Geometric Packing under Non-uniform Constraints*

Alina Ene[†] Sariel Har-Peled[‡] Benjamin Raichel[§]

August 16, 2016

Abstract

We study the problem of discrete geometric packing. Here, given weighted regions (say in the plane) and points (with capacities), one has to pick a maximum weight subset of the regions such that no point is covered more than its capacity. We provide a general framework and an algorithm for approximating the optimal solution for packing in hypergraphs arising out of such geometric settings. Using this framework we get a flotilla of results on this problem (and also on its dual, where one wants to pick a maximum weight subset of the points when the regions have capacities). For example, for the case of fat triangles of similar size, we show an $O(1)$-approximation and prove that no PTAS is possible.

1 Introduction

Motivation and examples. Consider the problem of ***obnoxious facility location*** [Tam91, Cap99]; that is, you have to place several facilities, but these facilities are undesired (i.e., obnoxious). Facilities of this type include nuclear reactors, wind farms, airports, power plants, factories, prisons, etc. Facilities can also be semi-desirable – a customer might want to have supermarkets close to their home, but they do not want to have too many of them close by as they increase traffic, noise, etc. One natural way to model this geometrically is to associate each obnoxious facility with its region of undesirability. We also have customers (modeled as points), and each customer has a threshold of how many obnoxious facilities it is willing to accept covering it. Different customers may have different thresholds, for example because more affluent people have stronger political power and it is harder to place obnoxious facilities near their homes.

Naturally, if you allow only a single region to cover each customer, then this is a classical packing problem, and much work has been done on packing disks/balls [SMC+07]. However, there are many cases where allowing limited interaction between the packed regions is allowed (after all, these facilities are

*A preliminary version of this paper appeared in SoCG 2012. The full version of the paper is available from the arXiv [EHR11].

[†]Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; ene1@uiuc.edu; http://www.cs.uiuc.edu/~ene1/. Work on this paper was partially supported by NSF grants CCF-0728782 and CCF-1016684.

[‡]Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sariel@uiuc.edu; http://www.uiuc.edu/~sariel/. Work on this paper was partially supported by a NSF AF awards CCF-0915984, CCF-1421231, and CCF-1217462.

[§]Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; raichel2@uiuc.edu; http://www.cs.uiuc.edu/~raichel2. Work on this paper was partially supported by a NSF AF award CCF-0915984.

required for modern existence). As a concrete example of this type of problem, consider the placement of radio stations/cellphone towers. While airports allow only very limited levels of interference[1], higher levels of such interference is acceptable in residential neighborhoods. However, at a certain point there is going to be resistance to placing more wireless towers in residential areas, as these towers are viewed as causing cancer (this fear might be baseless, but it does not change the political reality of the difficulty of placing such towers). On the other hand, there is little resistance to placing such towers along highways in sparsely populated areas.

In this paper, we are interested in the modeling of such problems and in the computation of an efficient approximation to the optimal solution of such problems.

Modeling.

As hinted by the above, perhaps the most natural way to model this problem is as a generalization of the well known independent set problem.

Independent set. Computing an independent set in a graph is a fundamental discrete optimization problem. Unfortunately, it is not only computationally hard, but it is even hard to approximate to within a factor of $n^{1-\varepsilon}$, for any constant $\varepsilon > 0$ [Has99] (under the assumption that $\mathsf{NP} \neq \mathsf{P}$). Surprisingly, the problem is considerably easier in some geometric settings. For example, there is a PTAS[2] [Cha03, EJS05] for the following problem: Given a set of unit disks in the plane, find a maximum cardinality subset of the disks whose interiors are disjoint. Furthermore, a simple local search algorithm yields the desired approximation: For any $\varepsilon > 0$, the local search algorithm that tries to swap subsets of size $O(1/\varepsilon^2)$ yields a $(1 - \varepsilon)$-approximation in $n^{O(1/\varepsilon^2)}$ time [CH09, CH11].

The discrete independent set problem. In this paper, we consider packing problems in geometric settings that are natural extensions of the geometric independent set problem described above. As a starting point, motivated by practical applications, we consider the discrete version of the geometric independent set problem in which, in addition to a set of weighted regions, we are given a set of points, and the goal is to select a maximum weight subset of the regions so that each point is contained in at most one of the selected regions. We refer to this problem as the ***discrete independent set*** problem. Chan and Har-Peled [CH11] studied this discrete variant and proved that one can get a good approximation if the union complexity of the regions is small.

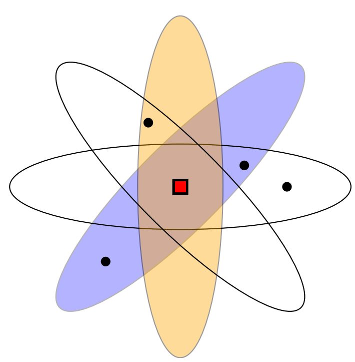

Figure 1: Flower

[1]See http://tinyurl.com/7td67v3 for a story of an airport closing down because of radio interference.
[2]Polynomial time approximation scheme.

Note that the discrete independent set problem captures the continuous version of the independent set problem, since we can place a point in each face of the induced arrangement of the given regions. Furthermore, the discrete version is considerably harder in some cases than the continuous variant. The challenge is that several regions forming a valid solution to an instance of a discrete independent set problem may contain a common point that is not part of the set of points given as input. Figure 1 shows an example in which the middle point, marked as a square, is covered twice by the given valid solution.

