Computational Geometry 43 (2010) 601–610

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Approximating the volume of unions and intersections of high-dimensional geometric objects ☆

Karl Bringmann [a], Tobias Friedrich [b,*]

[a] Saarland University, Saarbrücken, Germany
[b] Max-Planck-Institut für Informatik, Saarbrücken, Germany

ARTICLE INFO

Article history:
Received 27 October 2008
Received in revised form 15 December 2009
Accepted 25 March 2010
Available online 27 March 2010
Communicated by K. Fukuda

Keywords:
Union of geometric objects
Intersection of geometric objects
Boxes
Klee's measure problem
Volume
Measure
Approximation

ABSTRACT

We consider the computation of the volume of the union of high-dimensional geometric objects. While showing that this problem is **#P**-hard already for very simple bodies, we give a fast FPRAS for all objects where one can (1) test whether a given point lies inside the object, (2) sample a point uniformly, and (3) calculate the volume of the object in polynomial time. It suffices to be able to answer all three questions approximately. We show that this holds for a large class of objects. It implies that Klee's measure problem can be approximated efficiently even though it is **#P**-hard and hence cannot be solved exactly in polynomial time in the number of dimensions unless **P** = **NP**. Our algorithm also allows to efficiently approximate the volume of the union of convex bodies given by weak membership oracles.

For the analogous problem of the intersection of high-dimensional geometric objects we prove **#P**-hardness for boxes and show that there is no multiplicative polynomial-time $2^{d^{1-\varepsilon}}$-approximation for certain boxes unless **NP** = **BPP**, but give a simple additive polynomial-time ε-approximation.

1. Introduction

Given n bodies in the d-dimensional space, how efficiently can we compute the volume of the union and the intersection? We consider this basic geometric problem for different kinds of bodies. The tractability of this problem highly depends on the representation and the complexity of the given objects. For many classes of objects already computing the volume of one body can be hard. For example, calculating the volume of a polytope given either as a list of vertices or as a list of facets is **#P**-hard [11,19]. For convex bodies given by a membership oracle one can also show that even though there can be no deterministic $(\mathcal{O}(1)d/\log d)^d$-approximation for $d \geqslant 2$ [4], one can still approximate the volume by an FPRAS (fully polynomial-time randomized approximation scheme). In a seminal paper Dyer, Frieze, and Kannan [12] gave an $\mathcal{O}^*(d^{23})$ algorithm, which was subsequently improved in a series of papers [2,16,21,22] to $\mathcal{O}^*(d^4)$ [23] (where the asterisk hides powers of the approximation ratio and $\log d$).

Volume computation of unions can be hard not only for bodies whose volume is hard to calculate. One famous example for this is *Klee's Measure Problem* (KMP). Given n axis-parallel boxes in the d-dimensional space (d constant), the problem asks for the measure of their union. In 1977, Victor Klee showed that it can be solved in time $\mathcal{O}(n \log n)$ for $d = 1$ [20]. This was generalized to $d > 1$ dimensions by Bentley [6] in the same year. He presented an algorithm

☆ A preliminary conference version of this paper appeared in Bringmann and Friedrich (2008) [7].

* Corresponding author.
E-mail address: tobias.friedrich@mpi-inf.mpg.de (T. Friedrich).

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2010.03.004

which runs in $\mathcal{O}(n^{d-1}\log n)$, which was later improved by van Leeuwen and Wood [28] to $\mathcal{O}(n^{d-1})$. In 1988, Overmars and Yap [25] obtained an $\mathcal{O}(n^{d/2}\log n)$ algorithm. This was the fastest algorithm for $d \geqslant 3$ until very recently Chan [9] presented a slightly improved version of Overmars and Yap's algorithm that runs in time $n^{d/2}2^{\mathcal{O}(\log^* n)}$, where $\log^*$ denotes the iterated logarithm. So far, the only known lower bound is $\Omega(n\log n)$ for any d [13]. Chan [9] also proves that no algorithm of runtime $n^{o(d)}$ is possible assuming $\mathbf{W[1]} \neq \mathbf{FPT}$, which is a weaker result than $\mathbf{P} \neq \mathbf{NP}$ and a commonly accepted conjecture on fixed-parameter tractability. Note that the worst-case combinatorial complexity (i.e., the number of faces of all dimensions on the boundary of the union) of $\Theta(n^d)$ does not imply any bounds on the computational complexity. There are various algorithms for special cases, e.g., for hypercubes [1,17] and unit hypercubes [8]. In this paper we explore the opposite direction and examine the union of more general geometric objects.

1.1. Our results

It is not hard to see that KMP is **#P**-hard (see Theorem 1). Hence it cannot be solved in time polynomial in the number of dimensions unless $\mathbf{P} = \mathbf{NP}$. This shows that *exact* volume computation of unions is intractable for all classes of bodies that contain axis-parallel boxes. This motivates the development of *approximation algorithms* for the volume computation of unions. Based on an FPRAS for #DNF by Karp, Luby, and Madras [18], we give an FPRAS for a large class of bodies including boxes, spheres, polytopes, convex bodies determined by an oracle, and schlicht domains. Additionally, also fixed affine transformations of the fore-mentioned objects can be allowed. The underlying bodies B just have to support the following oracle queries in polynomial time:

- PointQuery(x, B): Is point $x \in \mathbb{R}^d$ an element of body B?
- VolumeQuery(B): What is the volume of body B?
- SampleQuery(B): Return a random uniformly distributed point $x \in B$.

PointQuery is a very natural condition which is fulfilled in almost all practical cases. The VolumeQuery condition is important as it could be the case that no efficient approximation of the volume of one of the bodies itself is possible. This, of course, prevents an efficient approximation of the union of such bodies. The SampleQuery is crucial for our FPRAS. In Section 2.3 we will show that it is efficiently computable for a wide range of bodies.

An important feature of our algorithm is that it suffices that all three oracles are weak. More precisely, we allow the following *relaxation* for every body B ($\mathrm{VOL}(B)$ denotes the volume of a body B in the standard Lebesgue measure on $\mathbb{R}^d$, more details are given in Section 2):

- PointQuery(x, B) answers true if and only if $x \in B'$ for a fixed $B' \subset \mathbb{R}^d$ with $\mathrm{VOL}((B' \setminus B) \cup (B \setminus B')) \leqslant \varepsilon_P \mathrm{VOL}(B)$.
- VolumeQuery(B) returns a value V' with $(1-\varepsilon_V)\mathrm{VOL}(B) \leqslant V' \leqslant (1+\varepsilon_V)\mathrm{VOL}(B)$.
- SampleQuery(B) returns only an *almost* uniformly distributed random point [16], that is, it suffices to get a random point $x \in B'$ (with B' as above) such that for the probability density f we have for every point x:

$$\left|f(x) - 1/\mathrm{VOL}(B')\right| < \varepsilon_S.$$

Let $P(d)$ be the worst PointQuery runtime[1] of any of our bodies, analogously $V(d)$ for VolumeQuery, and $S(d)$ for SampleQuery. Then our FPRAS has a runtime of $\mathcal{O}(nV(d) + \frac{n}{\varepsilon^2}(S(d) + P(d)))$ for producing an ε-approximation[2] with probability $\geqslant \frac{3}{4}$ if the errors of the underlying oracles are small, i.e., $\varepsilon_S, \varepsilon_V \leqslant \frac{\varepsilon^2}{47n}$ and $\varepsilon_P \leqslant \frac{\varepsilon^2}{47n^2}$. For example for boxes (e.g., for KMP), this reduces to $\mathcal{O}(\frac{dn}{\varepsilon^2})$ and is the first FPRAS for this problems. In Section 2.3 we also show that our algorithm is an FPRAS for the computation of the volume of the union of convex bodies.

