Weighted Geometric Set Cover via Quasi-Uniform Sampling

Kasturi Varadarajan*
Department of Computer Science
University of Iowa
Iowa City, IA 52242
kvaradar@cs.uiowa.edu

ABSTRACT

There has been much progress on geometric set cover problems, but most known techniques only apply to the unweighted setting. For the weighted setting, very few results are known with approximation guarantees better than that for the combinatorial set cover problem. In this article, we employ the idea of *quasi-uniform sampling* to obtain improved approximation guarantees in the weighted setting for a large class of problems for which such guarantees were known in the unweighted case. As a consequence of this sampling method, we obtain new results on the fractional set cover packing problem.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*

General Terms

Algorithms, Theory

Keywords

Set cover, epsilon nets, approximation

1. INTRODUCTION

The combinatorial set cover problem has a privileged place in computer science for the wealth of theoretical results known about it and its applications. Here one is given a ground set X together with a family $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$ where each S_i is a subset of X and has associated with it a non-negative weight $w(S_i)$. The family $\mathcal{F}$ covers X, that is, X is contained in the union of the elements of $\mathcal{F}$. The goal is to find a minimum weight subfamily $\mathcal{G} \subseteq \mathcal{F}$ that also covers X. Here, the weight of a subfamily $\mathcal{G}$ is simply the sum of the weights of the elements in it. This NP-hard problem admits several different polynomial time algorithms that guarantee an approximation factor of $O(\log |X|)$ – some of these are combinatorial greedy algorithms and some are based on (randomized) rounding of the natural linear programming relaxation [33]. These results hold for both the weighted and the unweighted case, which is obtained when $w(S_i) = 1$ for each i. This approximation factor has also been shown to be the best possible under standard complexity theoretic assumptions [30].

We are interested in the geometric set cover problem where X is a (usually finite) subset of some fixed dimensional Euclidean space – in this paper we will assume this to be the two dimensional plane unless stated otherwise. The family $\mathcal{F}$ of subsets of X is induced by some family of objects, for example, disks, triangles, or visibility polygons. In the set cover problem with disks, for instance, we are given a set $\{D_1, D_2, \ldots, D_n\}$ of disks along with their weights, and we wish to find the minimum weight subset of disks that cover (whose union contains) X. This is clearly an instance of the combinatorial set cover problem by computing $S_i = X \cap D_i$. Given the constraints imposed by geometry, it is reasonable to expect that we can obtain approximation factors better than $O(\log |X|)$ for the combinatorial version. Indeed, as we review below, we now have a few techniques that yield better approximation factors for certain types of objects. However, these techniques have hitherto worked only for the unweighted case. Our main aim in this paper is to extend some of these techniques to the weighted case.

We now review the techniques that have yielded improved approximation factors for the unweighted case. Combinatorial algorithms cleverly designed for the problem at hand have been successful for some problems, see for example [22, 6]. Very recently, the general technique of local search has been shown, quite surprisingly, to yield polynomial time approximation schemes (PTAS's) for some problems [9, 27]. For instance, this approach yields a PTAS for set cover with disks [27]. However, it has recently been shown that such a PTAS does not exist (unless P = NP) for fairly "nice" objects, for example, fat triangles of roughly the same size [19]. An interesting question that is open is whether local search or other combinatorial approaches can be shown to give weaker but sub-logarithmic approximation guarantees for other classes of objects.

Such guarantees can often be obtained by rounding the natural linear programming relaxation of the set cover problem. The approximation factor obtained by such rounding is related to the well known combinatorial problem of bound-

*Supported by NSF grant CCF 0915543.

ing the size of an ε-*net*. Suppose that we have a collection of n disks and an integer $1 \le L \le n$. An $\frac{L}{n}$-net is a subset of the disks that covers all points in the plane that are L-deep, that is, contained within at least L of the disks. The question is, what is a good upper bound on the size of an $\frac{L}{n}$-net over all families of n disks and all $1 \le L \le n$? Clearly this bound must be at least $\frac{n}{L}$ and we want to know how much larger it must be. We know [7, 16] that if for any type of object (disks, triangles, axes-parallel rectangles) there are $\frac{L}{n}$-nets of size at most $\frac{n}{L}g(\frac{n}{L})$ (and we can efficiently compute such nets), then the integrality gap and the approximation factor obtained via the natural linear programming relaxation of the problem is $O(g(\frac{n}{L}))$. Nets of size $O(\frac{n}{L}\log\frac{n}{L})$ are known to exist for fairly general geometric objects, those of "constant description complexity" – this includes disks, triangles, rectangles, and by far all the types we are concerned with in this paper [11, 20]. In fact, a random sample of this size is a net with constant probability. Thus for these objects we obtain an approximation guarantee of $O(\log\lambda^*)$ where λ^* is the size of the optimal solution.