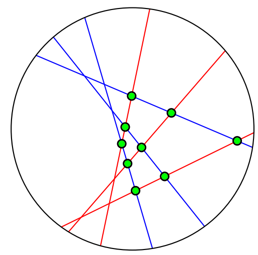

Figure 2

To illustrate the difference in difficulty, consider the case when the input consists of a set S of segments (in general position) with their endpoints on a circle, such that for every pair of segments, its members intersect. Clearly, in the continuous version, the maximum independent set of segments is a single segment. However, in this case, the discrete version captures the graph independent set problem. More precisely, given any instance of independent set on a graph $G = (V, E)$, every vertex $v \in V$ is mapped to a segment s_v of S, and every edge $uv \in E$, is mapped to the point $\mathsf{s}_u \cap \mathsf{s}_v$ (which is added to a set of points P). Clearly, an independent set of segments of S (in relation to the point set P) corresponds to an independent set in G. For example, Figure 2 depicts the resulting instance encoding independent set for $K_{3,3}$.

The packing problem. In this paper, we are interested in the natural extension of the discrete independent set problem to the case where every point has a capacity and might be covered several times (but not exceeding its capacity). The resulting problem has a flavor of a packing problem, and is defined formally as follows.

Problem 1.1. **(PackRegions)** *Given a set $\mathcal{D}$ of regions and a set P of points such that each region r has a weight $w(r)$ and each point p has a capacity $\#(\mathsf{p})$, find a maximum weight subset $X \subseteq \mathcal{D}$ of the regions such that, for each point p, the number of regions in X that contain p is at most its capacity $\#(\mathsf{p})$.*

We emphasize that different points might have different capacities, which makes the problem considerably more challenging to solve than the unit capacities case (i.e., the discrete independent set problem). We also consider the following dual problem in which the points have weights and the regions have capacities.

Problem 1.2. **(PackPoints)** *Given a set $\mathcal{D}$ of regions and a set P of points such that each region r has a capacity $\#(r)$ and each point p has a weight $w(\mathsf{p})$, find a maximum weight subset $X \subseteq \mathsf{P}$ of the points such that each region r contains at most $\#(r)$ points of X.*

Hypergraph framework. These two problems can be stated in a unified way in the language of hypergraphs[3]. Given an instance of PackRegions, we construct a hypergraph as follows: Each weighted

[3] A ***hypergraph*** G is a pair (V, E), where V is a set of vertices and E is a collection of subsets of V which are called ***hyperedges***.

region is a vertex, and all the regions containing a given point of capacity k become a hyperedge (consisting of these regions) of capacity k. A similar reduction works for PackPoints, where the given weighted points are the vertices, and each region of capacity k becomes a hyperedge of capacity k consisting of all of the points contained in this region. Therefore the previous two problems are special cases of the following problem.

Problem 1.3. (HGraphPacking) *Given a hypergraph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ with a weight function $w(\cdot)$ on the vertices and a capacity function $\#(\cdot)$ on the hyperedges, find a maximum weight subset $X \subseteq \mathsf{V}$, such that $\forall f \in \mathsf{E}$ we have $|X \cap f| \leq \#(f)$.*

We will be interested primarily in hypergraphs with certain hereditary properties. A hypergraph property is *hereditary* if the sub-hypergraph induced by any subset of the vertices has the property; an example of a hereditary property of hypergraphs is having bounded VC dimension. Roughly, we are interested in hypergraphs having the ***bounded growth property***: For any induced sub-hypergraph on t vertices the number of its hyperedges that contain exactly k vertices is near linear in t and its dependency on k is bounded by $2^{O(k)}$, see Definition 2.2. Such hypergraphs arise naturally when considering points and "nice" regions in the plane.

Our results.

- Main result. Our main result is an algorithm that provides a good approximation for HGraphPacking as a function of the growth of the hypergraph, see Theorem 3.11. Our result can be viewed as an extension of the work of Chan and Har-Peled [CH11] to these more general and intricate settings. For simplicity, we focus on linear weight functions; we show in Section 3.5 that our main result extends to the case in which the weight function is a non-negative submodular function.
- Regions with low union complexity. In Section 4, we apply our main result to regions that have low union complexity, and we get the following results:
  (A) If the union complexity of n regions is $O(nu(n))$ then the algorithm returns an $O\big(u(n)^{1/\chi}\big)$-approximation for PackRegions, where χ is the minimum capacity of any point in the given instance. (That is, the problem becomes easier as the minimum capacity increases.) For the case where all the capacities are one, this is the discrete independent set problem, and our algorithm specializes to the algorithm of Chan and Har-Peled [CH11], which gives an $O(u(n))$-approximation.
  (B) As a special case, we get a constant factor approximation for PackRegions if the union complexity of the regions is linear. This holds for (i) fat-triangles of similar size, (ii) unit axis-parallel cubes in 3d, and (iii) pseudo-disks. See Corollary 4.3.
  (C) Similarly, since the union complexity of fat triangles in the plane is $O(n \log^* n)$ [EAS11, AdBES14], we get an $O\Big(\big(\log^* n\big)^{1/\chi}\Big)$ approximation for such instances of PackRegions.
- Bi-criteria approximation. Our main result also implies a bi-criteria approximation algorithm. That is, we can improve the quality of the solution, at the cost of potentially violating low capacity regions. Formally, if the input instance $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ of HGraphPacking has at most $F_k(t) = 2^{O(k)} F(t)$ edges of size k when restricted to any subset of t vertices, then for any integer $\phi \geq 1$, our algorithm yields an $\big(O\big((F(n)/n)^{1/\phi}\big), \phi\big)$-approximation to the given instance G of HGraphPacking. Specifically, the value of the generated solution X is at least $\Omega\big(\mathrm{opt}/(F(n)/n)^{1/\phi}\big)$, where opt is the value of the optimal solution, and for every hyperedge $f \in \mathsf{E}$, we have $|f \cap X| \leq \max(\phi, \#(f))$.