The canonical next question is the computation of the volume of the *intersection of bodies* in $\mathbb{R}^d$. It is clear that most of the problems from above apply to this question, too. **#P**-hardness for general, i.e., not necessarily axis-parallel, boxes follows directly from the hardness of computing the volume of a polytope [11,19]. This leaves open whether there are efficient approximation algorithms for the volume of intersection. In Section 3 we show that there cannot be a (deterministic or randomized) multiplicative $2^{d^{1-\varepsilon}}$-approximation in general, unless $\mathbf{NP} = \mathbf{BPP}$ by identifying a hard subproblem. Instead we give an additive ε-approximation, which is therefore the best we can hope for. It has a runtime of $\mathcal{O}(nV(d) + \varepsilon^{-2}S(d) + n\varepsilon^{-2}P(d))$, which gives $\mathcal{O}(\frac{dn}{\varepsilon^2})$ for boxes.

[1] The runtime of the oracles can also depend on the required approximation guarantees. In order to simplify the notation, this dependency is not made explicit.

[2] We will always assume that ε is small, that is, $0 < \varepsilon < 1$.

2. Volume computation of unions

In this section we show that the volume computation of unions is **#P**-hard already for very simple axis-parallel boxes. After that we give an FPRAS for approximating the volume of the union of bodies which satisfy the three aforementioned oracles and describe several classes of objects for which the oracles can be answered efficiently.

2.1. Computational complexity of union calculations

Consider the following problem: Let $\mathcal{S}$ be a set of n axis-parallel boxes in $\mathbb{R}^d$ of the form $B = [a_1, b_1] \times \cdots \times [a_d, b_d]$ with $a_i, b_i \in \mathbb{R}, a_i < b_i$. The volume of one such box is $\textsc{vol}(B) = \prod_{i=1}^{d}(b_i - a_i)$. To compute the volume of the union of these boxes is known as Klee's Measure Problem (KMP).

It is know that the associated decision problem of deciding whether there is a point that is not in the union is **NP**-hard [9]. We consider the actual counting problem and prove in the following Theorem 1 that KMP is **#P**-hard. To the best of our knowledge there is no published result that explicitly states that KMP is **#P**-hard. However, without mentioning this implication, Suzuki and Ibaraki [27] sketch a reduction from #SAT to KMP. We present a reduction from #MON-CNF to KMP which we can reuse in Theorem 6 for the hardness proof for intersections. #MON-CNF counts the number of satisfying assignments of a Boolean formula in conjunctive normal form in which all variables are unnegated. While the problem of deciding satisfiability of such formula is trivial, counting the number of satisfying assignments is **#P**-hard [26].

Theorem 1. *KMP is* **#P***-hard.*

Proof. To reduce #MON-CNF to KMP, let $f = \bigwedge_{k=1}^{n} \bigvee_{i \in C_k} x_i$ be a monotone Boolean formula given in CNF with $C_k \subset [d] := \{1, \ldots, d\}$, for $k \in [n]$, d the number of variables, n the number of clauses. Since the number of satisfying assignments of f is equal to 2^d minus the number of satisfying assignments of its negation, we instead count the latter: Consider the negated formula $\bar{f} = \bigvee_{k=1}^{n} \bigwedge_{i \in C_k} \neg x_i$. First, we construct a box $A_k = [0, q_1^{(k)}] \times \cdots \times [0, q_d^{(k)}]$ in $\mathbb{R}^d$ for each clause C_k with one vertex at the origin and the opposite vertex at $(q_1^{(k)}, \ldots, q_d^{(k)})$, where we set

$$q_i^{(k)} = \begin{cases} 1, & \text{if } i \in C_k, \\ 2, & \text{if } i \notin C_k, \end{cases} \qquad i \in [d].$$

Observe that the union of the boxes A_k can be written as a union of boxes of the form $U_x = [x_1, x_1 + 1] \times \cdots \times [x_d, x_d + 1]$ with $x = (x_1, \ldots, x_d) \in \{0, 1\}^d$. Let $x \in \{0, 1\}^d$ and $U_x \subseteq \bigcup_{k=1}^{n} A_k$. Then there is a k such that $q_i^{(k)} = 2$ for all i with $x_i = 1$. By definition of $q_i^{(k)}$, this implies that $\bigwedge_{i \in C_k} \neg x_i$ and also $\bar{f}$ are satisfied.

The same holds in the opposite direction, that is, if x satisfies $\bar{f}$ then $U_x \subseteq \bigcup_{k=1}^{n} A_k$. Hence, since $\textsc{vol}(U_x) = 1$, we have $\textsc{vol}(\bigcup_{k=1}^{n} A_k) = |\{x \in \{0, 1\}^d \mid x \text{ satisfies } \bar{f}\}|$. Thus a polynomial time algorithm for KMP would result in a polynomial time algorithm for #MON-CNF, which proves the claim. □

Note that we actually proved a little bit more than stated in the theorem. That is, we proved that even calculating the volume of the union of boxes which all have the point 0^d in common is **#P**-hard. This specific problem is known as *hypervolume indicator* [30] and is a very popular and widely used measure of fitness of Pareto sets in evolutionary multi-objective optimization.

2.2. Approximation algorithm for the volume of unions

In this section we present an FPRAS for computing the volume of the union of objects for which we can answer PointQuery, VolumeQuery, and SampleQuery in polynomial time. The input of our algorithm ApproxUnion are the approximation ratio ε and the bodies $B_1, \ldots, B_n$ in $\mathbb{R}^d$ defined by the three oracles. It computes an approximation $\tilde{U} \in \mathbb{R}$ of $U := \textsc{vol}(\bigcup_{i=1}^{n} B_i)$ such that

$$\Pr\big[(1 - \varepsilon)U \leqslant \tilde{U} \leqslant (1 + \varepsilon)U\big] \geqslant 3/4 \tag{1}$$

in time polynomial in n, $1/\varepsilon$ and the query runtimes. Note that the constant 3/4 can be increased to any number by using a probability amplification technique.

