

University of Texas Rio Grande Valley

ScholarWorks @ UTRGV

Computer Science Faculty Publications and
Presentations

College of Engineering and Computer Science

8-2020

Polyhedra Circuits and Their Applications

Bin Fu

The University of Texas Rio Grande Valley

Pengfei Gu

The University of Texas Rio Grande Valley

Yuming Zhao

Follow this and additional works at: https://scholarworks.utrgv.edu/cs_fac



Part of the [Computer Sciences Commons](#)

Recommended Citation

Fu B., Gu P., Zhao Y. (2020) Polyhedral Circuits and Their Applications. In: Zhang Z., Li W., Du DZ. (eds) Algorithmic Aspects in Information and Management. AAIM 2020. Lecture Notes in Computer Science, vol 12290. Springer, Cham. https://doi.org/10.1007/978-3-030-57602-8_2

This Article is brought to you for free and open access by the College of Engineering and Computer Science at ScholarWorks @ UTRGV. It has been accepted for inclusion in Computer Science Faculty Publications and Presentations by an authorized administrator of ScholarWorks @ UTRGV. For more information, please contact justin.white@utrgv.edu, william.flores01@utrgv.edu.

Polyhedra Circuits and Their Applications *

Bin Fu¹, Pengfei Gu¹, and Yuming Zhao²

¹Department of Computer Science
University of Texas - Rio Grande Valley, Edinburg, TX 78539, USA

²School of Computer Science
Zhaoqing University, Zhaoqing, Guangdong 526061, P.R. China

Preliminary Version

Abstract

We introduce polyhedra circuits. Each polyhedra circuit characterizes a geometric region in \mathbb{R}^d . They can be applied to represent a rich class of geometric objects, which include all polyhedra and the union of a finite number of polyhedra. They can be used to approximate a large class of d -dimensional manifolds in \mathbb{R}^d . Barvinok [3] developed polynomial time algorithms to compute the volume of a rational polyhedra, and to count the number of lattice points in a rational polyhedra in a fixed dimensional space \mathbb{R}^d with a fix d . Define $T_V(d, n)$ be the polynomial time in n to compute the volume of one rational polyhedra, $T_L(d, n)$ be the polynomial time in n to count the number of lattice points in one rational polyhedra with d be a fixed dimensional number, $T_I(d, n)$ be the polynomial time in n to solve integer linear programming time with d be the fixed dimensional number, where n is the total number of linear inequalities from input polyhedra. We develop algorithms to count the number of lattice points in the geometric region determined by a polyhedra circuit in $O(nd \cdot r_d(n) \cdot T_V(d, n))$ time and to compute the volume of the geometric region determined by a polyhedra circuit in $O(n \cdot r_d(n) \cdot T_I(d, n) + r_d(n)T_L(d, n))$ time, where n is the number of input linear inequalities, d is number of variables and $r_d(n)$ be the maximal number of regions that n linear inequalities with d variables partition \mathbb{R}^d . The applications to continuous polyhedra maximum coverage problem, polyhedra maximum lattice coverage problem, polyhedra $(1 - \beta)$ -lattice set cover problem, and $(1 - \beta)$ -continuous polyhedra set cover problem are discussed. We also show the NP-hardness of the continuous maximum coverage problem and set cover problem when each set is represented as union of polyhedra.

Keywords: Lattice points, Volume, Polyhedra circuits, Union

*This research is supported in part by National Science Foundation Early Career Award 0845376 and Bensten Fellowship of the University of Texas - Rio Grande Valley.

1. Introduction

Polyhedra is an important topic in mathematics, and has close connection to theoretical computer science. There are two natural topics in polyhedra, computing the volume and counting the number of lattice points.

The problem of counting the number of lattice points in a polyhedra has a wide variety of applications in many areas, for example, number theory, combinatorics, representation theory, discrete optimization, and cryptography. It is related to the following problem: given a polyhedra P which is given by a list of its vertices or by a list of linear inequalities, our goal is to compute the number $|P \cap \mathbb{Z}^d|$ of lattice points in P . Researchers have paid much attention to this problem. Ehrhart [15] introduced Ehrhart polynomials that were a higher-dimensional generalization of Pick's theorem in the Euclidean plane. Dyer [14] found polynomial time algorithms to count the number of lattice points in polyhedra when dimensional number $d = 3, 4$. Barvinok [4, 5] designed a polynomial time algorithm for counting the number of lattice points in polyhedra when the dimension is fixed. The main ideas of the algorithm were using exponential sums [8, 9] and decomposition of rational cones to primitive cones [26] of the polyhedra. Dyer and Kannan [13] simplified Barvinok's polynomial time algorithm and showed that only very elementary properties of exponential sums were needed to develop a polynomial time algorithm. De Loera et al. [22] described the first implementation of Barvinok's algorithm called LattE. Some other algebraic-analytic algorithms have been proposed by many authors (for example, see [2, 6, 7, 20, 23, 24].)

Computing exactly the volume of a polyhedra is a basic problem that have drawn lots of researchers' attentions [19, 11, 17, 1, 21, 3]. It is known that this problem is $\#P$ -complete if the polyhedra is given by its vertices or by its facets [12, 18]. Cohen and Hickey [11] and Von Hohenbalken [17] proposed to compute the volume of a polyhedra by triangulating and summing the polyhedra. Allgower and Schmidt [1] triangulated the boundary of the polyhedra to compute the volume of polyhedra. Lasserre [19] developed a recursive method to compute the volume of polyhedra. Lawrence [21] computed the volume of polyhedra based on Gram's relation for polyhedra. Barvinok [3] developed a polynomial-time algorithm to compute the volume of polyhedra by using the exponential integral.

Motivation: The existing algorithms related to count the number of lattice points and to compute the volume of the polyhedra concerned with only one polyhedra. In order to have broader applications, it is essential to develop algorithms to deal with geometric objects that can be generated by a list of polyhedra via unions, intersections, and complementations. In this paper, we propose polyhedra circuits. Polyhedra circuits can be used to represent a large class of geometric objects. We propose algorithms to compute the volume of the geometric regions by polyhedra circuits, and to count the number of lattice points in the geometric regions by polyhedra circuits.

Contributions: We have the following contributions to polyhedra. 1. We introduce polyhedra circuits. Each polyhedra circuit characterizes a geometric

region in \mathbb{R}^d . They can be applied to represent a rich class of geometric objects, which include all polyhedra and the union of a finite number of polyhedra. They can be used to approximate a large class of d -dimensional manifolds in \mathbb{R}^d . 2. We develop an algorithm to compute the volume of the geometric region determined by a polyhedra circuit. We also develop an algorithm to count the number of lattice points of the geometric region determined by a polyhedra circuit. Our method is based on Barvinok's algorithm, which is only suitable for polyhedra. 3. We apply the methods for polyhedra circuits to support new greedy algorithms for the maximum coverage problem and set cover problem, which involve more complex geometric objects. The existing research results about the geometric maximum coverage problem and set cover problem only handle simple objects such as balls and rectangular shapes. All of our algorithms run in polynomial time in \mathbb{R}^d with a fixed d .