As an example, for any set of n regions in the plane such that the boundaries of any pair of them intersect $O(1)$ times, the above implies that one can get an $\left(O\left(n^{1/\phi}\right), \phi\right)$-approximation for PackRegions.

- Axis-parallel boxes. The union complexity of axis-parallel rectangles can be as high as quadratic, and therefore we cannot immediately apply our main result to get a good approximation. Instead, we decompose the union of axis-parallel rectangles into regions of low union complexity, and this decomposition together with our main result gives us an $O(\log n)$ approximation for instances of PackRegions in which the regions are axis-parallel rectangles in the plane (see Lemma 4.10).
  A more involved analysis also applies to the three dimensional case, where we get an $O(\log^3 n)$ approximation for PackRegions for axis parallel boxes (see Lemma 4.12).
- Dual problem. We show in Section 4.2 that, by standard lifting techniques, we can apply our result for PackRegions, where the regions are disks, to the dual problem of PackPointsInDisks. However, for other regions, the dual problem PackPoints seems to be more challenging. Specifically, this is true for the case of axis-parallel rectangles. For this case, we first provide a constant factor approximation for *baseline* instances of the problem; a baseline instance is a set of rectangles that lie on the x-axis. Interestingly, if the set of rectangles is defined in relation to a set of points (and each rectangle contains only a few points), then one can define a near-linear (in the number of points) sized set of rectangles such that each original rectangle is the union of two new rectangles. Combining this with the baseline result and a sparsifying technique, we get an $(O(\log n), 2)$-approximation; that is, every rectangle b contains at most $\max(2, \#(\mathsf{b}))$ points of the solution constructed, and the total weight of the solution is $\Omega(\mathrm{opt}/\log n)$ (see Theorem 4.22). (Note that, by applying our general framework directly to this setting, we only get an $\left(O\left(n^{1/\phi}\right), \phi\right)$-approximation, for any integer $\phi > 0$.)
- Packing points into fat triangles. We provide a polylog bi-criteria approximation for the problem of packing points into fat triangles. This requires proving that one can compute, for a given point set, a small number of canonical subsets such that the point set covered by any fat-triangle (if the set is sufficiently small) is the union of a constant number of these canonical subsets. Proving this requires non-trivial modifications of the result of Aronov *et al.* [AES10]. In addition, we show that a measure defined over a fat triangle can be covered by a few fat triangles, each one of them containing only a constant fraction of the original measure. We believe these two results are of independent interest. Plugging this into the machinery previously developed for axis parallel rectangles yields the new approximation algorithm. See Section 5 for details.
- PTAS for disks and planes. We adapt the techniques of Mustafa and Ray [MR10] in order to get a PTAS for instances consisting of unweighted disks and unit-capacity points: we lift the problem to 3d, we construct an approximate conflict graph (as done by Mustafa and Ray), and we use a local search algorithm. This result also implies a PTAS for PackPoints for unweighted points and uniform capacity halfspaces in $\mathbb{R}^3$. See Section 6 for the details.
- Hardness. We show some hardness results for our problems. In particular, we show that PackPoints for fat triangles in the plane is as hard as independent set in general graphs (see Lemma 7.2). We also show that PackRegions is APX-hard (and thus there is no PTAS) for similarly sized fat triangles in the plane (thus "matching" the result of Corollary 4.3).

Main technical contribution. In addition to the results mentioned above, our work further develops and extends the techniques for rounding linear programming relaxations for geometric packing problems. Our algorithms use the randomized-rounding-with-alteration technique to round a fractional solution rising out of a natural LP relaxation; this technique has been used in much more general settings [Sri01]. The rounding uses the fractional solution to construct a random sample of the regions. The sampled regions might not form a feasible solution and therefore we need to pick a feasible subset of the sample. This step typically involves selecting an ordering in which to consider the sampled regions and greedily picking a feasible subset based on the ordering [CH09, CCH09, Var10]. The main technical difficulty lies in finding the ordering of the regions. The main idea behind previous approaches is to build a conflict graph and argue that there exists a vertex of low degree; this vertex gives us a region that is a good candidate for the last region in the ordering, and we can recursively consider the remaining regions. We use a similar approach to construct an ordering, but the conflicts that are relevant in our settings are more complicated. For example, in the geometric independent set problem, a conflict is a pair of regions that overlap. However, in our setting, a conflict involves a larger number of regions and therefore we need to consider a conflict hypergraph. We show that there exists a vertex of low degree in this hypergraph and therefore we are able to extend some of the previous approaches to this more general setting. Unsurprisingly, this extension requires a more involved analysis.

Previous work. The work that is closest to ours is the paper of Chan and Har-Peled [CH09] which addresses an easier special case of the problems we consider. Fox and Pach [FP11] presented an n^{ε} approximation for independent set for segments in the plane. The usage of LP relaxations for approximating such problems is becoming more popular. In particular, Chalermsook and Chuzhoy [CC09] use a natural LP relaxation to get an $O(\log \log n)$-approximation for independent set of axis parallel rectangles in the plane. The geometric set cover problem and the more general problem, the geometric set multi-cover problem, have approximation algorithms that use ε-nets to round the natural LP relaxation; see [CCH09] and references therein. Chan and Har-Peled [CH09] used local search to get a PTAS for independent set of pseudo-disks. Independently, Mustafa and Ray [MR10] used similar ideas to get a PTAS for hitting set of pseudo-disks in the plane. There is not much work on the hardness of optimization problems in the geometric settings we are interested in. [CC07] shows that the problem of independent set of axis-parallel boxes in three dimensions is APX-hard (the problem is known to be NP-Hard in the plane). See also [GC11, Har09] and references therein for some recent hardness results. Naturally, in non-geometric settings, there is a vast literature on the problems and techniques we use, see [WS11]. As we already mentioned, our algorithms use the randomized-rounding-with-alteration technique to round a fractional solution. This technique was used in [Sri01] to find an approximate solution to packing integer programs (PIPs) of the form $\left\{ \max wx \;\middle|\; Ax \leq b, x \in \mathbb{Z}_+^n \right\}$, where A is a matrix whose entries are either 0 or 1. The approximation guarantee given in [Sri01] is $O(n^{1/B})$, where $B = \min_i b_i$ (this is similar in spirit to our bound that depends on the minimum capacity).