We are following the algorithm of Karp et al. [18] which the authors used for approximating #DNF and other counting problems on discrete sets. The two main differences are that here we are handling continuous bodies in $\mathbb{R}^d$ and that we allow erroneous oracles. The latter relaxation is crucial to incorporate, amongst other things, the class of convex bodies. Our algorithm ApproxUnion is shown on page 604. The total number of steps of the algorithm is T. This number is chosen in advance such that one can prove that Eq. (1) holds. The algorithm itself is very simple. First, it queries the volumes[3] V_i' of

[3] Note that here and in the remainder an unprimed variable denotes an exact value and a primed variable denotes a value subject to some error introduced by the erroneous oracles.

Algorithm 1. ApproxUnion($\mathcal{S}, \varepsilon, \varepsilon_P, \varepsilon_V, \varepsilon_S$) calculates an ε-approximation of $U = \mathrm{VOL}(\bigcup_{i=1}^n B_i)$ for a set of bodies $S = \{B_1, \ldots, B_n\}$ in $\mathbb{R}^d$ determined by the oracles PointQuery, VolumeQuery and SampleQuery with error ratios $\varepsilon_P, \varepsilon_V, \varepsilon_S$.

```
ε̃ := (ε−ε_V)/(1+ε_V)
C̃ := ((1+ε_S)(1+ε_V)(1+nε_P))/((1−ε_V)(1−ε_P))
T := (24 ln(2)(1+ε̃)n)/(ε̃²−8(C̃−1)n)
for all i ∈ [n] do
    compute V'_i := VolumeQuery(B_i)
od
V' := Σ_{i=1}^n V'_i
for M := 0 to ∞ do
    choose i ∈ [n] with probability V'_i/V'
    x := SampleQuery(B_i)
    t_M := 0
    repeat
        if t_0 + ··· + t_M ⩾ T then return TV'/(nM)
        choose j ∈ [n] uniformly at random
        t_M := t_M + 1
    until PointQuery(x, B_j)
od
```

the bodies B_i and computes $V' = \sum_{i=1}^n V'_i$. Then it repeats the following: It chooses a body B_i with probability (roughly[4]) proportional to its volume and chooses a point $x \in B_i$ (roughly[4]) uniformly at random. Afterwards, the algorithm chooses bodies B_j with probability $1/n$ and (roughly[4]) checks whether $x \in B_j$. The number t_M of chosen bodies until we find a B_j with $x \in B_j$ can then be used to estimate how many bodies B_j cover x.

For this, observe that the point x is chosen with probability density $k(x)/V'$ with $k(x) = |i \in [n] \mid x \in B_i|$. Each number t_M has expected value (roughly[4]) $n/k(x)$ for a fixed $x \in \bigcup_i B_i$. Hence when PointQuery(x, B_j) answers yes, t_M should be of the order of (roughly[4]) $\int_x n/k(x) \cdot k(x)/V' \, dx = \int_x (n/V') \, dx = nU/V'$. This implies that when $\tilde{U}$ is returned and $T = t_0 + \cdots + t_M$, the value nUM/V' is near T, i.e., TV'/nM is near U. This gives the intuition why the algorithm returns the approximation $\tilde{U} = \frac{TV'}{nM}$.

In Section 2.4 we show correctness of ApproxUnion. More precisely, we show that it returns an ε-approximation with probability $\geqslant \frac{3}{4}$ and $T = \mathcal{O}(\frac{n}{\varepsilon^2})$ if $\varepsilon_S, \varepsilon_V \leqslant \frac{\varepsilon^2}{47n}$ and $\varepsilon_P \leqslant \frac{\varepsilon^2}{47n^2}$. The last inequality reflects the fact that we cannot be arbitrarily accurate if the given oracles are inaccurate. If all oracles can be calculated accurately, i.e., if $\varepsilon_P = \varepsilon_S = \varepsilon_V = 0$, the algorithm runs for just $T = \frac{8\ln(8)(1+\varepsilon)n}{\varepsilon^2}$ many steps.

Overall, the runtime of ApproxUnion is clearly $\mathcal{O}(n \cdot V(d) + M \cdot S(d) + T \cdot P(d)) = \mathcal{O}(n \cdot V(d) + T \cdot (S(d) + P(d)))$, where $V(d)$ is the worst VolumeQuery time for any of the bodies, analogously $S(d)$ for SampleQuery and $P(d)$ for PointQuery. If $\varepsilon_S, \varepsilon_V \leqslant \frac{\varepsilon^2}{47n}$ and $\varepsilon_P \leqslant \frac{\varepsilon^2}{47n^2}$, the runtime is $\mathcal{O}(nV(d) + \frac{n}{\varepsilon^2}(S(d) + P(d)))$.

For boxes all three oracles can be computed exactly in $\mathcal{O}(d)$. This implies that our algorithm ApproxUnion gives an ε-approximation of KMP with probability $\geqslant \frac{3}{4}$ in runtime $\mathcal{O}(\frac{nd}{\varepsilon^2})$. For more complex objects like convex bodies determined by a membership oracle, there are no exact oracles. The following section discusses different classes of objects for which our algorithm can be applied.

2.3. Classes of objects supported by our FPRAS

Finding an FPRAS for the union of a certain class of geometric objects now reduces to calculating the respective PointQuery, VolumeQuery and SampleQuery in polynomial time. We assume that we can get a random real number in constant time. Then all three oracles can be calculated in time $\mathcal{O}(d)$ for d-dimensional boxes. This already yields an FPRAS for the volume of the union of arbitrary boxes, e.g., for KMP. Note that if we have a body for which we can answer all those queries, all affine transformations of this body fulfill these three oracles, too. We will now present three further classes of geometric objects.

Generalized spheres and boxes. Let $\mathbf{B}_k$ be the class of boxes of dimension k, i.e., $\mathbf{B}_k = \{[a_1, b_1] \times \cdots \times [a_k, b_k] \mid a_i, b_i \in \mathbb{R}, a_i < b_i\}$ and $\mathbf{S}_\ell$ the class of spheres of dimension ℓ. We can combine any box $B \in \mathbf{B}_k$ and sphere $S \in \mathbf{S}_{d-k}$ to get a d-dimensional object $B \times S$. Furthermore, we can permute the dimensions afterwards to get a generalized "box-sphere." in $\mathbb{R}^3$ this corresponds to boxes, spheres and cylinders. VolumeQuery can be computed easily as we can compute the volume of a sphere by a well-known formula and thus the volume of the product $B \times S$. As one can check whether a given point $x = (x_1, \ldots, x_d)$ lies in $B \times S$ by checking whether $(x_1, \ldots, x_k)$ lies in B and $(x_{k+1}, \ldots, x_d)$ lies in S, also PointQuery is a standard task of geometry. To answer SampleQuery, it suffices to choose a random point $(x_1, \ldots, x_k)$ in B and to choose a random point inside the sphere S, which can be done in polynomial time as described, e.g., by Muller [24].

[4] "Roughly" only for erroneous oracles.