Organization: The rest of paper is organized as follows. In Section 2, we introduce some important theorems and definitions. In Section 3, we introduce polyhedra circuits and develop polynomial time algorithms to compute the volume of a given list of linear inequalities and to count the number of lattice points in a given list of linear inequalities. In Section 4, we apply the algorithm to polyhedra maximum coverage problem. In Section 5, we apply the algorithm to polyhedra maximum lattice coverage problem. In Section 6, we apply the algorithm to polyhedra $(1 - \beta)$ -lattice set cover problem. In Section 7, we apply the algorithm to $(1 - \beta)$ -continuous polyhedra set cover problem. Section 8 shows the NP-hardness of the maximum coverage problem and set cover problem when each set is represented as union of polyhedrons. Section 9 summarizes the conclusions.

2. Preliminaries

In this section, we introduce some definitions and some important theorems that play a basic role in our algorithms.

Definition 1. [[4]] A *rational polyhedra* $P \in \mathbb{R}^d$ is a set defined by finitely many linear inequalities:

$$P = \left\{ x : \sum_{j=1}^d a_{ij}\xi_j \leq b_i \text{ for } x = (\xi_1, \dots, \xi_d) \text{ and } i \in I \right\}$$

for some finite (possibly empty) set I , where a_{ij} and b_i are integer numebrs.

Note that we always assume that the polyhedra that we deal with are rational polyhedra throughout the paper.

Theorem 2 ([3]). *There exists a polynomial time algorithm for computing the volume of one rational polyhedra.*

Theorem 3 ([4]). *Let us fix $d \in \mathbb{N}$. Then there exists a polynomial time algorithm for counting lattice points in one rational polyhedra when the dimension d is fixed.*

Definition 4. [[25]] *Integer Linear Programming* problem is a constrained optimization problem of the form:

$$\max \{cx \mid Ax \leq b; x \text{ integral}\},$$

where A is a given $n \times d$ matrix, b an n -vector and c a d -vector, and the entries of A , b and c are rational numbers.

Theorem 5 ([10]). *Let us fix $d \in \mathbb{N}$. Then there exists a las vegas algorithm to solve the integer linear programming problem when the dimension d is small.*

Definition 6. Let us fix $d \in \mathbb{N}$. Define $T_V(d, n)$ be the polynomial time in n to compute the volume of one rational polyhedra by Theorem 2, $T_L(d, n)$ be the polynomial time in n to count the number of lattice points in one rational polyhedra with d be a fixed dimensional number by Theorem 3, $T_I(d, n)$ be the polynomial time in n to solve integer linear programming time with the fixed dimensional number d by Theorem 5, where n is the total number of linear inequalities from input polyhedra.

3. Algorithms about Polyhedra Circuits

In this section, we give the definition of polyhedra circuits and develop algorithms to compute the volume of a region given by a polyhedra circuit and to count the number of lattice points inside a region given by a polyhedra circuit. We always assume that the linear inequalities that we deal with are linear inequalities in \mathbb{R}^d and the coefficient of each linear inequalities are integers throughout the paper, which means the polyhedra that we deal with are rational polyhedra throughout the paper.

Definition 7. A *hyper plane* in \mathbb{R}^d can be defined by a linear equation of the form

$$a_1x_1 + a_2x_2 + \cdots + a_dx_d = b,$$

where a_1, a_2, \dots, a_d and b are constants.

Definition 8. Define $r_d(n)$ to be the maximal number of regions that n hyper planes partition \mathbb{R}^d .

Lemma 9 gives the lower bound and upper bound of the maximal number $r_d(n)$ of regions that n hyper planes partition \mathbb{R}^d .

Lemma 9. $(\lfloor \frac{n}{d} \rfloor)^d \leq r_d(n) \leq \frac{n^d}{d!} + (n+1)^{d-1}$ regions.

Proof: We have the recursion $r_{d+1}(n+1) = r_{d+1}(n) + r_d(n)$. This is because the $n+1$ -th d -dimensional hyper plane can have at most $r_d(n)$ regions in it, and increases the number of regions by at most $r_d(n)$. It is trivial $r_1(n) \leq n+1$, and also trivial for $n=0$ or 1 for all $d \geq 1$. It follows a simple induction. Assume that $r_d(n) \leq \frac{n^d}{d!} + (n+1)^{d-1}$ and $r_{d+1}(n) \leq \frac{n^{d+1}}{(d+1)!} + (n+1)^d$.

$$\begin{aligned}
 r_{d+1}(n+1) &= r_{d+1}(n) + r_d(n) \\
 &\leq \left(\frac{n^{d+1}}{(d+1)!} + (n+1)^d \right) + \left(\frac{n^d}{d!} + (n+1)^{d-1} \right) \\
 &= \frac{n^{d+1} + (d+1)n^d}{(d+1)!} + ((n+1)^d + (n+1)^{d-1}) \\
 &< \frac{(n+1)^{d+1}}{(d+1)!} + (n+2)^d.
 \end{aligned}$$

For the left side of the inequalities, we just let there are $\frac{n}{d}$ hyper planes to be parallel to each of d axis in \mathbb{R}^d . Therefore, we have the lower bound $(\lfloor \frac{n}{d} \rfloor)^d$. ■

Definition 10. For a set of linear inequalities in \mathbb{R}^d , a region is *atomic* if it is formed by a subset of linear inequalities, and does not contain any proper subregion that can be formed by another subset of linear inequalities.

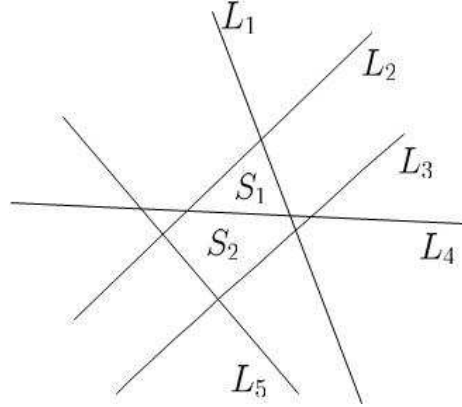


Figure 1: Example of atomic regions with S_1 and S_2 are atomic regions, but $S_1 \cup S_2$ is not atomic region, although it is formed by a subset linear inequalities.

Definition 11. A *polyhedra circuit* consists is a circuit that consists of multiple layers of gates:

- Each input gate is a linear inequality of format $\sum_{i=1}^d a_i x_u \leq b$ or $\sum_{i=1}^d a_i x_u < b$ (represent a half space in \mathbb{R}^d), where b, a_1, \dots, a_d are integers,
- Each internal gate is either union or intersection operation, and
- The only output gate is on the top of the circuit.

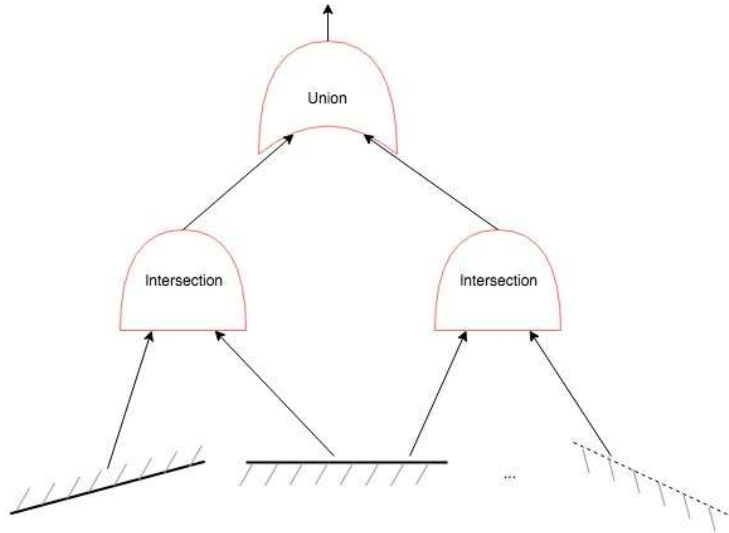


Figure 2: Example of a Polyhedra Circuit

For example, the expression $((-x < -1) \wedge (x \leq 3)) \vee ((-x \leq -2) \wedge (x \leq 5))$ is \mathbb{R}^1 polyhedra circuit to express to the union of $(1, 3]$ and $[2, 5]$. Its output is the region $(1, 5]$ in \mathbb{R}^1 .