Organization. In Section 2 we define the problem and the associated LP relaxation, and describe some basic tools used throughout the paper. In Section 3 we present the approximation algorithm for the hypergraph case. In Section 3.5, we extend our main result to the case in which the weight function is a submodular function. In Section 4 we present various applications of our main result. In Section 5 we present the algorithm for packing points into fat triangles. In Section 6 we present a PTAS for some restricted cases. In Section 7 we present some hardness results. We conclude in Section 8.

2 Preliminaries

Definition 2.1. *For a maximization problem, an algorithm provides an* **α-approximation** *if it outputs a solution of value at least* opt/α*, where* opt *is the value of the optimal solution. An* **(α, β)-approximation** *algorithm for* HGraphPacking *is an algorithm that returns a (potentially infeasible)* *solution of value at least* opt/α *such that each hyperedge* f *contains at most* $\max(\#(f), \beta)$ *vertices of the solution.*

α-fat triangles. For $\alpha \geq 1$, a triangle $\triangle$ is ***α-fat*** if the ratio between its longest edge and its height on this edge is bounded by α (there are several equivalent definitions of this concept). A set of triangles is α-fat if all the triangles in the set are α-fat. The union complexity of n α-fat triangles is $O(n \log^* n)$ [EAS11, AdBES14] (the constant in the O depends on α, which is assumed to be a constant).

2.1 LP Relaxation and the Rounding Scheme

We consider the following natural LP relaxation for the HGraphPacking problem. For each vertex v, we have a variable x_v with the interpretation that x_v is 1 if v is selected, and 0 otherwise. For each hyperedge f, we have a constraint that enforces that the number of vertices of f that are selected is at most the capacity of f.

$$\textsc{Hypergraph-LP}: \quad \max \sum_{v \in \mathsf{V}} w_v x_v$$
$$\sum_{v \in f} x_v \leq \#(f) \qquad \forall f \in \mathsf{E}$$
$$0 \leq x_v \leq 1 \qquad \forall v \in \mathsf{V}.$$

The ***energy*** of a subset $X \subseteq \mathsf{V}$ is $\mathcal{E}(X) = \sum_{v \in X} x_v$. In the following, $\mathcal{E}$ denotes the ***energy*** of the LP solution; that is $\mathcal{E} = \mathcal{E}(\mathsf{V}) = \sum_{v \in \mathsf{V}} x_v$. Note that the energy is at most the number of vertices of the hypergraph. Also, we assume that $\mathcal{E} \geq 1$ (which is always true since all the capacities are at least one).

The minimum capacity of a packing instance is a useful measure of how hard the instance is; formally, the ***minimum capacity*** of a given instance G is

$$\chi = \chi(\mathsf{G}) = \min_{f \in \mathsf{E}} \#(f). \tag{1}$$

Let $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ be a hypergraph, and let $X \subseteq \mathsf{V}$ be a subset of its vertices. The sub-hypergraph of G ***induced*** by X is $\mathsf{G}_X = \left(X, \left\{ f \cap X \;\middle|\; f \in \mathsf{E} \right\}\right)$.

Definition 2.2. *Let* $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ *be a hypergraph. For any integer* k*, let* $F_k(\cdot)$ *denote the function*

$$F_k(t) = \max_{X \subseteq \mathsf{V}, |X| \leq t} \left| \left\{ f \;\middle|\; f \in \mathsf{E} \text{ and } |X \cap f| = k+1 \right\} \right|;$$

that is, $F_k(t)$ *is the maximum number of hyperedges of size* $k+1$ *of a sub-hypergraph of* G *that is induced by a subset of at most* t *vertices. We say that* G *has the* **bounded growth** *property if the following conditions are satisfied:*

(A) There exists a non-decreasing function $\gamma(\cdot)$ *such that* $F_k(t) \leq 2^{O(k)} t \gamma(t)$ *for any* k *and* t*.*

(B) There exists a constant c *such that* $F_k(xt) \leq c F_k(t)$ *for any* t, k *and* x *such that* $1 \leq x \leq 2$*.*

This notion of bounded growth is a hereditary property of the hypergraph, and it is somewhat similar to the bounds on the size of set systems with bounded VC dimension. Hypergraphs with bounded growth arise naturally in geometric settings.

Running Example 2.3. To keep the presentation accessible, consider an instance of PackRegions in which the regions are disks. Specifically, we are given a weighted set of disks $\mathcal{D}$ and set of points P with capacities. The hypergraph has a vertex for each disk in $\mathcal{D}$ and a hyperedge for each point $\mathsf{p} \in \mathsf{P}$; the hyperedge f_{p} consists of the vertices corresponding to all disks of $\mathcal{D}$ that contain p.

In this case, the quantity $F_k(t)$ (see Definition 2.2 above) is bounded by the number of faces in an arrangement of t disks that have depth exactly $k+1$. Since the union complexity of t disks is linear, a standard application of the Clarkson technique [Cla88, CS89] implies that $F_k(t) = O(kt)$. Thus in this case we have $\gamma(t) = O(1)$.