Convex bodies. As mentioned in the introduction, exact calculation of VolumeQuery for a polytope given as a list of vertices or facets is **#P**-hard [11,19]. Since there are randomized approximation algorithms (see Dyer et al. [12] for the first one) for the volume of a convex body determined by a membership oracle, we can answer VolumeQuery approximately. The same holds for SampleQuery as these algorithms make use of an almost uniform sampling method on convex bodies. See Lovász and Vempala [23] for a result showing that VolumeQuery can be answered with $\mathcal{O}^*(\frac{d^4}{\varepsilon_V^2})$ questions to the membership oracle and SampleQuery with $\mathcal{O}^*(\frac{d^3}{\varepsilon_S^2})$ queries, for arbitrary errors $\varepsilon_V, \varepsilon_S > 0$ (where the asterisk hides factors of $\log(d)$ and $\log(1/\varepsilon_V)$ or $\log(1/\varepsilon_S)$). PointQuery can naturally be answered with a single question to the membership oracle. By choosing $\varepsilon_V = \varepsilon_S = \frac{\varepsilon^2}{47n}$, Theorem 2 together with Lemma 3 shows that ApproxUnion is an FPRAS for the volume of the union of convex bodies which uses $\mathcal{O}^*(\frac{n^3d^3}{\varepsilon^4}(d+\frac{1}{\varepsilon^2}))$ membership queries.

Star-shaped bodies. Star-shaped bodies are a generalization of convex bodies which have at least one point such that every line through the point has a convex intersection with the body. They can also be viewed as the union of convex sets, with all the convex sets having a nonempty intersection. The subset of points that can "see" the full set is called the kernel of the star-shaped set. Assuming that we are given membership oracles for the body as well as for the kernel, Chandrasekaran, Dadush, and Vempala [10] recently showed that for star-shaped bodies SampleQuery can be answered with $\mathcal{O}^*(\frac{d^3}{\eta^3\varepsilon_S^2})$ questions to the membership oracle, where η is the fraction of the volume taken up by the kernel. We can also approximate the volume of the (convex) kernel with $\mathcal{O}^*(\frac{d^4}{\varepsilon_V^2})$ questions to the membership oracle as discussed above and estimate η by $\mathcal{O}(\frac{1}{\eta^2\varepsilon_V^2})$ samples. Then the runtime of VolumeQuery is $\mathcal{O}^*(\frac{d^4}{\varepsilon_V^2}+\frac{d^3}{\eta^5\varepsilon_V^2\varepsilon_S^2})$. PointQuery can again naturally be answered with a single question to the membership oracle.

Schlicht domains. Let $a_i, b_i\colon \mathbb{R}^{i-1} \to \mathbb{R}$ be continuous functions with $a_i \leqslant b_i$, where a_1, b_1 are constants. Let $K \subset \mathbb{R}^d$ be defined as the set of all points $(x_1,\ldots,x_d) \in \mathbb{R}^d$ such that $a_1 \leqslant x_1 \leqslant b_1, a_2(x_1) \leqslant x_2 \leqslant b_2(x_1),\ldots,a_d(x_1,\ldots,x_{d-1}) \leqslant x_d \leqslant b_d(x_1,\ldots,x_{d-1})$. K is called a schlicht domain in functional analysis. Fubini's theorem for schlicht domains states that we can integrate a function $f\colon K \to \mathbb{R}$ by iteratively integrating first over x_d, then over $x_{d-1},\ldots,$ until we reach x_1. This way, by integrating $f(\cdot)=1$, we can compute the volume of a schlicht domain as long as the integrals are computable in polynomial time, and thus answer VolumeQuery. Similarly, we can choose a random uniformly distributed point inside K: Let $K(y) = \{(x_1,\ldots,x_d) \in K \mid x_1 = y\}$. Then $K(y)$ is another schlicht domain for every $a_1 \leqslant y \leqslant b_1$. Assume that we can determine the volume of every such $K(y)$ and the integral $I(y) = \int_{a_1}^{y} K(x)\,dx$. Then the inverse function $I^{-1}\colon [0,V] \to \mathbb{R}$, where $V = \int_{a_1}^{b_1} K(x)\,dx$ is the volume of K, allows us to choose a y in $[a_1,b_1]$ with probability proportional to $\mathrm{vol}(K(y))$. By this we can iteratively choose a value y for x_1 and recurse to find a uniformly random point $(y_2,\ldots,y_d)$ in $K(y)$, plugging both together to get a uniformly distributed point $(y_1,\ldots,y_d)$ in K. Hence, as long as we can compute the involved integrals and inverse functions (or at least approximate them good enough), we can answer SampleQuery. Since PointQuery is trivially computable – as long as we can evaluate a_i and b_i efficiently – this gives an example showing that the classes of objects that fulfill our three conditions include not only convex bodies, but also certain schlicht domains.

Note that all above mentioned classes of geometric objects contain boxes and hence our hardness results still hold and an ε-approximation algorithm is the best one can hope for (unless $\mathbf{P} = \mathbf{NP}$).

2.4. Analysis of our algorithm

We now show correctness of our algorithm ApproxUnion described in Section 2.2 and prove bounds for its approximation ratio. The following theorem is our main result for ApproxUnion. It holds for exact and weak oracles.

Theorem 2. *Given errors $0 \leqslant \varepsilon_P, \varepsilon_S, \varepsilon_V < 1$ of the queries, the algorithm* ApproxUnion$(\{B_1,\ldots,B_n\},\varepsilon,\varepsilon_P,\varepsilon_V,\varepsilon_S)$ *returns a value $\tilde{U}$ with*

$$\Pr\big[(1-\varepsilon)U \leqslant \tilde{U} \leqslant (1+\varepsilon)U\big] \geqslant \frac{3}{4}$$

choosing

$$T = \frac{24\ln(2)(1+\tilde{\varepsilon})n}{\tilde{\varepsilon}^2 - 8(\tilde{C}-1)n},$$

under the condition

$$\varepsilon > \varepsilon_V + 2(1+\varepsilon_V)\sqrt{2(\tilde{C}-1)n} \tag{2}$$

where $U := \mathrm{vol}(\bigcup_{i=1}^n B_i)$, $\tilde{\varepsilon} := \frac{\varepsilon-\varepsilon_V}{1+\varepsilon_V}$, and $\tilde{C} := \frac{(1+\varepsilon_S)(1+\varepsilon_V)(1+n\varepsilon_P)}{(1-\varepsilon_V)(1-\varepsilon_P)}$.

First note that with accurate oracles, i.e., if $\varepsilon_P = \varepsilon_S = \varepsilon_V = 0$, we get $\tilde{\varepsilon} = \varepsilon$, $\tilde{C} = 1$ and, thus, $T = \frac{24\ln(2)(1+\varepsilon)n}{\varepsilon^2}$. As the condition (2) becomes trivial, above theorem implies that our algorithm is indeed an FPRAS.

Given non-zero query errors, one clearly cannot be arbitrary accurate, which is reflected by the lower bound (2) for ε. However, the following lemma shows that condition (2) is fulfilled for small enough ε_P, ε_S and ε_V.