There is hope that nets of size $O(\frac{n}{L})$ should exist for very general subclasses of these objects, but this is an outstanding open problem. One situation where nets of size better than $O(\frac{n}{L}\log\frac{n}{L})$ are known to exist is when the *combinatorial complexity of the boundary of the union* (for brevity, union complexity) is near-linear [13, 4, 32]. Given m disks, the boundary of their union is a set of one-dimensional components each of which can be obtained by gluing together pieces of the boundaries of the disks; the union complexity is roughly the number of such pieces. We refer the reader to [31] for more precise definitions. If for a certain type of object, the union complexity of m of the objects is at most $mh(m)$, then there are $\frac{L}{n}$-nets of size $O(\frac{n}{L}\log h(\frac{n}{L}))$, and we thus obtain an $O(\log h(\lambda^*))$ approximation for the corresponding set cover problem [4]. Objects for which improved approximation algorithms are obtained using this approach include disks and fat triangles.

1.1 Our Contribution

Compared to this wealth of results for the unweighted setting, very few instances are known in the weighted setting where we have an approximation guarantee that is better than $O(\log|X|)$, where X, we remind the reader, is the ground set in the input for set cover. Two such instances are *unit* disks in the plane, for which constant factor approximations are known based on dynamic programming [3] and the linear programming approach [29], and the problem of guarding a one-dimensional terrain for which a constant factor approximation is given based on the linear programming approach [15]. A third and somewhat easier instance is when the objects doing the covering are translates of a fixed polygon, and here a constant factor approximation is implied by results in [18]. In these three instances, very special structure of the problem at hand is exploited. Thus, even for arbitrary disks and fat triangles, we do not currently have an approximation guarantee that is better than $O(\log|X|)$. It is easy to set up examples in the weighted setting where the basic local search approach doesn't yield improved bounds even for object types on which it yields a PTAS in the unweighted setting. Turning to the linear programming approach, we are now interested in an upper bound on the *weight* of an $\frac{L}{n}$-net and not its size. Clearly, this bound on the weight is at least as large as $\frac{W}{L}$, where W is the total weight of all the objects, and can be shown to be $O(\frac{W}{L}\log n)$ for fairly general geometric objects [16]. A bound of $O(\frac{W}{L}g(n))$ on the maximum weight of a net implies an approximation guarantee of $O(g(n))$ for the corresponding weighted set cover problem. Unfortunately, a bound better than $O(\frac{W}{L}\log n)$ on the maximum weight of a net is not known even for objects with small union complexity such as fat triangles or disks.

Why do the approaches in [13, 4, 32], which yield improved nets for objects such as fat triangles and disks, fail to do so in the weighted case? The answer is that they resort to random sampling methods that are in effect non-uniform. That is, while they guarantee that the number of objects in the net is small in the aggregate, they do not bound the probability of any particular object being in the net. This shortcoming is explicitly identified in [32].

The main contribution of this article is a method for computing nets based on *quasi-uniform sampling*, where the probability of any object being in the $\frac{L}{n}$-net is close to $\frac{1}{L}$, the natural bound. The method yields low weight nets for objects with near-linear union complexity, but for concreteness we state it for fat triangles. A triangle is ρ-fat if the ratio of the radius of its smallest circumscribing circle to the radius of its largest inscribed circle is bounded by ρ. By a fat triangle, we mean one that is ρ-fat for some arbitrary constant ρ. The union complexity of m fat triangles is bounded by $O(m\log\log m)$ [25]. Our main result is the following theorem on computing nets by quasi-uniform sampling:[1]

THEOREM 1. *There is a probabilistic polynomial time algorithm that takes as input a set T of n fat triangles and a parameter $1 \le L \le n$, and outputs a subset $T' \subset T$ that covers all points that are L-deep with respect to T, and has the following property: for any triangle $t \in T$, the probability that t is in the output T' is at most*

$$\frac{O(\log\log\log n)}{L}.$$

Clearly, this algorithm outputs an $\frac{L}{n}$-net whose expected weight is $O(\log\log\log n)\frac{W}{L}$ and we obtain an $O(\log\log\log n)$ approximation to the weighted set cover problem with fat triangles. At a high level, the theorem proceeds by showing a variant of it where we allow the sampling probability to be $\frac{O(\log L)}{L}$, but we ensure that L-deep points are $\log L$-deep with respect to the sample. This variant exploits the sparsity of the *shallow level* and in this sense it is similar to [32] and an algorithm of [9] for the independent set problem. But there are crucial differences needed to obtain the quasi-uniformity. The variant is then applied again to the sample, but with L now set to be $\log L$. (This is similar to [32].) Three such iterations suffice to obtain the Theorem.