2.2 Basic tools

The following lemma demonstrates that the packing problem can be solved in a straightforward fashion if all the capacities are the same (i.e., uniform capacities). This is done by repeatedly applying a procedure to find and remove a "heavy" independent set in the remaining induced sub-hypergraph (for example, one can use the algorithm of Chan and Har-Peled [CH11] to compute such an independent set).

Lemma 2.4. *Let* G *be a hypergraph for which there is a polynomial time algorithm* `alg` *that takes as input a fractional solution to* Hypergraph-LP *for an* HGraphPacking *instance on* G*, or any induced subgraph of* G*, with unit capacities — i.e., an independent set instance — and it constructs an integral solution whose value is at least an* α *fraction of the value of the fractional solution. Then one can compute in polynomial time a* $(\alpha+1)$*-approximation for any instance of* HGraphPacking *on* G *with uniform capacities (see Definition 2.1) .*

Proof: Let $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ be an instance of HGraphPacking in which all hyperedges have the same capacity, say k. Let $\mathsf{G}_0 = \mathsf{G}$, and in the ith iteration, for $i = 1, \ldots, k$, compute a maximum weight independent set Y_i in G_{i-1} using `alg`, and let $\mathsf{G}_i = \mathsf{G}_{\mathsf{V} \setminus U_i}$, where $U_i = Y_1 \cup \cdots \cup Y_i$. We claim that U_k is the required approximation.

Clearly, no hyperedge of G contains more than k vertices of U_k as it is the union of k independent sets, and as such it is a valid solution. Now, let V_{opt} be the optimal solution. If $w(V_{\mathrm{opt}} \cap U_k) \geq w(V_{\mathrm{opt}})/(\alpha+1)$ then we are done. Otherwise, we have that $w(V_{\mathrm{opt}} \setminus U_k) \leq (1 - \frac{1}{\alpha+1}) w(V_{\mathrm{opt}}) = \frac{\alpha}{\alpha+1} w(V_{\mathrm{opt}})$.

Next, consider the hypergraph G_{i-1}, and observe that $V_{\mathrm{opt}} \setminus U_{i-1}$ is a valid solution for HGraphPacking for this graph (with uniform capacities k). Interpreting this integral solution as a solution to the LP, and scaling it down by k, we get a fractional solution to the independent set LP of this hypergraph of value $w(V_{\mathrm{opt}} \setminus U_{i-1})/k$. Since Y_i was constructed using `alg` on $\mathrm{opt}_{\mathrm{LP}}(\mathsf{G}_{i-1})$, the optimal fractional solution to the independent set LP of this hypergraph,

$$w(Y_i) \geq \frac{\mathrm{opt}_{\mathrm{LP}}(\mathsf{G}_{i-1})}{\alpha} \geq \frac{w(V_{\mathrm{opt}} \setminus U_{i-1})}{k\alpha} \geq \frac{w(V_{\mathrm{opt}} \setminus U_k)}{k\alpha} \geq \frac{\alpha w(V_{\mathrm{opt}})}{(\alpha+1)k\alpha} \geq \frac{w(V_{\mathrm{opt}})}{k(\alpha+1)},$$

which implies that $w(U_k) = w(Y_1) + \cdots + w(Y_k) \geq w(V_{\mathrm{opt}})/(\alpha+1)$. ■

A hypergraph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ ***shatters*** $X \subseteq \mathsf{V}$ if the number of hyperedges in G_X is $2^{|X|}$. The **VC *dimension*** of G is the size of the largest set of vertices it shatters.

The following is a "sparsification" lemma, which will later be used in Section 4 and Section 5. Here we get better bounds than the standard technique, as we are using stronger sampling results known for spaces with bounded VC dimension.

Lemma 2.5. *Let* $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ *be an instance of* HGraphPacking *with VC dimension* d*, and consider its fractional LP solution of value* opt *and with energy* $\mathcal{E}$*. Then, one can compute, in polynomial time, a valid fractional solution for the LP of* G *such that:*

(A) *The value of the new fractional solution is* $\geq \mathrm{opt}/12$*.*
(B) *The number of vertices with non-zero value is* $O(d\mathcal{E}\log\mathcal{E})$*.*
(C) *The value of each non-zero variable is equal to* i/M *for some integer* $i \leq M$*, where* $M = O(d\log\mathcal{E})$*.*
(D) *The total energy in the new solution is* $\Theta(\mathcal{E})$*.*

Proof: Let $\varepsilon = 1/\mathcal{E}$, where $\mathcal{E} = \sum_v x_v$, and x_v is the value the LP assigns to $v \in \mathsf{V}$ in the optimal LP solution. Let $T = O(d\log\mathcal{E})$ and Υ be a random sample of V of (expected) size $\tau = \mathcal{E}T = O((d/\varepsilon)\log(1/\varepsilon))$, created by picking each vertex v independently with probability $x_v \cdot T$[4]. This sample is a relative $(\varepsilon, 1/2)$-approximation [Har11, HS11][5], with probability of failure $\leq \rho_1 = \varepsilon^{O(d)}$. That implies that for any hyperedge $f \in \mathsf{E}$ such that $x(f) = \sum_{v\in f} x_v$ we have

$$\frac{|\Upsilon \cap f|}{|\Upsilon|} \leq (1 + 1/2)\left(\frac{x(f)}{\mathcal{E}} + \varepsilon\right).$$

To interpret the above, observe that $\mathbf{E}\big[|\Upsilon \cap f|\big] = x(f)T$ and $\tau = \mathbf{E}\big[|\Upsilon|\big] = \mathcal{E}T$, as such, a rough estimate of the expectation of $|\Upsilon \cap f| \,/\, |\Upsilon|$ is $x(f)/\mathcal{E}$. Thus, the above states (somewhat opaquely) that no hyperedge is being over-sampled by Υ. Since the expected size of Υ is τ, by Chernoff's inequality [MR95], we have $\mathbf{Pr}\big[|\Upsilon| \geq 2\tau\big] \leq \exp(-\tau/4) = \varepsilon^{O(d/\varepsilon)}$. Arguing in a similar fashion, we have that $\mathbf{Pr}[|\Upsilon| \leq \tau/2] \leq \varepsilon^{O(d/\varepsilon)}$. That is $\tau/2 \leq |\Upsilon| \leq 2\tau$ with probability at least $1 - \rho_2$, where $\rho_2 = \varepsilon^{O(d\mathcal{E})}$ (as $\mathcal{E} \geq 1$).