Lemma 3. *For $\varepsilon_S, \varepsilon_V \leqslant \varepsilon^2/(47n)$ and $\varepsilon_P \leqslant \varepsilon^2/(47n^2)$ the condition (2) of Theorem 2 is fulfilled and we have $T = \mathcal{O}(\frac{n}{\varepsilon^2})$.*

Proof. As the right-hand side of (2) is increasing in ε_P, ε_V and ε_S, we can assume w.l.o.g. that $\varepsilon_V = \varepsilon_S = \frac{\varepsilon^2}{47n}$ and $\varepsilon_P = \frac{\varepsilon^2}{47n^2}$. This gives $\tilde{C} \leqslant (1+\frac{\varepsilon^2}{47n})^3(1-\frac{\varepsilon^2}{47n})^2$. Observe that $\frac{\varepsilon^2}{47n} \leqslant \frac{1}{47}$. Since $(1+x)^3(1-x)^{-2} \leqslant 1+\alpha x$ holds for $\alpha := \frac{8081}{1521}$ and also $0 \leqslant x \leqslant \frac{1}{40}$, we have $\tilde{C} \leqslant 1+\alpha\frac{\varepsilon^2}{47n}$. Hence, we can upper bound the right-hand side of (2) by

$$\varepsilon_V + 2(1+\varepsilon_V)\sqrt{2(\tilde{C}-1)n} \leqslant \frac{\varepsilon^2}{47n} + 2\left(1+\frac{\varepsilon^2}{47n}\right)\sqrt{2\alpha\frac{\varepsilon^2}{47}} \leqslant \varepsilon\left(\frac{1}{47} + 2\left(1+\frac{1}{47}\right)\sqrt{\frac{2\alpha}{47}}\right)$$

as $1/n \leqslant 1$ and $\varepsilon \leqslant 1$. Since $\frac{1}{47} + 2(1+\frac{1}{47})\sqrt{\frac{2\alpha}{47}} < 1$, condition (2) is fulfilled.

We now bound the terms $\tilde{\varepsilon}$ and $\tilde{C}$. Since $\varepsilon_V \geqslant 0$, we clearly have $\tilde{\varepsilon} \leqslant \varepsilon$ implying $1+\tilde{\varepsilon} \leqslant 2$. Furthermore, since $\varepsilon_V \leqslant \frac{\varepsilon^2}{47n}$ it also holds that

$$\tilde{\varepsilon} = \frac{\varepsilon - \varepsilon_V}{1+\varepsilon_V} \geqslant \frac{\varepsilon - \frac{\varepsilon^2}{47n}}{1+\frac{\varepsilon^2}{47n}} = \frac{\varepsilon(47n-\varepsilon)}{\varepsilon^2+47n} \geqslant \frac{\varepsilon 46n}{48n} = \frac{23}{24}\varepsilon$$

where we used $\varepsilon < 1$ and $n \geqslant 1$. Using this and the upper bound for $\tilde{C}$ we get for the denominator of T:

$$\tilde{\varepsilon}^2 - 8(\tilde{C}-1)n \geqslant \left(\frac{23}{24}\varepsilon\right)^2 - \frac{8}{47}\alpha\varepsilon^2 = \frac{64\,375}{4\,575\,168}\varepsilon^2.$$

Therefore, we get

$$T = \frac{24\ln(2)(1+\tilde{\varepsilon})n}{\tilde{\varepsilon}^2 - 8(\tilde{C}-1)n} \leqslant \frac{48\ln(2)n}{\frac{64\,375}{4\,575\,168}\varepsilon^2} < 2365\frac{n}{\varepsilon^2} = \mathcal{O}\left(\frac{n}{\varepsilon^2}\right). \qquad \square$$

In order to prove Theorem 2, we will generalize the corresponding proof of Karp et al. [18] to cover weak oracles. For most lemmas it suffices to insert the error constants $\varepsilon_P, \varepsilon_S, \varepsilon_V$, but for the main proof of Theorem 2 one has to be little bit more careful.

First, recall that we are given bodies $B_1, \ldots, B_n$ by oracles, where PointQuery(x, B_i) returns true for every $x \in B_i'$, such that the result is wrong for the set $W_i = (B_i \backslash B_i') \cup (B_i' \backslash B_i)$ with $\mathrm{vol}(W_i) < \varepsilon_P \mathrm{vol}(B_i)$, which implies

$$(1-\varepsilon_P)\mathrm{vol}(B_i) \leqslant \mathrm{vol}(B_i') \leqslant (1+\varepsilon_P)\mathrm{vol}(B_i). \tag{3}$$

Furthermore, the volume V_i' of body B_i is computed by VolumeQuery. V_i' is an ε_V-approximation of the corresponding exact volume V_i, i.e.,

$$(1-\varepsilon_V)V_i \leqslant V_i' \leqslant (1+\varepsilon_V)V_i. \tag{4}$$

We set $V' := \sum_{i=1}^n V_i'$ and $V := \sum_{i=1}^n V_i$. Then it clearly holds that

$$(1-\varepsilon_V)V \leqslant V' \leqslant (1+\varepsilon_V)V. \tag{5}$$

Furthermore, let U be the exact volume of the union of the B_i's and $\mu = U/V$.

As in Karp et al. [18], we define for a point $x \in \mathbb{R}^d$ the number of covering bodies $\mathrm{cov}(x) = |\{i \in [n] \mid \text{PointQuery}(x, B_i) = \text{true}\}|$. Additionally, we set

$$R_k := \left\{x \in \mathbb{R}^d \;\middle|\; \mathrm{cov}(x) = k\right\}$$

and $r_k := \mathrm{vol}(R_k)$. Then we have $\sum_{k=1}^n k r_k = \sum_{i=1}^n \mathrm{vol}(B_i')$, so that

$$(1-\varepsilon_P)V \leqslant \sum_{k=1}^n k r_k \leqslant (1+\varepsilon_P)V. \tag{6}$$

Furthermore, $\sum_{k=1}^n r_k = \mathrm{vol}(\bigcup_{i=1}^n B_i')$, so that

$$(1 - n\varepsilon_P)U \leqslant \sum_{k=1}^{n} r_k \leqslant (1 + n\varepsilon_P)U. \tag{7}$$

To get a sample point in our algorithm, we first choose an $i \in [n]$ with probability V'_i/V' and then choose a random point x in B'_i via SampleQuery(B_i). We consider the probability of this random point x to lie in the region R_k. With error-free oracles this probability would be $\Pr[x \in R_k] = \frac{kr_k}{V}$ as exactly k bodies cover each point of R_k and we have $\sum_{k=1}^{n} kr_k = V$ in the error-free setting. With errors, simple calculations using inequalities (3), (4) and (5) yield

$$\frac{(1-\varepsilon_S)(1-\varepsilon_V)}{(1+\varepsilon_V)(1+\varepsilon_P)}\frac{kr_k}{V} \leqslant \Pr[x \in R_k] \leqslant \frac{(1+\varepsilon_S)(1+\varepsilon_V)}{(1-\varepsilon_V)(1-\varepsilon_P)}\frac{kr_k}{V}. \tag{8}$$

In the algorithm, t_m denotes the number of iterations in the inner loop during the m-th iteration of the main loop, i.e., the number of trials to find a box containing the m-th point x. These t_m are independent identically distributed random variables. Let t be a variable distributed as each t_m. Then the following lemma holds.