For disks in the plane, our version of the theorem has a sampling probability bounded by $\frac{2^{O(\log^* n)}}{L}$, and we obtain a $2^{O(\log^* n)}$ approximation for the corresponding set cover problem. For objects with union complexity $mh(m)$, we obtain an approximation factor of $2^{O(\log^* n)}\log h(n)$, but this bound can sometimes be tightened further.

[1]The reader may find the flavor of the statement similar to that in completely different contexts, such as probabilistic approximation of metrics by tree metrics [5, 17].

1.2 Consequences for Set Cover Packing

In the *set cover packing problem*, we are given a ground set X, a family $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$ where each S_i is a subset of X, and the goal is to partition $\mathcal{F}$ into as many blocks as possible so that each block is a cover for X. In the *fractional set cover packing problem*, we want to compute a set of blocks plus a fraction $0 \leq f(B) \leq 1$ corresponding to each block B, so that (a) each block covers X, and (b) for each i the sum of the fractions of the blocks to which S_i belongs is at most 1, and the objective is to maximize the sum of the fractions. One motivation for both problems comes from the world of sensors [8] – here X is a region that needs to be monitored, S_i is the sub-region monitored by the i-th sensor, each sensor has a battery life of one time unit, and the goal is find a schedule for the sensors so that X remains monitored for as long as possible. The set cover packing problem corresponds to the *non-preemptive* scenario where once a sensor is turned on, it runs for one continuous unit of time. The fractional version corresponds to the *preemptive* scenario where we may turn a sensor on and off as often as we like.

In the combinatorial setting the best approximations for these problems is logarithmic (see [1, 8, 14] and the references therein). Given the motivation from sensors, geometric versions of the problem have been extensively studied recently. Let L denote the *load* of the instance, that is, the largest integer value so that each point in X is contained in at least L of the sets in $\mathcal{F}$. Clearly, the load is an upper bound on the objective function value in both versions of the problem. Previous work has shown that we can obtain a solution whose objective function value is (a) $\Omega(\frac{1}{\log(n/L)})$ of the load for the integral problem when the sets S_i are induced by fairly general geometric objects [8]; (b) $\Omega(1)$ of the load for the integral problem when the sets S_i are induced by *translates* of a fixed convex polygon [2, 18]; (c) $\Omega(1)$ of the load for the fractional problem when the sets S_i are induced by unit disks [29].

Our results on quasi-uniform sampling extend the frontier for such results:

Corollary 2. *Given an instance of the fractional set cover packing problem with X being a set of points in the plane, and $\mathcal{F}$ consisting of subsets induced by a set T of n fat triangles, there is a solution whose value is at least $\Omega(\frac{1}{\log\log\log n})$ of the load L.*

Proof. Consider the algorithm of Theorem 1 for computing an $\frac{L}{n}$-net for T, and the distribution over covers of all L-deep points that it induces. For any cover B that is output by this distribution, set $f(B) = \Pr(B) * \frac{L}{c\log\log\log n}$, where $c > 0$ is a sufficiently large constant, and $\Pr(B)$ is the probability that B is output. $\square$

The algorithm that outputs this fractional set cover packing is of course not polynomial, because it needs to compute the distribution corresponding to the output of the algorithm of Theorem 1. However, as we describe in some detail in this paper, we can obtain a randomized polynomial algorithm that computes a packing that is nearly as good by sampling from this distribution.

Organization of the Paper. In Section 2 we review some rather standard notions that we need. In Section 3, we establish Theorem 1. In Sections 4 and 5, we derive its consequences for the weighted set cover problem and the fractional set cover packing, respectively. We conclude in 6 with some remarks on generalizing these results to object families with small union complexity and with some open problems.

2. PRELIMINARIES

In this section, we review the notion of an arrangement of triangles in the plane, and state the well known result bounding the number of shallow cells when the triangles are fat.

