partitioning of Rectilinear Polygons," Proc. 20th Annual Allerton Conference on Communication, Control, and Computing, Monticello, Illinois, Oct. 1982.

[L1] Levcopoulos, C., "Minimum Length and Thickest-First Rectangular Partitions of Polygons," Proceedings of the 23rd Allerton Conference on Communication, Control and Computing, U. of Illinois, Oct. 1985.

[L2] Levcopoulos, C., "Fast Heuristics for Minimum Length Rectangular Partitions of Polygons," Proceedings of the 2nd Computational Geometry Conference, June 1986.

[R] Rivest, R. L., The "PI" (Placement and Interconnect) System, Proc. 19th Design Automation Conference, June 1982.