Lemma 4. *Let* $0 \leqslant \lambda \leqslant \frac{1}{2}$, *and* $\tilde{C} = \frac{(1+\varepsilon_S)(1+\varepsilon_V)(1+n\varepsilon_P)}{(1-\varepsilon_V)(1-\varepsilon_P)}$. *Then*

$$\mathrm{Ex}\big[e^{\lambda t/n}\big] \leqslant \tilde{C}e^{(\lambda+2\lambda^2)\mu} \quad \text{and}$$
$$\mathrm{Ex}\big[e^{-\lambda t/n}\big] \leqslant \tilde{C}e^{-(\lambda-\lambda^2)\mu}.$$

These bounds closely match Lemmas 7 and 9 of Karp et al. [18]. Adapting the proof is straightforward. The factor $\tilde{C}$ arises by the usage of inequalities (7) and (8).

In the remainder, let $S_\ell := \sum_{i=0}^{\ell} t_i$ be the step at which the ℓ-th trial is completed and let N_i be the number of trials completed after step i, so that in the end $M = N_T$. Then $N_i < \ell$ if and only if $S_\ell > i$.

Corollary 5. *Let* $\varepsilon \leqslant 2$. *Then*

$$\Pr\big[S_\ell > (1+\varepsilon)n\mu\ell\big] \leqslant \tilde{C}^\ell e^{-\mu\varepsilon^2\ell/8} \quad \text{and}$$
$$\Pr\big[S_\ell < (1-\varepsilon)n\mu\ell\big] \leqslant \tilde{C}^\ell e^{-\mu\varepsilon^2\ell/8}.$$

This corresponds to Corollaries 8 and 10 in Karp et al. [18]. We can reuse their proof word for word; the only change is our Lemma 4 which brings in the factor $\tilde{C}$.

It remains to prove Theorem 2. The corresponding theorem of Karp et al. [18] is called Self-Adjusting Coverage Algorithm Theorem II. However, adapting their proof is not as straightforward as for the previous lemmas. It is presented in more detail in the remainder of this section.

Proof of Theorem 2. Let $\tilde{\varepsilon} = \frac{\varepsilon-\varepsilon_V}{1+\varepsilon_V}$ and $\tilde{C} = \frac{(1+\varepsilon_S)(1+\varepsilon_V)(1+n\varepsilon_P)}{(1-\varepsilon_V)(1-\varepsilon_P)}$ and assume

$$\varepsilon > \varepsilon_V + 2(1+\varepsilon_V)\sqrt{2(\tilde{C}-1)n}. \tag{9}$$

This implies $\tilde{\varepsilon} > 0$ and $\tilde{\varepsilon}^2 - 8(\tilde{C}-1)n > 0$. Let

$$k_1 := \frac{24\ln(2)(1+\tilde{\varepsilon})}{\mu(\tilde{\varepsilon}^2 - 8(\tilde{C}-1)n)(1+\tilde{\varepsilon})}$$

and

$$k_2 := \frac{24\ln(2)(1+\tilde{\varepsilon})}{\mu(\tilde{\varepsilon}^2 - 8(\tilde{C}-1)n)(1-\tilde{\varepsilon})}.$$

Then $T = k_1 n\mu(1+\tilde{\varepsilon}) = k_2 n\mu(1-\tilde{\varepsilon})$. Now, if we have $k_1 \leqslant M \leqslant k_2$, then

$$\frac{T}{k_2} \leqslant \frac{T}{M} \leqslant \frac{T}{k_1}$$

and thus

$$\frac{TV'}{nk_2} \leqslant \frac{TV'}{nM} = \tilde{U} \leqslant \frac{TV'}{nk_1}.$$

By plugging in T, k_1 and k_2 we get

$$(1-\tilde{\varepsilon})\mu V' \leqslant \tilde{U} \leqslant (1+\tilde{\varepsilon})\mu V'.$$

A little calculus shows that $(1-\tilde{\varepsilon})(1-\varepsilon_V) \geqslant 1-\varepsilon$, only based on the definition of $\tilde{\varepsilon}$, the non-negativity of ε_V and ε, and $\varepsilon_V \leqslant \varepsilon$. By using this, Eq. (5), the fact $(1+\tilde{\varepsilon})(1+\varepsilon_V) = 1+\varepsilon$, and $\mu = U/V$ we get

$$(1-\varepsilon)U \leqslant \tilde{U} \leqslant (1+\varepsilon)U,$$

and thus the estimation is an ε-approximation, if $k_1 \leqslant M \leqslant k_2$. Hence, it suffices to show

$$\Pr[M > k_2] + \Pr[M < k_1] \leqslant \frac{1}{4}. \tag{10}$$

We have

$$\Pr[M < k_1] = \Pr[S_{k_1} > T] = \Pr\bigl[S_{k_1} > k_1 n\mu(1+\tilde{\varepsilon})\bigr].$$

By $\tilde{\varepsilon} > 0$ we have, using Corollary 5,

$$\Pr[M < k_1] \leqslant \tilde{C}^{k_1} e^{-\mu\tilde{\varepsilon}^2 k_1/8} \leqslant e^{(\tilde{C}-1)k_1} e^{-\mu\tilde{\varepsilon}^2 k_1/8} = e^{(\tilde{C}-1-\mu\tilde{\varepsilon}^2/8)k_1} = e^{-3\ln(2)\frac{\mu\tilde{\varepsilon}^2-8(\tilde{C}-1)}{\mu\tilde{\varepsilon}^2-8(\tilde{C}-1)n\mu}}.$$

By inequality (9) we get $\mu\tilde{\varepsilon}^2 - 8(\tilde{C}-1)n\mu > 0$ and since $\mu \geqslant 1/n$ we have $\mu\tilde{\varepsilon}^2 - 8(\tilde{C}-1) \geqslant \mu\tilde{\varepsilon}^2 - 8(\tilde{C}-1)n\mu$. Hence,

$$\Pr[M < k_1] \leqslant e^{-3\ln(2)} = \frac{1}{8}.$$

We analogously get $\Pr[M > k_2] \leqslant \frac{1}{8}$. Plugging both results in Eq. (10) finishes the proof. □

3. Volume computation of intersections

In this section we are considering the complement to the union problem. We show that surprisingly the volume of a intersection of a set of bodies is often much harder to calculate than its union. For many classes of geometric objects there is even no randomized approximation possible.