A triangle t can be partitioned into seven *features* – the three vertices, the three edges (we don't include the endpoints of an edge when we talk about it as a feature), and its interior. Now given a set T of triangles in the plane, we can define an equivalence relation on the plane – two points are related if they are contained in exactly the same set of features. The connected components of the resulting equivalence classes are the *cells* in the *arrangement* of T. A cell can be a point (thus zero-dimensional), a line segment (thus one-dimensional), or an open polygonal region (thus two-dimensional).

We define the *level* or *depth* of a point in a given set T of triangles to be the number of triangles it is contained in. Note that two points that belong to the same cell have the same level, because they are contained in the same set of triangles. Thus we can speak of the level or depth of a cell. The level of a point is an integer between 0 and $|T|$. We will say that a point is *k-deep* with respect to a set T of triangles if its level is at least k.

We will need the following well known result, which follows from the fact that the union complexity of m fat triangles is $O(m \log\log m)$ [25] and a certain application of the probabilistic method [12]. The lemma bounds the number of cells that are at level at most α, and it is through this that the union complexity comes into play in our results.

Lemma 3. *Let F be a set of m ρ-fat triangles in general position, and $1 \leq \alpha \leq m$ be a parameter. The number of cells with level at most α in the arrangement of F is $O(\alpha m \log\log \frac{m}{\alpha})$.*

3. QUASI-UNIFORM SAMPLING

In this section, we establish the main result of this paper, which is Theorem 1. Throughout this section, let $\phi(x)$ denote the function $\log\log x$. We first need the following:

Theorem 4. *Let F be a set of fat triangles in the plane, and let $1 \leq \alpha \leq |F|$ be a parameter. There is a probabilistic polynomial-time algorithm that outputs a subset F' of F that has at least $\log\alpha$ triangles covering each α-deep point in F, and has the following property: For any triangle $t \in F$, the probability that $t \in F'$ is at most $\frac{O(\max\{\log\alpha, \log\phi(|F|/\alpha)\})}{\alpha}$.*

Proof. For notational convenience, let $m = |F|$. To prove the theorem we first describe a probabilistic process that outputs a cover that has at least $\log\alpha$ triangles covering each cell of the arrangement of F whose depth is between α and 2α. With this in mind, let $\beta = \max\{\log\alpha, \log\phi(m/\alpha)\}$. We need a certain ordering of the triangles in F.

Let $N = F$ initially. Let $\mathcal{C}$ denote the set of cells in the arrangement of N with level at most 2α. So the size of $\mathcal{C}$ is $O(|N|\alpha\phi(|N|/\alpha))$, due to Lemma 3. The number of pairs in $(t, \tau) \in N \times \mathcal{C}$ such that triangle t contains cell τ is bounded by $|\mathcal{C}| \cdot 2\alpha \leq d|N|\alpha^2\phi(\frac{|N|}{\alpha})$, where $d > 0$ is some constant. So there is a triangle in N that contains at most $d\alpha^2\phi(\frac{|N|}{\alpha}) \leq d\alpha^2\phi(\frac{m}{\alpha})$ cells in $\mathcal{C}$.

We remove such a triangle t from N, and recursively compute a sequence for the resulting N and append this sequence to t. By "recursively compute a sequence for N", we simply mean: find a triangle t' in the new N that contains at most $d\alpha^2\phi(\frac{m}{\alpha})$ cells in the arrangement of the new N with level at most 2α, recursively compute a sequence for $N \setminus \{t'\}$, and append to it t'. The recursion bottoms out when all triangles have been removed, that is, N becomes empty.

Now, let σ be the *reverse* of the sequence of triangles we just computed. Recall that our goal is a probabilistic algorithm that outputs a cover with at least $\log\alpha$ triangles covering any point at depth between α and 2α in F. We consider the triangles of σ in order and looking at each one we make an instant decision about adding it to our cover. When considering triangle t, we add it if t covers some point p at depth between α and 2α in F, and not adding t would result in having less than $\log\alpha$ triangles covering p in our cover. More precisely, suppose that n_1 triangles that contain p have been already added to the cover, and there are n_2 triangles that contain p and have not yet been considered; we add t if $n_1+1+n_2 \leq \log\alpha$. Let us call this a *forced addition* of t. If t is not forced, we add it with probability $\frac{c\beta}{\alpha}$, where c is a sufficiently large positive constant.