Now, consider a hyperedge f with capacity k, and observe that $x(f) \leq k$. Therefore, $|\Upsilon \cap f| \leq (1 + 1/2)(x(f)/\mathcal{E} + \varepsilon)\,|\Upsilon| \leq (3/2)(k+1)\varepsilon 2\tau \leq 6kT$. In particular, if v appears t_v times in Υ (Υ is a multiset), then we assign it the fractional value $y_v = t_v/6T$. We then have that $y(f) = \sum_{v\in f} y_v \leq |\Upsilon \cap f| \,/6T \leq k$ (and this holds for all hyperedges with probability $\geq 1 - \rho_1$). As such, the fractional solution defined by the y's is valid.

As for the value of this fractional solution, consider the random variable $Z = \sum_v y_v w(v)$, which is a function of the random sample Υ. Observe that

$$\mathbf{E}\Big[Z\Big] = \sum_v w(v)\,\mathbf{E}\left[\frac{t_v}{6T}\right] = \sum_v \frac{w(v)}{6T}\,\mathbf{E}\Big[t_v\Big] = \sum_v \frac{w(v)}{6T} x_v T = \frac{1}{6}\sum_v w(v)x_v = \frac{\mathrm{opt}_{\mathrm{LP}}}{6},$$

as $\mathrm{opt}_{\mathrm{LP}} = \sum_v w(v)x_v$. In particular, since no vertex can have $w(v) > \mathrm{opt}_{\mathrm{LP}}$ (otherwise, we would choose it as the solution), it follows that

$$Z = \sum_v y_v w(v) \leq \mathrm{opt}_{\mathrm{LP}} \sum_v \frac{t_v}{6T} \leq \mathrm{opt}_{\mathrm{LP}} \frac{|\Upsilon|}{6T} \leq \mathrm{opt}_{\mathrm{LP}} \frac{2\tau}{6T} = \mathrm{opt}_{\mathrm{LP}} \frac{2\mathcal{E}T}{6T} \leq \frac{\mathcal{E}}{3}\mathrm{opt}_{\mathrm{LP}}.$$

[4] A minor technicality is that $x_v T$ might be larger than one. In this case, we put $\lfloor x_v T \rfloor$ copies of v into Υ, and we put an extra copy of v into Υ with probability $x_v T - \lfloor x_v T \rfloor$. It is straightforward to verify that our argumentation goes through in this case. Observe that such large values work in our favor by decreasing the probability of failure.

[5] Somewhat imprecisely, a ***relative*** (ε, δ)***-approximation*** is a sample that approximates up to a multiplicative error of $(1 \pm \delta)$ any range that is heavier than ε fraction of the ground set. We avoid the formal definition for the sake of simplicity. Observe, however, that in the text we use a weaker version of (one side of) this relative approximation property.

This implies that $\mathbf{Pr}\big[Z \geq \mathrm{opt}_{\mathrm{LP}}/12\big] \geq 1/4\mathcal{E} = \varepsilon/4$. Indeed, if not,

$$\begin{aligned}\mathbf{E}\big[Z\big] &\leq \frac{\mathrm{opt}_{\mathrm{LP}}}{12}\,\mathbf{Pr}\left[Z \leq \frac{\mathrm{opt}_{\mathrm{LP}}}{12}\right] + \mathbf{Pr}\left[Z \geq \frac{\mathrm{opt}_{\mathrm{LP}}}{12}\right]\frac{\mathcal{E}}{3}\mathrm{opt}_{\mathrm{LP}} \\ &< \frac{\mathrm{opt}_{\mathrm{LP}}}{12} + \frac{1}{4\mathcal{E}}\cdot\frac{\mathcal{E}}{3}\mathrm{opt}_{\mathrm{LP}} \leq \frac{\mathrm{opt}_{\mathrm{LP}}}{6},\end{aligned}$$

a contradiction. Therefore, a random sample Υ corresponds to a valid assignment with value at least $\mathrm{opt}_{\mathrm{LP}}/12$ with probability at least $\mathbf{Pr}[Z \geq \mathrm{opt}_{\mathrm{LP}}/12] - \rho_1 - \rho_2 \geq \varepsilon/8$, as $\rho_1 + \rho_2$ is an upper bound on the sample Υ failing to have the desired properties. As such, taking $u = O(\mathcal{E}\log\mathcal{E})$ independent random samples one of them is the required assignment, with probability $\geq 1 - (1-\varepsilon/8)^u \geq 1 - 1/\mathcal{E}^{O(1)}$. We take this good sample together with its associated LP values as the desired fractional solution to the LP. Also, note that the total energy of the new solution is $\Theta(\mathcal{E})$. ∎

3 Approximate packing for hypergraphs

In this section, we present the algorithm for computing a packing for a given hypergraph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$. We assume that $|\mathsf{E}|$ is polynomial in $|\mathsf{V}|$ and that G has the bounded growth property introduced in Definition 2.2 (both properties hold for the hypergraphs arising in geometric settings). Let x be an optimal solution to the Hypergraph-LP relaxation described in Section 2.1.