As the problem of computing the volume of a polytope is **#P**-hard [11,19], so is the computation of the volume of the intersection of general (i.e., not necessarily axis-parallel) boxes in $\mathbb{R}^d$. This can be seen by describing a polytope as an intersection of half-planes and representing these as general boxes.

Let us now consider the convex bodies again. Trivially, the intersection of convex bodies is convex itself. From the oracles defining the given bodies $B_1, \ldots, B_n$ one can simply construct an oracle which answers PointQuery for the intersection of those objects: Given a point $x \in \mathbb{R}^d$ it asks all n oracles and returns true if and only if x lies in all the bodies. One could now believe that we can apply the result of Dyer et al. [12] and the subsequent improvements mentioned in the introduction to approximate the volume of the intersection and get an FPRAS for the problem at hand. The problem with this approach is that the intersection is not "well-guaranteed." That is, there is no point known that lies in the intersection, not to speak of a sphere inside it. However, the algorithm of Dyer et al. [12] relies vitally on the assumption that the given body is well-guaranteed and hence cannot be applied for approximating the volume of the intersection of convex bodies.

We will now present a hard subproblem which shows that the volume of the intersection cannot be approximated (deterministic or randomized) in general.

Definition 1. For $p \in [0,1]^d$, let $B_p := \{x \mid 0 \leqslant x_i \leqslant p_i\ \forall 1 \leqslant i \leqslant d\}$. We call $\overline{B}_p := [0,1]^d \setminus B_p$ a *co-box*.

A co-box is a box where we cut out another box at one corner. The resulting object can itself be a box, too, but in general it is not even convex. It can be seen as the complement of a box B_p relative to a larger background box $[0,1]^d$. Note that it is easy to calculate the union of a set of co-boxes $\{\overline{B}_{p_1}, \overline{B}_{p_2}, \ldots, \overline{B}_{p_n}\}$ with $p_1, p_2, \ldots, p_n \in [0,1]^d$ as $\bigcup \overline{B}_{p_i} = [0,1]^d \setminus \bigcap B_{p_i}$. On the other hand, the calculation of the intersection of a set of co-boxes is **#P**-hard by Theorem 1 as $\bigcup B_{p_i} = [0,1]^d \setminus \bigcap \overline{B}_{p_i}$. The following theorem shows that it is not even approximable.

Theorem 6. *Let $p_1, p_2, \ldots, p_n \in \mathbb{R}^d_{\geqslant 0}$. Then the volume of $\bigcap_{i=1}^n \overline{B}_{p_i}$ cannot be approximated in (deterministic or randomized) polynomial time by a factor of $2^{d^{1-\varepsilon}}$ for any $\varepsilon > 0$ unless* **NP** = **BPP**.

Proof. Consider again the problem #MON-CNF already defined in Section 2. Let $f = \bigwedge_{k=1}^n \bigvee_{i \in C_k} x_i$ be a monotone Boolean formula given in CNF as defined in the proof of Theorem 1. We now construct for every clause C_k a co-box $\overline{B}_{p_k}$ with $p_k = (p_1^{(k)}, \ldots, p_d^{(k)})$ and $p_i^{(k)} = 1/2$ if $i \in C_k$, and $p_i^{(k)} = 1$ otherwise. The boxes B_{p_k} here correspond to the A_k in the proof of Theorem 1. For $x = (x_1, \ldots, x_d) \in \{0,1\}^d$ let $U_x = [\frac{1}{2}x_1, \frac{1}{2}(x_1+1)] \times \cdots \times [\frac{1}{2}x_d, \frac{1}{2}(x_d+1)]$. As in the proof of Theorem 1 one shows that $U_x \subseteq \bigcup_{k=1}^n B_{p_k}$ if and only if x satisfies $\bar{f}$ and hence also $U_x \subseteq \bigcap_{k=1}^n \overline{B}_{p_k}$ if and only if x satisfies f. This

Table 1
Results for the computational complexity of the calculation of the volume of union and intersection (asymptotic in the dimension d).

Geometric objects	Volume of the union	Volume of the intersection
Axis-parallel boxes	**#P**-hard + FPRAS	easy
General boxes	**#P**-hard + FPRAS	**#P**-hard
Co-boxes	easy	**#P**-hard + APX-hard[b]
Schlicht domains	**#P**-hard + FPRAS[a]	**#P**-hard + APX-hard[b]
Convex bodies	**#P**-hard + FPRAS	**#P**-hard

[a] If the integrals are computable in polynomial time (cf. Section 2.3).

[b] Theorem 6 even proves that for every fixed $\varepsilon > 0$ approximating the volume within $2^{d^{1-\varepsilon}}$ is **NP**-hard.

implies that the volume of $\bigcap_{i=1}^{n} \overline{B}_{p_i}$ times 2^d equals the number of satisfying assignments of f. Roth [26] showed that #MON-CNF cannot be approximated by a factor of $2^{d^{1-\varepsilon}}$ unless $\mathbf{NP} = \mathbf{BPP}$. By above reduction, the same inapproximability must hold for the volume of $\bigcap_{i=1}^{n} \overline{B}_{p_i}$. $\square$

This shows that in general there does not exist a polynomial time multiplicative ε-approximation of the volume of the intersection of bodies in $\mathbb{R}^d$. This holds for all classes of objects which include co-boxes, e.g. schlicht domains (cf. Section 2.3). Though there is no multiplicative approximation, we can still give an additive approximation algorithm, that is, we can efficiently find a number $\tilde{V}$ such that

$$\Pr[V - \varepsilon V_{\min} \leqslant \tilde{V} \leqslant V + \varepsilon V_{\min}] \geqslant \frac{3}{4}$$

where V is the exact volume of the intersection and $V_{\min}$ is the minimal volume of any of the given bodies $B_1, \ldots, B_n$. If we could replace $V_{\min}$ by V in the equation above, we would have an FPRAS. This is not possible in general as the ratio of V and $V_{\min}$ can be arbitrarily small. Hence, such an ε-approximation is not relative to the exact result, but to the volume of some greater body. This is an additive approximation since after rescaling, so that $V_{\min} \leqslant 1$ we get an additive error of ε. Clearly, we get the result from above by uniform sampling in the body $B_{\min}$ corresponding to the volume $V_{\min}$. Consider $\tilde{V} = V_{\min}(Z_1 + \cdots + Z_N)/N$, where Z_i is a random variable valued 1, if the i-th sample point $x_i = \textsc{SampleQuery}(B_{\min})$ lies in the intersection of $B_1, \ldots, B_n$, and 0 otherwise. Using Chebyshev's inequality one can show quite easily that $\tilde{V}$ gives an approximation as desired, if we choose N proportional to $1/\varepsilon^2$ with the right factor. This gives an approximation algorithm with runtime $\mathcal{O}(nV(d) + \frac{1}{\varepsilon^2} S(d) + \frac{n}{\varepsilon^2} P(d))$, yielding $\mathcal{O}(\frac{dn}{\varepsilon^2})$ for (not necessarily axis-parallel) boxes.