Clearly, at the end we have a cover that has at least $\log\alpha$ triangles covering each point whose depth in F is between α and 2α. Fix a triangle t. We argue that the probability that t is added to the cover is $\frac{O(\beta)}{\alpha}$. Clearly, it suffices to show that the probability of a forced addition of t is bounded by $1/\alpha$.

Consider the set N corresponding to which t makes an appearance in (the reverse of) sequence σ. So t is a triangle that contains at most $d\alpha^2\phi(\frac{m}{\alpha})$ cells in the set $\mathcal{C}$ of cells in the arrangement of N with depth at most 2α. For any cell τ in the set $\mathcal{C}$, let Z_τ denote the triangles in N containing τ, minus the triangle t itself.

If t is forced, there is a point p contained in some cell $\tau \in \mathcal{C}$ so that (a) t contains τ, (b) $|Z_\tau| \geq \alpha - \log\alpha$, and (c) less than $\log\alpha$ triangles in Z_τ have been added.

Consider a cell τ for which (a) and (b) hold. We can now bound the probability that (c) happens. Note that $|Z_\tau| \geq \alpha - \log\alpha \geq \frac{\alpha}{2}$. The probability that (c) happens is upper bounded by the probability that in a sequence of $\frac{\alpha}{2}$ independent and identical coin tosses with a probability of heads being $\frac{c\beta}{\alpha}$, less that $\log\alpha$ coins turn up heads. The expected number of heads in such a sequence is $\mu \equiv \frac{c\beta}{2}$. Since $\log\alpha \leq \beta = \max\{\log\alpha, \log\phi(m/\alpha)\}$, it suffices to bound the probability that there are fewer than β heads. For c sufficiently large, $\beta \leq \mu/2$. By applying the Chernoff bound (see for instance Theorem 4.5 (2) of [26]), we can thus bound the probability of (c) by $1/e^{c\beta/16}$, which is at most $\frac{1}{d(\alpha\phi(m/\alpha))^4}$ for c sufficiently large. Since there are at most $d\alpha^2\phi(m/\alpha)$ cells in $\mathcal{C}$ for which (a) and (b) hold, the probability of a forced addition of t is at most

$$\frac{d\alpha^2\phi(m/\alpha)}{(\alpha\phi(m/\alpha))^4} \leq \frac{1}{\alpha}.$$

Thus we have a probabilistic algorithm that outputs a set that covers points at depth between α and 2α in F at least $\log\alpha$ times, and has the property that the probability that a given triangle is added to our cover is bounded by $\frac{O(\max\{\log\alpha, \log\phi(m/\alpha)\})}{\alpha}$.

We repeat the same algorithm for points at depth between α_i and $2\alpha_i$ in F, for each $1 \leq i \leq \log_2 m$, where $\alpha_i = 2^i\alpha$. For a fixed value of i, we output a set that covers any point at such a depth at least $\log(\alpha_i)$ times; the probability that a certain triangle t is added to the set is bounded by $\frac{O(\max\{\log\alpha_i, \log\phi(m/\alpha_i)\})}{\alpha_i}$.

We output F', the union of the sets output for each i. A summation over i shows that the probability of a particular triangle being in F' is still bounded by

$$\frac{O(\max\{\log\alpha, \log\phi(m/\alpha)\})}{\alpha}.$$

$\square$

3.1 Proof of Theorem 1

The following is the algorithm that corresponds to Theorem 1. Recall that the input to this algorithm is a set T of n fat triangles and a parameter $1 \leq L \leq n$.

1. Let $F \leftarrow T$ and $\alpha \leftarrow L$.
2. While $\alpha > \log\phi(n)$:
   (a) Run the algorithm of Theorem 4 with F and α to obtain output F',
   (b) Reset $F \leftarrow F'$, and reset $\alpha \leftarrow \log\alpha$.
3. Return $T' = F$.

From the guarantee of Theorem 4, it follows that the T' that is output is a cover for all points that are L-deep with respect to T. We now bound the probability that a particular triangle t is part of T'.