Note that we do not consider the negation operation in our circuit gates, since a negation operation in a gate can be handled by taking the negation of the input linear inequalities via De Morgan's laws. We note also that a union of multiple polyhedra can produce very complicate shape in a d -dimensional space for $d \geq 2$.

Lemma 12 shows that the output of a polyhedra circuit is a disjoint union of atomic regions.

Lemma 12. *The region generated by a polyhedra circuit is a disjoint union of atomic regions via the hyper planes of the input polyhedra.*

Proof: It can be proven via an induction on the depth of the circuits. It is trivial when the circuits has depth zero. Assume that there are G_1, \dots, G_k gates in depth k , and each G_i generates a region to be the union of atomic regions for $i = 1, \dots, k$. We need to show that each gate G_j in depth $k + 1$ generates a region to be the union of atomic regions.

The union or intersection of the union of atomic regions also be the union of atomic regions. Every gate is either union or intersection via Definition 11, therefore, every gate G_j in depth $k + 1$ generates a region to be the union of atomic regions.

Assume that G is the output gate. By the induction, we can say that region from G is also the union of atomic regions. ■

Lemma 13 shows that the volume of rational polyhedra formed by $Ax < b$ equals the volume of rational polyhedra formed by $Ax \leq b$.

Lemma 13. *The volume of atomic region formed by $Ax < b$ equals the volume of atomic region formed by $Ax \leq b$, where A is a given $n \times d$ matrix, b is an n -dimensional vector*

Proof: Since the boundaries of the atomic region formed by $Ax \leq b$ have no volume, then the volume of atomic region that is formed by $Ax < b$ equals the volume of atomic region that is formed by $Ax \leq b$, where A is a given $n \times d$ matrix and b is an n -dimensional vector. ■

Lemma 14 shows that the number of lattice points in rational polyhedra formed by $Ax < b$ equals the the number of lattice points in rational polyhedra formed by $Ax \leq b - I$, where all the elements of $n \times d$ matrix A and n -dimensional vector b are inegers, and I is n -dimensional vector with all 1s.

Lemma 14. *The number of lattice points in atomic region formed by $Ax < b$ equals the number of lattice points in atomic region formed by $Ax \leq c$, where A is a given $n \times d$ matrix with integer elements, b is an n -dimensional vector with integer elements and c is an n -dimensional vector with $c = b - I$ for n -dimensional vector I whose elements are all 1s.*

Proof: The number of lattice points in linear inequality $a_1x_1 + \dots + a_dx_d < b_1$ is the same as the number of lattice points in linear inequality $a_1x_1 + \dots + a_dx_d < b_1 - 1$, becasue a_1, \dots, a_d and b_1 are integers and each x_i will take an integer value. Therefore, the number of lattice points in atomic region formed by $Ax < b$ equals the number of lattice points in atomic region formed by $Ax \leq c$, where A is a given $n \times d$ matrix with integer elements, b is an n -dimensional vector with integer elements and c is an n -dimensional vector with $c = b - I$ for n -dimensional vector I whose elements are all 1s. ■

Lemma 15 shows that a list of linear inequalities partition space \mathbb{R}^d into all of the atomic regions with an interior point in each atomic region.

Lemma 15. *There is a $O(nd \cdot r_d(n) \cdot T_V(d, n))$ time algorithm such that given a list of n linear inequalities in \mathbb{R}^d , it produces all of the atomic regions with one interior point from each of them, where $T_V(\cdot)$ is the time to compute the volume of one rational polyhedra and $r_d(n)$ be the maximal number of regions that n hyper planes partition \mathbb{R}^d .*

Proof: It can be proven via an induction method. Let L_1, L_2, \dots, L_n be the linear inequalities. If there is only one linear inequality, say L_1 , L_1 partitions \mathbb{R}^d into two atomic regions S_1 and S_2 , where S_1 is formed by L_1 and S_2 is formed by $\neg L_1$, it is easy to generate two interior points $p_1 \in S_1$ and $p_2 \in S_2$.

Assume that we have obtained atomic regions S_1, S_2, \dots, S_t that are formed by L_1, \dots, L_k , and each S_i has an interior point $p_i \in S_i$ for $i = 1, \dots, t$. We need to show that we can obtain all of the atomic regions that are formed by L_1, \dots, L_k, L_{k+1} , and one interior point from each of them.

For each S_i , assume that S_i is formed by hyper planes L'_1, \dots, L'_s with $s \leq k$. Consider L_{k+1} , and the volume of $S'_{i,1}$ formed by adding L_{k+1} to hyper planes L'_1, \dots, L'_s and $S'_{i,2}$ formed by adding $\neg L_{k+1}$ to hyper planes L'_1, \dots, L'_s . Note that we compute the volume of $S'_{i,1}$ and $S'_{i,2}$ by using Theorem 2. Note also that we can compute the volume of $S'_{i,1}$ and $S'_{i,2}$ correctly because of Lemma 13.

We discuss two cases:

Case 1: If the volume of $S'_{i,1}$ or $S'_{i,2}$ equals to 0, then S_i is not separated by L_{k+1} . In this case, the interior point is $p_i \in S_i$.

Case 2: Otherwise, L_{k+1} partitions S_i into two regions $S_{i,1}$ and $S_{i,2}$, and we have to find the interior point for each region. In this case, L_{k+1} intersection S_i with a $(d-1)$ -dimensional atomic region, let us denote as B_{d-1} . Similarly, this $(d-1)$ -dimensional atomic region B_{d-1} can be partitioned into $(d-2)$ -dimensional atomic regions. By using this way, we can easily find a interior point in 0-dimensional atomic region, which is an interior point p_j of B_{d-1} .

Construct line $l = p_j + tv$ perpendicular to L_{k+1} where v is the vector that is perpendicular to L_{k+1} . Find the points p_h, p_z that l intersection of the first hyper plane of $S_{i,1}$ and the first hyper plane of $S_{i,2}$, respectively (See Fig. 3), then the interior point of $S_{i,1}$ is $\frac{p_j + p_h}{2}$, and the interior point of $S_{i,2}$ is $\frac{p_j + p_z}{2}$.

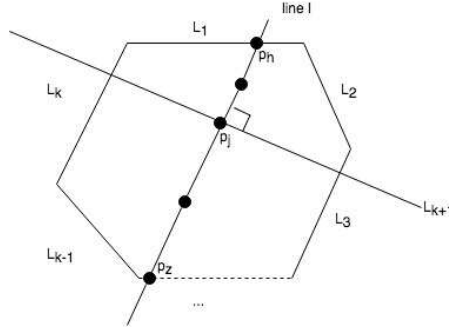


Figure 3

Therefore, we have obtained atomic regions $S_1, S_2, \dots, S_t, \dots, S_h$ that are formed by L_1, \dots, L_k, L_{k+1} for some interger $h > t$, and one interior point from each of them.