3.1 The algorithm

We round the fractional solution to an integral solution using a standard randomized-rounding-with-alteration approach. The first step is to choose an appropriate ordering of the vertices. We describe later how to choose a good ordering. For now, assume that the ordering is given. The rounding then proceeds in two phases, the ***selection phase*** and the ***alteration phase***. In the selection phase, we pick a random sample $\mathcal{R}$ of the vertices by selecting each vertex v independently at random with probability x_v/Δ, where

$$\Delta = \alpha\left(\gamma(\mathcal{E})\right)^{1/\chi}, \tag{2}$$

α is a sufficiently large constant, and χ is the minimum capacity in the given instance (see Eq. $(1)_{\mathrm{p7}}$ and Definition 2.2). In the alteration phase, a subset of $\mathcal{R}$ is picked as follows: Consider the sampled vertices in the given ordering and add the current vertex to the solution (starting with the empty set) if the resulting solution remains feasible. A vertex is ***selected*** if it is present in the sample, and it is ***accepted*** if it is present in the solution. The main insight is that, by the bounded growth property of the hypergraph, there is an ordering such that each vertex is accepted with constant probability, provided that it is selected. This will immediately imply that the algorithm achieves an $O(\Delta)$-approximation.

3.1.1 Constructing a good ordering

The main challenge is to prove that a good ordering for the alteration phase exists – which is addressed next. This suggests a natural brute force algorithm that runs in $O\big(n^{\overline{\chi}+O(1)}\big)$ time, to actually compute this good ordering, where $\overline{\chi}$ is the maximum capacity of an edge in the given instance. Section 3.3 show how one can remove the exponential dependency on $\overline{\chi}$, thus making the running time polynomial in the input size.

Before describing how to construct a good ordering of the vertices, it is useful to understand what will force a vertex to be rejected in the alteration phase. With this goal in mind, consider an ordering of the vertices. Let $\mathcal{R}$ be a sample of the vertices in V such that each vertex v is in $\mathcal{R}$ independently at random with probability x_v/Δ. Let v be a vertex in $\mathcal{R}$. When v is considered in the alteration phase, it is rejected if and only if there exists a hyperedge f of capacity $\#(f)$ such that f contains v and we have already accepted $\#(f)$ vertices of f.

Observation 3.1. *The above event, that the algorithm already accepted $\#(f)$ vertices of f is difficult to analyze. However, one can settle for a more conservative analysis that upper bounds the probability that v is rejected, given that* all *of the vertices in $\mathcal{R}$ that appear before v in the ordering are accepted. (In the alteration phase, it is possible that not all vertices in $\mathcal{R}$ that appear before v will be accepted, but this can only help.) Since we are only interested in the event that $\mathcal{R}$ contains $k+1$ vertices — the vertex v and k other vertices that appear before v in the ordering — that are contained in a hyperedge of capacity k, only the set of vertices that appear before v in the sample matter, and not the actual ordering of the vertices. (This idea is used below in the proof of Lemma 3.2.)*

With this observation in mind, we define a ***k-conflict*** to be a set of $k+1$ vertices that are contained in a hyperedge of capacity k. In the following, $\mathcal{H}_k$ denotes the set of all k-conflicts, and $\mathcal{H} = \cup_k \mathcal{H}_k$ denotes the set of all conflicts. The quantity of interest is the probability of the event that all of the vertices of a k-conflict, h, are present in the sample, and this probability is the ***Δ-potential*** of the conflict, $\rho_\Delta(h)$. For the analysis it is also useful to define the unscaled version of this quantity that is the probability that all the vertices of a conflict are present given that each vertex was sampled with probability x_v instead of x_v/Δ. This quantity is the ***potential*** of the conflict, $\rho(h)$. Formally, we have

$$\rho_\Delta(h) = \prod_{v \in h} \frac{x_v}{\Delta} \qquad \text{and} \qquad \rho(h) = \prod_{v \in h} x_v.$$

Another quantity of interest is the expected number of conflicts in which a vertex v participates, given that v is in the sample. This quantity is the ***Δ-resistance*** of a vertex v in a set of vertices $X \subseteq \mathsf{V}$, and it is denoted by

$$\eta_\Delta(v, X) = \frac{\Delta}{x_v} \sum_{h \in \mathcal{H}, h \subseteq X, v \in h} \rho_\Delta(h)\,. \tag{3}$$

The ordering. Note that if the Δ-resistance of v with respect to the set X of vertices that come before it in the ordering is small, the probability of rejecting v is also small. This suggests that the vertex with least resistance (with respect to V) should be the last vertex in the ordering. This suggests the following algorithm for constructing an ordering: Compute the vertex of least resistance and put it last in our ordering (i.e., it is v_n). Next, recursively consider the remaining vertices and compute an ordering for them. In the following, we assume for simplicity that the resulting ordering is $v_1, \ldots, v_n$.

Note that computing the resistance of a vertex by brute force takes $O\big(n^{\overline{\chi}+O(1)}\big)$ time, where $\overline{\chi}$ is the maximum capacity of a hyperedge, and therefore this algorithm is not efficient. We give a polynomial time algorithm for constructing the ordering in Section 3.3.

3.2 Analysis

Our main insight is that if the hypergraph satisfies the bounded growth property defined in Definition 2.2, then for any set $X \subseteq \mathsf{V}$ there exists a vertex $v \in X$ such that $\eta_\Delta(v, X) \leq 1/4$. See Section 3.2.3 below for a proof (see Lemma 3.9).