Let τ denote the number of times the body of the While loop is executed. If in an iteration of this loop we have $\alpha \leq \phi(|F|/\alpha)$ in Step 2a, then this is the final iteration of the loop. This is because the next time the condition in the While statement is checked, we will have $\alpha \leq \log\phi(n)$. Thus, in all iterations but the last, we can invoke Theorem 4 to bound the probability that t is output by the algorithm in Step 2a (conditioned on t being in F) by

$$\frac{O(\max\{\log\alpha, \log\phi(|F|/\alpha)\})}{\alpha} = \frac{O(\log\alpha)}{\alpha}.$$

In the last iteration, we bound this probability simply by

$$\frac{O(\max\{\log\alpha, \log\phi(|F|/\alpha)\})}{\alpha}.$$

A simple calculation shows that the probability that $t \in T'$ is bounded above by

$$2^{O(\tau)}\frac{\log\phi(n)}{L}.$$

(Illustration: if $\tau = 3$, the probability that $t \in T'$ is bounded by

$$\frac{O(\log L)}{L} \times \frac{O(\log\log L)}{\log L} \times \frac{O(\max\{\log\log\log L, \log\phi(|F|/\log\log L)\})}{\log\log L},$$

where F denotes its value the last time Step 2a is executed.)

Now, τ is at most 3, since $\alpha \leq n$ initially and the While loop terminates when $\alpha \leq \log\phi(n)$. Thus the probability that t is output is at most $\frac{O(\log\phi(n))}{L}$.

4. WEIGHTED SET COVER

Suppose we are given a set P of k points in the plane, and a set T_1 of fat triangles, each triangle t with a positive integer weight $\mathrm{w}(t)$. The union of the triangles covers P, and we seek the minimum weight subset of T_1 that also covers P. We now describe an efficient approximation algorithm for the problem that is based on solving the natural linear programming relaxation and then applying Theorem 1 to do the rounding. As we have remarked in Section 1, the fact that an algorithm for constructing nets can be used for rounding is well known [7, 23, 16].

Since two triangles that contain the same subset of P are equivalent, we may assume that $|T_1| \leq k^6$. (This may be seen by an ad hoc argument; for a more principled derivation, we can use the methods in Chapter 10 of [24].) Let λ^* denote the weight of the optimal cover. Solving in polynomial time the linear programming relaxation of the set cover problem, we find numbers $0 \leq x_t \leq 1$ for each $t \in T_1$, so that

$$\sum_{t \in T_1 : p \in t} x_t \geq 1 \text{ for each } p \in P,$$

and $\lambda \equiv \sum_{t \in T_1} \mathrm{w}(t) x_t \leq \lambda^*$. For each t such that $x_t \geq \frac{1}{2k^6}$, we make $\left\lfloor \frac{x_t}{1/(2k^6)} \right\rfloor$ copies of it, each with weight $\mathrm{w}(t)$. For each t with $x_t < \frac{1}{2k^6}$, we make no copy (we discard it). Let T denote the resulting set of triangles. Clearly, $\mathrm{w}(T) \leq 2k^6 \lambda$. For any $p \in P$, the number of triangles in T containing p is at least

$$\begin{aligned}
\sum_{t \in T_1 : p \in t, x_t \geq \frac{1}{2k^6}} \left\lfloor \frac{x_t}{1/(2k^6)} \right\rfloor &\geq k^6 \sum_{t \in T_1 : p \in t, x_t \geq \frac{1}{2k^6}} x_t \\
&\geq k^6 \left(1 - \sum_{t \in T_1 : p \in t, x_t < \frac{1}{2k^6}} x_t \right) \\
&\geq k^6 \left(1 - \frac{|T_1|}{2k^6} \right) \\
&\geq \frac{k^6}{2}.
\end{aligned}$$

We also have $|T| \leq 2k^6 |T_1| \leq 2k^{12}$. Applying the algorithm in Theorem 1 with $L = k^6/2$, we get a cover of P whose expected weight is at most

$$\begin{aligned}
O(\log\log\log |T|) \frac{\mathrm{w}(T)}{L} &= O(\log\log\log k) \cdot \lambda \\
&= O(\log\log\log k) \cdot \lambda^*.
\end{aligned}$$

We conclude:

THEOREM 5. *Given a set P of k points in the plane, and a set T_1 of fat triangles, each with a positive weight, whose union covers P, we can, in polynomial time, compute a subset of T_1 with expected weight $O(\log\log\log k) * \lambda^*$ that covers P, where λ^* is the weight of the min-weight subset of T_1 that covers P.*

5. FRACTIONAL SET COVER PACKING

In this section, we show how the fractional set cover packing problem with fat triangles can be approximated via Theorem 1. An instance of such a fractional set cover packing problem consists of a set X of points in the plane, and a family $\mathcal{F}$ consisting of subsets induced by a set T of n fat triangles. The goal is to compute $B_1, B_2, \ldots, B_\tau$, where each $B_i \subseteq T$ covers X, together with non-negative fractions $f(B_i)$ so that $\sum_i f(B_i)$ is maximized subject to the constraint that for each $t \in T$, $\sum_{i : t \in B_i} f(B_i) \leq 1$.