Define $T_R(d, n)$ be the running time to generate an interior point in d -

dimensional atomic region S_i with n linear inequalities. Then, we have

$$\begin{aligned} T_R(d, k+1) &= 2r_d(k) \cdot T_V(d, k) + T_R(d-1, k) \\ &= 2r_d(k) \cdot T_V(d, k) + \dots + 2r_1(k) \cdot T_V(1, k) \\ &\leq 2d \cdot r_d(k) \cdot T_V(d, k), \end{aligned}$$

and

$$\sum_{k=1}^{n-1} T_R(d, k+1) \leq 2(n-1)d \cdot r_d(k) \cdot T_V(d, k).$$

Therefore, the running of the algorithm is $O(nd \cdot r_d(n) \cdot T_V(d, n))$. ■

Lemma 16 shows that a list of linear inequalities partition space \mathbb{R}^d into all of the atomic regions with one integer point from each of them if exists.

Lemma 16. *There is a $O(n \cdot r_d(n) \cdot T_I(d, n))$ time algorithm such that given a list of n linear inequalities in \mathbb{R}^d , it produces all of the atomic regions, and one integer point from each of them if exists, where $T_I(\cdot)$ is the time to solve integer linear programming problem and $r_d(n)$ be the maximal number of regions that n hyper planes partition \mathbb{R}^d .*

Proof: It can be proven via an induction method. Let L_1, L_2, \dots, L_n be the linear inequalities. If there is only one linear inequality, say L_1 , L_1 partitions \mathbb{R}^d into two atomic regions S_1 and S_2 , where S_1 is formed by L_1 and S_2 is formed by $\neg L_1$, it is easy to generate two integer points $p_1 \in S_1$ and $p_2 \in S_2$.

Assume that we have obtained atomic regions S_1, S_2, \dots, S_t that are formed by L_1, \dots, L_k , and each S_i has an integer point $p_i \in S_i$ for $i = 1, \dots, t$. We need to show that we can obtain all of the atomic regions that are formed by L_1, \dots, L_k, L_{k+1} and one integer point from each of them if exists.

For each S_i , assume that S_i is formed by linear inequalities L'_1, \dots, L'_s with $s \leq k$.

Consider $S_{i,1}$ formed by adding L_{k+1} to linear inequalities L'_1, \dots, L'_s and $S_{i,2}$ formed by adding $\neg L_{k+1}$ to linear inequalities L'_1, \dots, L'_s . We use Theorem 5 to find if there is an integer point in $S_{i,1}$ and $S_{i,2}$. If it exists, we keep the corresponding atomic region.

Therefore, we have obtained atomic regions $S_1, S_2, \dots, S_t, \dots, S_h$ that are formed by L_1, \dots, L_k, L_{k+1} for some integer $h > t$, and one integer point from each of them.

Define $T_R(d, n)$ be the running time to generate an integer point in d -dimensional atomic region S_i with n linear inequalities. Then, we have

$$T_R(d, k+1) = 2r_d(k) \cdot T_I(d, k)$$

and

$$\sum_{k=1}^{n-1} T_R(d, k+1) = 2(n-1) \cdot r_d(k) \cdot T_V(d, k).$$

Therefore, the running of the algorithm is $O(n \cdot r_d(n) \cdot T_I(d, n))$. ■

Theorem 17 shows that the algorithm can compute the volume of the region by the polyhedra circuits.

Theorem 17. *There is a $O(nd \cdot r_d(n) \cdot T_V(d, n))$ time algorithm such that given a polyhedra circuit in \mathbb{R}^d , the algorithm computes the volume of the region determined by the input polyhedra circuit, where $T_V(\cdot)$ is the time to compute the volume of one rational polyhedra, and n is the number of linear inequalities in the input layer of the polyhedra circuit.*

Proof: By Lemma 12 the region generated by a polyhedra circuits is a disjoint union of atomic regions, by Lemma 15 there is an interior point p_i for each atomic region. For each interior point p_i , we check if it is contained by polyhedra circuits, if it does, then add the volume of the atomic regions which can be computed by Theorem 2.

And the running time of the algorithm is

$$O(nd \cdot r_d(n) \cdot T_V(d, n)) + r_d(n)T_V(d, n)$$

via Theorem 2. Therefore the running time $O(nd \cdot r_d(n) \cdot T_V(d, n))$. ■

Theorem 18 shows that the algorithm can count the number of lattice points in the region by the polyhedra circuits.

Theorem 18. *There is a $O(n \cdot r_d(n) \cdot T_I(d, n) + r_d(n)T_L(d, n))$ time algorithm such that given a polyhedra circuit in \mathbb{R}^d , the algorithm counts the number of lattice points in the region determined by the input polyhedra circuit, where $T_L(\cdot)$ is the time to compute the number of lattice points of one rational polyhedra and $T_I(\cdot)$ is the time to solve integer linear programming problem and n is the number of linear inequalities in the input layer of the polyhedra circuit.*

Proof: By Lemma 12 the region generated by a polyhedra circuits is a disjoint union of atomic regions, and by Lemma 16 there is an integer point p_i for each atomic region. For each integer point p_i , we check if it is contained by polyhedra circuits, if it does, then add the number of lattice points in the atomic region which can be computed Theorem 3.

And the running time of the algorithm is

$$O(n \cdot r_d(n) \cdot T_I(d, n)) + r_d(n)T_L(d, n)$$

via Theorem 2.

Therefore the running time $O(n \cdot r_d(n) \cdot T_I(d, n) + r_d(n)T_L(d, n))$. ■

An immediate application of the algorithm is to compute the volume of polyhedra union, and count the number of lattice points in the union of polyhedra.

Corollary 19. *There is a $O(nd \cdot r_d(n) \cdot T_V(d, n))$ time algorithm to compute the volume of the union of polyhedra when give a list of polyhedra, where $T_V(\cdot)$ is the time to compute the volume of one rational polyhedra.*

Proof: Let the output gate of the polyhedra circuits is union (See Fig. 4). Then it follows Theorem 17.

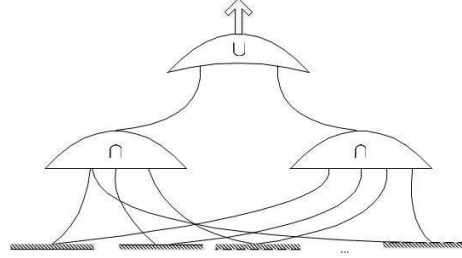


Figure 4

Corollary 20. *There is a $O(n \cdot r_d(n) \cdot T_I(d, n) + r_d(n)T_L(d, n))$ time algorithm to count the number of lattice points in the union of polyhedra when give a list of polyhedra, where $T_L(\cdot)$ is the time to compute the number of lattice points of one rational polyhedra and $T_I(\cdot)$ is the time to find a integer point in one rational polyhedra.*

Proof: Let the output gate of the polyhedra circuits is union (See Fig. 5). Then it follows Theorem 18.