4. Discussion and open problems

We have proven **#P**-hardness for the exact computation of the volume of the union of bodies in $\mathbb{R}^d$ as long as the class of bodies includes axis-parallel boxes. The same holds for the intersection if the class of bodies contains general boxes. We have also presented an FPRAS for approximating the volume of the union of bodies that allow three very natural oracles. Very recently, there appeared a few deterministic polynomial-time approximations (FPTAS) for hard counting problems (e.g. [3,5,14,15,29]). It seems to be a very interesting open question whether there exists a deterministic approximation for the union of some non-trivial class of bodies. Since the volume of convex bodies determined by oracles cannot be approximated to within a factor that is exponential in d [4], the existence of such a deterministic approximation for the union seems implausible. It is also open whether there is a constant C so that KMP can be efficiently deterministically approximated within a factor of C, i.e., if they are in APX?

For the intersection we proved that no multiplicative approximation (deterministic or randomized) is possible for co-boxes (cf. Definition 1), but we also presented a very simple additive approximation algorithm for the intersection problem. It would be interesting to know if there is a hard class for multiplicative approximation which contains only convex bodies.

Our results are summarized in Table 1. Note the correspondence between axis-parallel boxes and co-boxes. The discrete counterpart to their approximability and inapproximability is the approximability of #DNF and the inapproximability of #SAT.

Acknowledgements

We thank Joshua Cooper for suggesting the proof of Theorem 1, Ernst Albrecht for discussions on schlicht domains, and Eckart Zitzler for comments on an earlier version of this paper. This work was partially supported by a postdoctoral fellowship from the German Academic Exchange Service (DAAD).

References

[1] P.K. Agarwal, H. Kaplan, M. Sharir, Computing the volume of the union of cubes, in: Proc. 23rd Annual Symposium on Computational Geometry (SoCG '07), 2007, pp. 294–301.

[2] D. Applegate, R. Kannan, Sampling and integration of near log-concave functions, in: Proc. 23rd Annual ACM Symposium on Theory of Computing (STOC '91), 1991, pp. 156–163.
[3] A. Bandyopadhyay, D. Gamarnik, Counting without sampling: new algorithms for enumeration problems using statistical physics, in: Proc. 17th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '06), 2006, pp. 890–899.
[4] I. Bárány, Z. Füredi, Computing the volume is difficult, Discrete Comput. Geom. 2 (1987) 319–326.
[5] M. Bayati, D. Gamarnik, D.A. Katz, C. Nair, P. Tetali, Simple deterministic approximation algorithms for counting matchings, in: Proc. 39th Annual ACM Symposium on Theory of Computing (STOC '07), 2007, pp. 122–127.
[6] J.L. Bentley, Algorithms for Klee's rectangle problems, Department of Computer Science, Carnegie Mellon University, 1977, unpublished notes.
[7] K. Bringmann, T. Friedrich, Approximating the volume of unions and intersections of high-dimensional geometric objects, in: Proc. 19th International Symposium on Algorithms and Computation (ISAAC '08), in: LNCS, vol. 5369, 2008, pp. 436–447.
[8] T.M. Chan, Semi-online maintenance of geometric optima and measures, SIAM J. Comput. 32 (2003) 700–716.
[9] T.M. Chan, A (slightly) faster algorithm for Klee's measure problem, Comput. Geom. 43 (2010) 243–250.
[10] K. Chandrasekaran, D. Dadush, S. Vempala, Thin partitions: Isoperimetric inequalities and a sampling algorithm for star shaped bodies, in: Proc. 21st Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '10), 2010, pp. 1630–1645.
[11] M.E. Dyer, A.M. Frieze, On the complexity of computing the volume of a polyhedron, SIAM J. Comput. 17 (1988) 967–974.
[12] M.E. Dyer, A.M. Frieze, R. Kannan, A random polynomial time algorithm for approximating the volume of convex bodies, J. ACM 38 (1991) 1–17.
[13] M.L. Fredman, B.W. Weide, On the complexity of computing the measure of $\bigcup[a_i, b_i]$, Commun. ACM 21 (1978) 540–544.
[14] D. Gamarnik, D. Katz, Correlation decay and deterministic FPTAS for counting list-colorings of a graph, in: Proc. 18th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '07), 2007, pp. 1245–1254.
[15] N. Halman, D. Klabjan, C.-L. Li, J.B. Orlin, D. Simchi-Levi, Fully polynomial time approximation schemes for stochastic dynamic programs, in: Proc. 19th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '08), 2008, pp. 700–709.
[16] R. Kannan, L. Lovász, M. Simonovits, Random walks and an $O^*(n^5)$ volume algorithm for convex bodies, Random Struct. Algorithms 11 (1997) 1–50.
[17] H. Kaplan, N. Rubin, M. Sharir, E. Verbin, Counting colors in boxes, in: Proc. 18th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '07), 2007, pp. 785–794.
[18] R.M. Karp, M. Luby, N. Madras, Monte-Carlo approximation algorithms for enumeration problems, J. Algorithms 10 (1989) 429–448.
[19] L.G. Khachiyan, The problem of calculating the volume of a polyhedron is enumerably hard, Russian Math. Surv. 44 (1989) 199–200.
[20] V. Klee, Can the measure of $\bigcup[a_i b_i]$ be computed in less than $O(n \log n)$ steps? Amer. Math. Monthly 84 (1977) 284–285.
[21] L. Lovász, M. Simonovits, The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume, in: Proc. 31st Annual Symposium on Foundations of Computer Science (FOCS '90), 1990, pp. 346–354.
[22] L. Lovász, M. Simonovits, Random walks in a convex body and an improved volume algorithm, Random Struct. Algorithms 4 (1993) 359–412.
[23] L. Lovász, S. Vempala, Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm, J. Comput. Syst. Sci. 72 (2006) 392–417.
[24] M.E. Muller, A note on a method for generating points uniformly on n-dimensional spheres, Commun. ACM 2 (1959) 19–20.
[25] M.H. Overmars, C.-K. Yap, New upper bounds in Klee's measure problem, SIAM J. Comput. 20 (1991) 1034–1045.
[26] D. Roth, On the hardness of approximate reasoning, Artif. Intell. 82 (1996) 273–302.
[27] S. Suzuki, T. Ibaraki, An average running time analysis of a backtracking algorithm to calculate the measure of the union of hyperrectangles in d dimensions, in: Proc. 16th Canadian Conference on Computational Geometry (CCCG '04), 2004, pp. 196–199.
[28] J. van Leeuwen, D. Wood, The measure problem for rectangular ranges in d-space, J. Algorithms 2 (1981) 282–300.
[29] D. Weitz, Counting independent sets up to the tree threshold, in: Proc. 38th Annual ACM Symposium on Theory of Computing (STOC '06), 2006, pp. 140–149.
[30] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, IEEE Trans. Evolut. Comput. 3 (1999) 257–271.