Let L be the *load*, that is, the largest integer so that each $x \in X$ is in at least L of the triangles in T. The load L is clearly an upper bound on the objective function value. In Section 1, we saw that there is a solution with value $\Omega(\frac{L}{\log\log\log n})$, but this may take exponential time to construct.

We now describe a variant that runs in polynomial time and returns a solution that is nearly as good with high probability. Let N be a sufficiently large integer, to be chosen below. We run the algorithm of Theorem 1 N times independently; if B is a cover output by the algorithm, then we set

$$f(B) = \frac{\text{No. of times } B \text{ is output}}{N} * \frac{L}{c \log\log\log n},$$

where $c > 0$ is a sufficiently large constant.

We readily check that the objective function value of this solution to the fractional set cover problem is $\Omega(\frac{L}{\log\log\log n})$.

To show that the packing constraint is satisfied for each triangle, it suffices to argue that with high probability no triangle is chosen in more than $\frac{O(\log\log\log n)}{L} * N$ of the covers that are output. We choose $N = c_1 L \log n$ for sufficiently large $c_1 > 0$. Fix a triangle t. Since the probability that it is part of a single cover is $\frac{O(\log\log\log n)}{L}$, a Chernoff-bound argument implies that with probability at least $1 - \frac{1}{n^2}$, it is chosen in at most $\frac{O(\log\log\log n)}{L} * N$ of the covers. The union bound implies that with probability at least $1 - 1/n$, this event happens for every triangle in T.

THEOREM 6. *For the fractional set cover packing problem with points X and a set T of n fat triangles, there is a randomized polynomial time algorithm that with high probability outputs a solution that is within a factor of $O(\log\log\log n)$ of the load (and hence the best possible solution).*

6. CONCLUSIONS

We now state our result on quasi-uniform sampling for a family of disks in the plane.

THEOREM 7. *There is a probabilistic polynomial time algorithm that takes as input a set D of n disks and a parameter $1 \leq L \leq n$, and outputs a subset $D' \subset D$ that covers all points that are L-deep with respect to D, and has the following property: for any disk $d \in D$, the probability that it is output is at most*

$$\frac{2^{O(\log^* n)}}{L}.$$

The proof for this is almost exactly the same as that for fat triangles in Section 3: we just let the function $\phi(x) = 2$, since the version of Lemma 3 for disks has the number of cells at level at most α being $O(m\alpha)$. This in turn follows from the fact that union complexity for m disks is $O(m)$ [21].

We obtain corresponding a approximation guarantee of $2^{O(\log^* n)}$ for the weighted set cover problem and the fractional set cover packing problem with disks.

Similar remarks apply to other object families with near-linear union complexity, e.g., half-spaces in $\Re^3$. One question that remains is whether the approximation guarantee can be improved to match that of the unweighted case in [4]. Can we obtain an $O(1)$ guarantee for disks, for instance? A natural idea is to try to employ shallow cuttings as in [10], but at present, we do not know how to do this effectively.

Another interesting direction concerns the fractional set cover packing problem. Can we find interesting and natural geometric examples where the ratio of the load to the best fractional packing is not bounded by a constant? By an interesting example, we mean one for which we do not yet have super-linear lower bounds on the size of an epsilon net. Our hope is that perhaps this question is more tractable than super-linear lower bounds on epsilon nets. We note that there are already good lower bounds for *integral* set cover packing [28], but these do not work for the fractional case.

Acknowledgements. I thank Chandra Chekuri for his thought provoking remarks on the connections between the weighted set cover and fractional set cover packing problems via duality. Duality is not explicit here because the link has been made in a more elementary way. I also thank Esther Ezra and Sariel Har-Peled for very useful feedback, and the reviewers for their comments.

7. REFERENCES

[1] Z. Abrams, A. Goel, and S. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of the third international symposium on Information processing in sensor networks (ISPN)*, pages 424–432, 2004.

[2] G. Aloupis, J. Cardinal, S. Collette, S. Langerman, D. Orden, and P. Ramos. Decomposition of multiple coverings into more parts. In *Proceedings of the ACM -SIAM Symposium on Discrete Algorithms* (SODA), pages 302–310, 2009.

[3] C. Ambuhl, T. Erlebach, M. Mihalak, and M. Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In *Proceedings of APPROX-RANDOM*, pages 3–14, 2006.