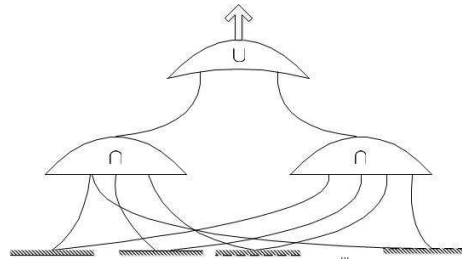


Figure 5

Definition 21. Define $T_{U,L}(d, n)$ be the running time to count the number of lattice points in the union of rational polyhedra by Corollary 20 and $T_{U,V}(d, n)$ be the running time to compute the volume of the union of rational polyhedra by Corollary 19 when given a list of polyhedra where n is the total number of linear inequalities from input polyhedra and d is number of variables of linear inequality.

4. Application in Continuous Polyhedra Maximum Coverage Problem

In this section, we show how to apply the method at Section 3 to the continuous polyhedra maximum coverage problem. Before present the algorithm, we give some definitions about the continuous polyhedra maximum coverage problem.

Definition 22. *Continuous Maximum Coverage Problem:* Given an integer k , a set of regions S_1, S_2, \dots, S_m , select k regions S_{i_1}, \dots, S_{i_k} such that $S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_k}$ has the maximum volume of $S_1 \cup S_2 \cup \dots \cup S_m$.

Definition 23. *Continuous Polyhedra Maximum Coverage Problem:* It is a continuous maximum coverage problem when S_1, \dots, S_m are all polyhedra.

Theorem 24. *There is a $O(km \cdot T_{U,V}(d, n))$ time approximation algorithm for the continuous polyhedra maximum coverage problem with approximation ratio $(1 - \frac{1}{e})$, where $T_{U,V}(d, n)$ be the running time to compute the volume of the union of polyhedra with n be the total number of linear inequalities from input polyhedra.*

Proof: It follows from greedy algorithm and Theorem 17. Let $vol(P)$ denotes the volume of region P and let V denotes $vol(S_1 \cup \dots \cup S_m)$. The greedy method works by picking, at each stage, the algorithm select region S that covers the maximum volume that are uncovered. Assume that t regions have been selected, say, $S_{i_1}, S_{i_2}, \dots, S_{i_t}$, the greedy approach needs to select S_j such that $vol(S_j - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}))$ is maximal for the next choice.

Since

$$vol(S_j - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})) = vol(S_j \cup S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}) - vol(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}),$$

and we can compute $vol(S_j \cup S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})$ and $vol(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})$ via Theorem 17 and Corollary 19, then at each stage, we just need to traversal m polyhedra to find S_j .

Let $V(l)$ be the volume of coverage for l regions, and let OPT be the maximum volume of union of a optimal solution $S_{i_1}, S_{i_2}, \dots, S_{i_k}$ for the continuous polyhedra maximum coverage problem.

First, we show that

$$V(t) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right) OPT$$

by using induction method.

It is trivial at $t = 1$, since $V(1) \geq \frac{OPT}{k}$. Assume $V(t) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right) OPT$. Consider the case $t + 1$.

$$\begin{aligned}
V(t+1) &= V(t) + \text{vol}(S_{i_{t+1}} - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})) \\
&\geq V(t) + \frac{OPT - V(t)}{k} \\
&= \left(1 - \frac{1}{k}\right) V(t) + \frac{OPT}{k} \\
&\geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right) \left(1 - \frac{1}{k}\right) OPT + \frac{OPT}{k} \\
&= \left(1 - \left(1 - \frac{1}{k}\right)^{t+1}\right) OPT.
\end{aligned}$$

Therefore,

$$V(t) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right) OPT.$$

Since $\left(1 - \frac{1}{k}\right)^k$ is increasing and $\frac{1}{e} = \left(1 - \frac{1}{k}\right)^k + \Omega\left(\frac{1}{k}\right)$, therefore,

$$V(k) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) OPT \geq \left(1 - \frac{1}{e}\right) OPT.$$

There are k stages. At each stage, the algorithm needs to check m polyhedra in order to select the next region that covers the maximum volume that are uncovered. By Definition 21, $T_{U,V}(d, n)$ is the running time to compute the volume of the union of polyhedra, therefore the running time of the algorithm is $O(km \cdot T_{U,V}(d, n))$. ■

5. Application in Polyhedra Maximum Lattice Coverage Problem

In this section, we show how to apply the method at Section 3 to the polyhedra maximum lattice coverage problem. Before present the algorithm, we give some definitions about polyhedra maximum lattice coverage problem.

Definition 25. *Maximum Lattice Coverage Problem:* Given an integer k , a set of regions S_1, S_2, \dots, S_m , select k regions S_{i_1}, \dots, S_{i_k} such that $S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_k}$ has the maximum number of lattice points of $S_1 \cup S_2 \cup \dots \cup S_m$.

Definition 26. *Polyhedra Maximum Lattice Coverage Problem:* It is a maximum lattice coverage problem when S_1, \dots, S_m are all polyhedra.

Theorem 27. *There is a $O(km \cdot T_{U,L}(d, n))$ time approximation algorithm for the polyhedra maximum lattice coverage problem with approximation ratio $(1 - \frac{1}{e})$, where $T_{U,L}(d, n)$ is the running time to count the number of lattice points in the union of polyhedra with n be the total number of linear inequalities from input polyhedra.*

Proof: It follows from greedy algorithm and Theorem 18. Let $L(P)$ denotes the volume of region P and let V denotes $L(S_1 \cup \dots \cup S_m)$. The greedy method works by picking, at each stage, the algorithm select region S that covers the maximum volume that are uncovered. Assume that t regions have been selected, say, $S_{i_1}, S_{i_2}, \dots, S_{i_t}$, the greedy approach needs to select S_j such that $L(S_j - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}))$ is maximal for the next choice.

Since

$$\begin{aligned} L(S_j - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})) &= L(S_j \cup S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}) \\ &\quad - L(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}), \end{aligned}$$

and we can compute $L(S_j \cup S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})$ and $L(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})$ via Theorem 18 and Corollary 20, then at each stage, we just need to traversal m polyhedra to find S_j .

Let $V(l)$ be the number of lattice points of coverage for l regions, and let OPT be the maximum number of lattice points of union of a optimal solution $S_{i_1}, S_{i_2}, \dots, S_{i_k}$ for the polyhedra maximum lattice coverage problem.

First, we show that

$$V(t) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right) OPT$$

by using induction method.

It is trivial at $t = 1$, since $V(1) \geq \frac{OPT}{k}$. Assume $V(t) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right) OPT$. Consider the case $t + 1$.

$$\begin{aligned} V(t+1) &= V(t) + L(S_{i_{t+1}} - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})) \\ &\geq V(t) + \frac{OPT - V(t)}{k} \\ &= \left(1 - \frac{1}{k}\right) V(t) + \frac{OPT}{k} \\ &\geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right) \left(1 - \frac{1}{k}\right) OPT + \frac{OPT}{k} \\ &= \left(1 - \left(1 - \frac{1}{k}\right)^{t+1}\right) OPT. \end{aligned}$$

Therefore,

$$V(t) \geq \left(1 - \left(1 - \frac{1}{k}\right)^t\right) OPT.$$

Since $(1 - \frac{1}{k})^k$ is increasing and $\frac{1}{e} = (1 - \frac{1}{k})^k + \Omega(\frac{1}{k})$, therefore,

$$V(k) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) OPT \geq \left(1 - \frac{1}{e}\right) OPT.$$

There are k stages. At each stage, the algorithm needs to check m polyhedra in order to select the next region that covers the maximum number of lattice points that are uncovered. By Definition 21, $T_{U,L}(d, n)$ is the running time to compute the number of lattice points of the union of polyhedra, therefore the running time of the algorithm is $O(km \cdot T_{U,L}(d, n))$. ■

6. Application in Polyhedra $(1 - \beta)$ -Lattice Set Cover Problem

In this section, we show how to apply the method at Section 3 to the polyhedra $(1 - \beta)$ -lattice set cover problem. Before present the algorithm, we give some definitions about polyhedra $(1 - \beta)$ -lattice set cover problem.

Definition 28. *$(1 - \beta)$ -Lattice Set Cover Problem:* For a real $\beta \in [0, 1)$, and a set of regions S_1, S_2, \dots, S_m , select k regions S_{i_1}, \dots, S_{i_k} such that $L(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_k}) \geq (1 - \beta)L(S_1 \cup \dots \cup S_m)$ with $L(P)$ denotes the number of lattice points of region P .

Definition 29. *Polyhedra $(1 - \beta)$ -Lattice Set Cover Problem:* It is a lattice set cover problem when S_1, \dots, S_m are all polyhedra.

Theorem 30. *For reals $\alpha \in (0, 1)$ and $\beta \in [0, 1)$, there is a $O(m^2 \cdot T_{U,L}(d, n))$ time approximation algorithm for the polyhedra $(1 - \beta)$ -lattice set cover problem with the output k regions S_{i_1}, \dots, S_{i_k} satisfying $L(S_{i_1} \cup \dots \cup S_{i_k}) \geq (1 - \alpha)(1 - \beta)L(S_1 \cup \dots \cup S_m)$ and $k \leq (1 + \ln \frac{1}{\alpha})H$, where H is the number of sets in an optimal solution for the polyhedra $(1 - \beta)$ -lattice set cover problem, and $T_{U,L}(d, n)$ is the running time to count the number of lattice points in the union of polyhedra with n be the total number of linear inequalities from input polyhedra.*

Proof: It follows from greedy algorithm and Theorem 17. Let $L(P)$ denotes the number of lattice points of region P and let V denotes $L(S_1 \cup \dots \cup S_m)$. The greedy method works by picking, at each stage, the algorithm select region S that covers the maximum number of lattice points that are uncovered. Assume that t regions have been selected, say, $S_{i_1}, S_{i_2}, \dots, S_{i_t}$, the greedy approach needs to select S_j such that $L(S_j - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}))$ is maximal for the next choice. The algorithm stops if $L(S_{i_1} \cup \dots \cup S_{i_k}) \geq (1 - \alpha)(1 - \beta)L(S_1 \cup \dots \cup S_m)$.

Since

$$\begin{aligned} L(S_j - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})) &= L(S_j \cup S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}) \\ &\quad - L(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}), \end{aligned}$$

and we can compute $L(S_j \cup S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})$ and $L(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})$ via Theorem 18 and Corollary 20, then at each stage, we just need to traversal m polyhedra to find S_j .

First, we show that

$$L(S_{i_1} \cup \dots \cup S_{i_t}) \geq \left(1 - \left(1 - \frac{1}{H}\right)^t\right) (1 - \beta) V$$

by using induction method.

It is trivial at $t = 1$, since $L(S_{i_1}) \geq \frac{(1-\beta)V}{H} = \left(1 - \left(1 - \frac{1}{H}\right)\right) (1 - \beta) V$. Assume $L(S_{i_1} \cup \dots \cup S_{i_t}) \geq \left(1 - \left(1 - \frac{1}{H}\right)^t\right) (1 - \beta) V$. Consider the case $t + 1$.

$$\begin{aligned} L(S_{i_1} \cup \dots \cup S_{i_t} \cup S_{i_{t+1}}) &= L(S_{i_1} \cup \dots \cup S_{i_t}) \\ &\quad + L(S_{i_{t+1}} - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})) \\ &\geq L(S_{i_1} \cup \dots \cup S_{i_t}) + \frac{(1 - \beta) V - L(S_{i_1} \cup \dots \cup S_{i_t})}{H} \\ &= \left(1 - \frac{1}{H}\right) L(S_{i_1} \cup \dots \cup S_{i_t}) + \frac{(1 - \beta) V}{H} \\ &\geq \left(1 - \left(1 - \frac{1}{H}\right)^t\right) \left(1 - \frac{1}{H}\right) (1 - \beta) V + \frac{(1 - \beta) V}{H} \\ &= \left(1 - \left(1 - \frac{1}{H}\right)^{t+1}\right) (1 - \beta) V. \end{aligned}$$

Therefore,

$$L(S_{i_1} \cup \dots \cup S_{i_t}) \geq \left(1 - \left(1 - \frac{1}{H}\right)^t\right) (1 - \beta) V. \quad (1)$$

For each k satisfying

$$\left(1 - \frac{1}{H}\right)^k \leq \alpha \quad (2)$$

then it will satisfy inequality (1).

Since

$$\left(1 - \frac{1}{H}\right)^k = \left(1 - \frac{1}{H}\right)^{H \cdot \frac{k}{H}} < \left(\frac{1}{e}\right)^{\frac{k}{H}},$$

then we have k as

$$k \geq H \cdot \ln \frac{1}{\alpha}$$

that will satisfy the inequality (2), which means it will also satisfy inequality (1).

We can let

$$k = \left\lceil H \cdot \ln \frac{1}{\alpha} \right\rceil \leq H \cdot \ln \frac{1}{\alpha} + 1 \leq H \left(1 + \ln \frac{1}{\alpha} \right).$$

Therefore,

$$k \leq \left(1 + \ln \frac{1}{\alpha} \right) H.$$

The algorithm has to traversal m polyhedra. At each stage, the algorithm needs to check m polyhedra in order to select the next region that covers the maximum number of lattice points that are uncovered. By Definition 21, the running time to compute the number of lattice points of the union of polyhedra is $O(T_{U,L}(d, n))$, therefore, the running time of the algorithm is $O(m^2 \cdot T_{U,L}(d, n))$. \blacksquare

7. Application in $(1 - \beta)$ -Continuous Polyhedra Set Cover

In this section, we show how to apply the method at Section 3 to $(1 - \beta)$ -continuous polyhedra set cover problem. Before present the algorithm, we give some definitions about $(1 - \beta)$ -continuous polyhedra set cover problem.

Definition 31. *$(1 - \beta)$ -Continuous Set Cover Problem:* For a real $\beta \in [0, 1)$, and a set of regions S_1, S_2, \dots, S_m , select k regions S_{i_1}, \dots, S_{i_k} such that $\text{vol}(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_k}) \geq (1 - \beta)\text{vol}(S_1 \cup \dots \cup S_m)$ with $\text{vol}(P)$ denotes the volume of region P .

Definition 32. *$(1 - \beta)$ -Continuous Polyhedra Set Cover Problem:* It is a $(1 - \beta)$ -continuous set cover problem when S_1, \dots, S_m are all polyhedra.