[4] B. Aronov, E. Ezra, and M. Sharir. Small-size ε-nets for axis-parallel rectangles and boxes. In *Proceedings of ACM Symposium on the Theory of Computing* (STOC), pages 639–648, 2009.

[5] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proceedings of IEEE Symposium on Foundations of Computer Science* (FOCS), pages 184–193, 1996.

[6] B. Ben-Moshe, M. J. Katz, and J. S. B. Mitchell. A constant-factor approximation algorithm for optimal terrain guarding. *Proc. ACM-SIAM Symposium on Discrete Algorithms* (SODA), 515–524, 2005.

[7] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.*, 14:263–279, 1995.

[8] A. L. Buchsbaum, A. Efrat, S. Jain, S. Venkatasubramanian, and K. Yi. Restricted strip covering and the sensor cover problem. In *Proceedings of the ACM-SIAM symposium on Discrete algorithms* (SODA), 1056–1063, 2007.

[9] T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. In *Proceedings of the Symposium on Computational Geometry*, pages 333–340, 2009.

[10] C. Chekuri, K. L. Clarkson, S. Har-Peled. On the set multi-cover problem in geometric settings In *Proceedings of the Symposium on Computational Geometry*, pages 341–350, 2009.

[11] K. L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.

[12] K. L. Clarkson and P. Shor. Applications of random sampling in computational geometry, II. *Discrete and Computational Geometry*, 4:387–421, 1989.

[13] K. Clarkson and K. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete and Computational Geometry*, 37:43–58, 2007,

[14] A. Deshpande, S. Khuller, A. Malekian, and M. Toossi. Energy efficient monitoring in sensor networks. In *Proceedings of LATIN*, LNCS volume 4957, pages 436–448, 2008.

[15] K. Elbassioni, E. Krohn, D. Matijevic, J. Mestre, and D. Severdija. Improved Approximation Algorithms for 1.5D Terrain Guarding. In *Proceedings of STACS*, 361–371, 2009.

[16] G. Even, D. Rawitz, and S. Shahar. Hitting sets when the VC-dimension is small. Information Processing Letters 95(2): 358–362, 2005.

[17] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the ACM Symposium on Theory of Computing* (STOC), pages 448–455, 2003.

[18] M. R. Gibson and K. Varadarajan. Decomposing multiple coverings and the planar sensor cover problem. In *Proceedings of IEEE Symposium on Foundations of Computer Science* (FOCS), 159–168, 2009.

[19] S. Har-Peled. Being fat and friendly is not enough. *CoRR*, abs/0908.2369, 2009.

[20] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

[21] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.

[22] V. S. A. Kumar and H. Ramesh. Covering rectilinear polygons with axis-parallel rectangles. *SIAM J. Comput.*, 32(6):1509–1541, 2003.

[23] P. M. Long. Using the pseudo-dimension to analyze approximation algorithms for integer programming. In *Proceedings of the 7th Workshop on Algorithms and Data Structures*, LNCS vol. 2125, 26–37, 2001.

[24] J. Matoušek. *Lectures on Discrete Geometry.* Springer-Verlag, 2002.

[25] J. Matoušek, J. Pach, M. Sharir, S. Sifrony, and E. Welzl. Fat triangles determine linearly many holes. *SIAM J. Comput.*, 23:154–169, 1994.

[26] M. Mitzenmacher and E. Upfal. *Probability and Computing.* Cambridge University Press, 2005.

[27] N. H. Mustafa and S. Ray. PTAS for geometric hitting set problems via local search. In *Proceedings of the Symposium on Computational Geometry*, pages 17–22, 2009.

[28] J. Pach, G. Tardos, and G. Toth. Indecomposable coverings. In *Proceedings of the China-Japan joint Conference on Discrete Geometry, Combinatorics, and Graph Theory* (CJCDGCGT 05), LNCS vol. 4381, 135–148, 2007.

[29] S. Pandit, S. V. Pemmaraju, and K. Varadarajan. Approximation algorithms for domatic partitions of unit disk graphs. In *Proceedings of APPROX-RANDOM*, pages 312–325, 2009.

[30] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the ACM Symposium on Theory of Computing* (STOC), pages 475–484, 1997.

[31] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, New York, 1995.

[32] K. Varadarajan. Epsilon nets and union complexity. In *Proceedings of the Symposium on Computational Geometry*, pages 11–16, 2009.

[33] V. V. Vazirani. *Approximation Algorithms.* Springer, 2001.