Theorem 33. *For reals $\alpha \in (0, 1)$ and $\beta \in [0, 1)$, there is a $O(m^2 \cdot T_{U,V}(d, n))$ time approximation algorithm for the $(1 - \beta)$ -continuous polyhedra set cover problem with the output k regions S_{i_1}, \dots, S_{i_k} satisfying $\text{vol}(S_{i_1} \cup \dots \cup S_{i_k}) \geq (1 - \alpha)(1 - \beta)\text{vol}(S_1 \cup \dots \cup S_m)$ and $k \leq \left(1 + \ln \frac{1}{\alpha} \right) H$, where H is the number of sets in an optimal solution for the $(1 - \beta)$ -continuous set cover problem, and $T_{U,V}(d, n)$ is the running time to compute the volume of the union of polyhedra with n be the total number of linear inequalities from input polyhedra.*

Proof: It follows from greedy algorithm and Theorem 17. Let $vol(P)$ denotes the volume of region P and let V denotes $vol(S_1 \cup \dots \cup S_m)$. The greedy method works by picking, at each stage, the algorithm select region S that covers the maximum volume that are uncovered. Assume that t regions have been selected, say, $S_{i_1}, S_{i_2}, \dots, S_{i_t}$, the greedy approach needs to select S_j such that $vol(S_j - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}))$ is maximal for the next choice. The algorithm stops if $vol(S_{i_1} \cup \dots \cup S_{i_k}) \geq (1 - \alpha)(1 - \beta) vol(S_1 \cup \dots \cup S_m)$.

Since

$$vol(S_j - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})) = vol(S_j \cup S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}) - vol(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t}),$$

and we can compute $vol(S_j \cup S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})$ and $vol(S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})$ via Theorem 17 and Corollary 19, then at each stage, we just need to traversal m polyhedra to find S_j .

First, we show that

$$vol(S_{i_1} \cup \dots \cup S_{i_t}) \geq \left(1 - \left(1 - \frac{1}{H}\right)^t\right) (1 - \beta) V$$

by using induction method.

It is trivial at $t = 1$, since $vol(S_{i_1}) \geq \frac{(1-\beta)V}{H} = \left(1 - \left(1 - \frac{1}{H}\right)\right) (1 - \beta) V$. Assume $vol(S_{i_1} \cup \dots \cup S_{i_t}) \geq \left(1 - \left(1 - \frac{1}{H}\right)^t\right) (1 - \beta) V$. Consider the case $t+1$.

$$\begin{aligned} vol(S_{i_1} \cup \dots \cup S_{i_t} \cup S_{i_{t+1}}) &= vol(S_{i_1} \cup \dots \cup S_{i_t}) \\ &\quad + vol(S_{i_{t+1}} - (S_{i_1} \cup S_{i_2} \cup \dots \cup S_{i_t})) \\ &\geq vol(S_{i_1} \cup \dots \cup S_{i_t}) + \frac{(1 - \beta) V - vol(S_{i_1} \cup \dots \cup S_{i_t})}{H} \\ &= \left(1 - \frac{1}{H}\right) vol(S_{i_1} \cup \dots \cup S_{i_t}) + \frac{(1 - \beta) V}{H} \\ &\geq \left(1 - \left(1 - \frac{1}{H}\right)^t\right) \left(1 - \frac{1}{H}\right) (1 - \beta) V + \frac{(1 - \beta) V}{H} \\ &= \left(1 - \left(1 - \frac{1}{H}\right)^{t+1}\right) (1 - \beta) V. \end{aligned}$$

Therefore,

$$vol(S_{i_1} \cup \dots \cup S_{i_t}) \geq \left(1 - \left(1 - \frac{1}{H}\right)^t\right) (1 - \beta) V. \quad (3)$$

For each k satisfying

$$\left(1 - \frac{1}{H}\right)^k \leq \alpha \quad (4)$$

then it will satisfy inequality (3).

Since

$$\left(1 - \frac{1}{H}\right)^k = \left(1 - \frac{1}{H}\right)^{H \cdot \frac{k}{H}} < \left(\frac{1}{e}\right)^{\frac{k}{H}},$$

then we have k as

$$k \geq H \cdot \ln \frac{1}{\alpha}$$

that will satisfy the inequality (4), which means it will also satisfy inequality (3).

We can let

$$k = \left\lceil H \cdot \ln \frac{1}{\alpha} \right\rceil \leq H \cdot \ln \frac{1}{\alpha} + 1 \leq H \left(1 + \ln \frac{1}{\alpha}\right).$$

Therefore,

$$k \leq \left(1 + \ln \frac{1}{\alpha}\right) H.$$

The algorithm has to traversal m polyhedra. At each stage, the algorithm needs to check m polyhedra in order to select the next region that covers the maximum volume that are uncovered. By Definition 21, the running time to compute the volume of the union of polyhedra is $O(T_{U,V}(d, n))$, therefore, the running time of the algorithm is $O(m^2 \cdot T_{U,V}(d, n))$. ■

8. NP-hardness and Inapproximation

In this section, we show the NP-hardness of the maximum coverage problem and set cover problem when each set is represented as union of polyhedrons.

Definition 34. The maximum region coverage problem is that given a list A_1, \dots, A_m of regions in \mathbb{R}^d , and an integer $k \geq 1$. Find k regions with the largest volume for their union.

Definition 35. Let A_1, \dots, A_k be a list axis parallel rectangles. Let A be the region to be union $A_1 \cup \dots \cup A_k$. Then A is called an *axis parallel rectangle union region*.

Definition 36. The maximum region coverage problem with each set to be axis parallel rectangle union region is called *axis parallel rectangle union maximum region coverage problem*.

Definition 37. The lattice set cover problem with each set to be all lattice points in axis parallel rectangle union region is called *lattice axis parallel rectangle union maximum coverage problem*.

Theorem 38. *The following statements are true:*

- i. The axis parallel rectangle union maximum region coverage problem is NP-hard. Furthermore, if there is a polynomial time c -approximation for axis parallel rectangle union maximum region coverage problem, then there is a polynomial time c -approximation for classical maximum coverage problem.*
- ii. The axis parallel rectangle union set problem is NP-hard. Furthermore, if there is a polynomial time c -approximation for lattice axis parallel rectangle union region set cover problem, then there is a polynomial time c -approximation for classical set cover problem.*

Proof: Let S_1, \dots, S_m be an input for a (classical) maximum coverage problem or set cover problem. Let U be $S_1 \cup S_2 \cup \dots \cup S_m$. We construct a axis parallel rectangle union maximum region coverage problem. Without loss of generality, we assume $U = \{1, 2, \dots, n\}$. For each S_i , we construct A_i , which is an axis parallel rectangle region. Let D represent the line segment of length $2n$ from $(0, \frac{1}{2})$ to $(2n-1, \frac{1}{2})$. Let B_k be the 1×1 square with left bottom corner at $(2(i-1), 0)$. Let $A_i = D \cup (\cup_{j \in S_i} B_j)$. It is easy to see that if $|S_{i_1} \cup \dots \cup S_{i_k}| = h$ if and only if the area of $A_{i_1} \cup \dots \cup A_{i_k}$ is h .

If $S_{i_1} \cup \dots \cup S_{i_k} = S_1 \cup \dots \cup S_m$ if and only if the area of $A_{i_1} \cup \dots \cup A_{i_k}$ is n . It is easy to see to this reduction reserves the ratio of approximation for both maximum coverage and set cover problems. ■

Using reasonable hypothesis of complexity theory, Feige [16] showed that $1 - \frac{1}{e}$ is the best possible polynomial time approximation ratio for the maximum coverage problem, and $\ln n$ is the best possible polynomial time approximation ratio for the set cover problem, where n is the number of elements in the universe set for the classical set cover. Our Theorem 38 shows the connection of inapproximation between the classical problems and their corresponding geometric versions.

9. Conclusions

We introduce polyhedra circuits. Each polyhedra circuit characterizes a geometric region in \mathbb{R}^d . They can be applied to represent a rich class of geometric objects, which include all polyhedra and the union of a finite number of polyhedra. They can be used to approximate a large class of d -dimensional manifolds in \mathbb{R}^d . Define $T_V(d, n)$ be the polynomial time in n to compute the volume of one rational polyhedra, $T_L(d, n)$ be the polynomial time in n to count the number of lattice points in one rational polyhedra with d be a fixed dimensional number, $T_I(d, n)$ be the polynomial time in n to solve integer linear programming time when the fixed dimensional number d is small, where n is the total number of linear inequalities from input polyhedra. We develop algorithms to count the number of lattice points in the geometric region determined by a polyhedra circuit in $O(nd \cdot r_d(n) \cdot T_V(d, n))$ time and to

compute the volume of the geometric region determined by a polyhedra circuit in $O(n \cdot r_d(n) \cdot T_I(d, n) + r_d(n)T_L(d, n))$ time, where n is the number of input linear inequalities, d is number of variables and $r_d(n)$ be the maximal number of regions that n linear inequalities with d variables partition \mathbb{R}^d . Define $T_{U,L}(d, n)$ be the running time to count the number of lattice points in the union of polyhedra and $T_{U,V}(d, n)$ be the running time to compute the volume of the union of polyhedra when given a list of polyhedra where n is the total number of linear inequalities from input polyhedra and d is number of variables. We applied the algorithms to continuous polyhedra maximum coverage problem and polyhedra maximum lattice coverage problem. The algorithm for continuous polyhedra maximum coverage problem gives $(1 - \frac{1}{e})$ -approximation in $O(km \cdot T_{U,V}(d, n))$ running time and the algorithm for polyhedra maximum lattice coverage problem gives $(1 - \frac{1}{e})$ -approximation in $O(km \cdot T_{U,L}(d, n))$ running time. We also applied the algorithms to polyhedra $(1 - \beta)$ -lattice set cover problem and $(1 - \beta)$ -continuous polyhedra set cover problem. The algorithm for polyhedra $(1 - \beta)$ -lattice set cover problem such that the optimal k regions cover $(1 - \alpha)(1 - \beta)$ of the number of lattice points V of region $S_1 \cup \dots \cup S_m$ with $k \leq (1 + \ln \frac{1}{\alpha}) H$ in $O(m^2 \cdot T_{U,L}(d, n))$ time, where H is the number of sets in an optimal solution for the polyhedra $(1 - \beta)$ -lattice set cover problem, and the algorithm for $(1 - \beta)$ -continuous polyhedra set cover problem such that the optimal k regions cover $(1 - \alpha)(1 - \beta)$ of the volume V of region $S_1 \cup \dots \cup S_m$ with $k \leq (1 + \ln \frac{1}{\alpha}) H$ in $O(m^2 \cdot T_{U,V}(d, n))$ running time, where H is the number of sets in an optimal solution for the $(1 - \beta)$ -continuous set cover problem. We also showed the NP-hardness of the continuous maximum coverage problem and set cover problem when each set is represented as union of polyhedra.

References

- [1] E. L. Allgower and P. M. Schmidt. Computing volumes of polyhedra. *Math. Comp.*, 46:171–174, 1986.
- [2] W. Baldoni-Silva and M. Vergne. Residues formulae for volumes and ehrhart polynomials of convex polytopes. arXiv:math/0103097, 2001.
- [3] A. I. Barvinok. Computing the volume, counting integral points, and exponential sums. *Discrete Comput. Geom.*, 10:123–141, 1993.
- [4] A. I. Barvinok. A polynomial time algorithm for counting lattice points in polyhedra when the dimension is fixed. *Math. Oper. Res.*, 19:769–779, 1994.
- [5] A. I. Barvinok and J. Pommersheim. An algorithmic theory of lattice points in polyhedra. new perspectives in algebraic combinatorics (berkeley, ca, 1996/97). *Math. Sci. Res. Inst. Publ. Cambridge Univ. Press, Cambridge*, 38:91–147, 1999.

- [6] M. Beck. Counting lattice points by means of the residue theorem. *Ramanujan J.*, 4(3):299–310, 2000.
- [7] M. Beck and D. Pixton. The ehrhart polynomial of the birkhoff polytope. *Discrete Comput. Geom.*, 30(4):623–637, 2003.
- [8] M. Brion. Points entiers dans les polyèdres convexes. *Ann. Sci. École Norm. Sup.*, 21:653–663, 1988.
- [9] M. Brion. Polyèdres et réseaux. *L’Enseignement Mathématique*, 38:71–88, 1992.
- [10] K. L. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*, 42(2):488–499, 1995.
- [11] J. Cohen and T. Hickey. Two algorithms for determining volumes of convex polyhedra. *J. Assoc. Comput. Mach.*, 26:401–414, 1979.
- [12] M. Dyer and A. M. Frieze. On the complexity of computing the volume of a polyhedron. *SIAM J. Comput.*, 17(5):967–974, 1988.
- [13] M. Dyer and R. Kannan. On barvinok’s algorithm for counting lattice points in fixed dimension. *Math. Oper. res.*, 22:545–549, 1997.
- [14] M. E. Dyer. On counting lattice points in polyhedra. *SIAM Journal on Computing*, 20:695–707, 1991.
- [15] E. Ehrhart. Sur un problème de géométrie diophantienne linéaire. i. polyèdres et réseaux. *J. Reine Angew. Math.*, 226:1–29, 1967.
- [16] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
- [17] B. V. Hohenbalken. Finding simplicial subdivisions of polytopes. *Math. Programming*, 21:233–234, 1981.
- [18] L. G. Khachiyan. The problem of calculating the volume of a polyhedron is enumerably hard (in russian). *Uspekhi Mat. Nauk*, 44(3):179–180, 1989.
- [19] J. B. Lasserre. n analytical expression and an algorithm for the volume of a convex polyhedron in \mathbb{R}^n . *J. Optim. Theory Appl.*, 39:363–377, 1983.
- [20] J. B. Lasserre and E. S. Zeron. Solving the knapsack problem via z -transform. *Oper. Res. Lett.*, 30:394–400, 2002.
- [21] J. Lawrence. Polytope volume computation. *Math. Comp.*, 57:259–271, 1991.
- [22] J. A. D. Loera, R. Hemmecke, J. Tauzer, and R. Yoshida. Effective lattice point counting in rational convex polytopes. *J. Symbolic Comput.*, 38:1273–1302, 2004.

- [23] P. A. MacMahon. *Combinatorial Analysis*. Chelsea, 1960.
- [24] R. Pemantle and M. Wilson. Asymptotics of multivariate sequences. i. smooth points of the singular variety. *J. Combin. Theory Ser. A*, 97:129–161, 2001.
- [25] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, New York, 1986.
- [26] R. P. Stanley. *Enumerative Combinatorics*. Cambridge, 1997.