3.2.1 Quality of approximation

Lemma 3.2. *Let $\mathcal{R}$ and $\mathcal{A}$ be the set of vertices that were selected and accepted by the algorithm, respectively. Furthermore, assume that $\eta_\Delta(v_i, X_i) \le 1/4$, for all i, where $X_i = \langle v_1, \ldots, v_i \rangle$. Then, for $i = 1, \ldots, n$, we have $\mathbf{Pr}[v_i \in \mathcal{A} \mid v_i \in \mathcal{R}] \ge 3/4$.*

Proof: If we selected v_i, we rejected v_i in the alteration phase only if v_i participates in a conflict with some of the vertices in $\{v_1, \ldots, v_{i-1}\} \cap \mathcal{R}$. Let Z_i be the number of conflicts of X_i that contain v_i and are realized in $\mathcal{R}$, i.e., $h \subseteq \mathcal{R}$. In the following, we show that the probability that Z_i is non-zero is at most $1/4$, which implies the lemma.

Consider a k-conflict $h = \{v_{j_1}, \ldots, v_{j_k}, v_i\}$, where each vertex of h is in X_i and h contains v_i. The probability that all of the vertices of h are selected, given that v_i is selected, is equal to $\frac{x_{j_1}}{\Delta} \cdot \frac{x_{j_2}}{\Delta} \cdots \frac{x_{j_k}}{\Delta} = \frac{\Delta}{x_i} \rho_\Delta(h)$. Therefore we have

$$\mathbf{E}\Big[Z_i \;\Big|\; v_i \in \mathcal{R}\Big] = \sum_{h \in \mathcal{H}, h \subseteq X_i, v_i \in h} \frac{\Delta}{x_i} \rho_\Delta(h) = \eta_\Delta(v_i, X_i) \le \frac{1}{4},$$

where the last inequality follows from Lemma 3.9 (shown below) and by assumption (i.e., v_i is the vertex of minimum resistance in X_i). Thus

$$\mathbf{Pr}\Big[v_i \notin \mathcal{A} \;\Big|\; v_i \in \mathcal{R}\Big] \le \mathbf{Pr}\Big[Z_i > 0 \;\Big|\; v_i \in \mathcal{R}\Big] \le \mathbf{E}\Big[Z_i \;\Big|\; v_i \in \mathcal{R}\Big] \le \frac{1}{4}.$$

(The first inequality above uses the argument of Observation 3.1.) Therefore, if v_i is selected, it is accepted with probability at least $3/4$. ■

Corollary 3.3. *Assume that $\eta_\Delta(v_i, X_i) \le 1/4$, for all i. Then the total expected weight of the set of vertices output by the algorithm is $\Omega\big(\mathrm{opt}/\gamma(\mathcal{E})^{1/\chi}\big)$, where opt is the weight of the optimal solution, and χ is the minimum capacity of the given instance.*

Proof: By Lemma 3.2, for each vertex $v \in \mathsf{V}$, we have

$$\mathbf{Pr}\big[v \in \mathcal{A}\big] = \mathbf{Pr}\big[(v \in \mathcal{A}) \cap (v \in \mathcal{R})\big] = \mathbf{Pr}\Big[v \in \mathcal{A} \;\Big|\; v \in \mathcal{R}\Big] \mathbf{Pr}\big[v \in \mathcal{R}\big] \ge \frac{3}{4} \mathbf{Pr}\big[v \in \mathcal{R}\big] = \frac{3x_v}{4\Delta},$$

as $\Delta = O\big(\gamma(\mathcal{E})^{1/\chi}\big)$. By linearity of expectation, the expected weight of the generated solution is then at least

$$\sum_{v \in \mathsf{V}} \frac{3x_v}{4\Delta} w_v = \Omega\left(\frac{\sum_v x_v w_v}{\gamma(\mathcal{E})^{1/\chi}}\right) = \Omega\left(\frac{\mathrm{opt}}{\gamma(\mathcal{E})^{1/\chi}}\right),$$

as $\sum_v x_v w_v$ is the value of the fractional LP solution, which is at least the weight of the optimal solution. ■

3.2.2 On the expected number of realized conflicts

To analyze the algorithm we need to understand how conflicts might form during its execution, and show that the damage of such conflicts to the generated solution is limited. To this end, for a subset $X \subseteq \mathsf{V}$, consider the quantity

$$F_k(t) = \max_{A \subseteq X, |A| \leq t} \left| \left\{ f \;\middle|\; f \in \mathsf{E} \text{ and } |A \cap f| = k+1 \right\} \right|.$$

(The reader may recall that this is the function from Definition 2.2 of bounded growth.) This is the maximum number of k-conflicts that can be realized by a subset of t vertices from X. The quantity of interest in the following is $\sum_{h \in \mathcal{H}_k} \rho(h)$, as it is the expected number of conflicts that would be realized if we sampled according to the LP solution. Our goal is to prove that $\sum_{h \in \mathcal{H}_k} \rho(h)$ is bounded by a function of $F_k(\mathcal{E}(X))$ (i.e., a function of the maximum number of k-conflicts that can be realized by a subset of X whose size is the energy of the LP).

With this goal in mind, we let Υ be a random sample of X such that each vertex $v \in X$ is in Υ independently at random with probability $x_v/2$. We stress that Υ is a random sample that we use for the purposes of defining a quantity $\mathcal{M}$ (i.e., the expected number of conflicts realized in Υ). In the following, we bound $\mathcal{M}$ from above in Lemma 3.4 and from below in Lemma 3.5. Putting these two bounds together imply the desired bound on $\sum_{h \in \mathcal{H}_k} \rho(h)$.

A conflict $h \in \mathcal{H}$ is ***realized*** in Υ if there is a hyperedge $f \in \mathsf{E}$ such that $h = f \cap \Upsilon$ and $|h| = \#(f)+1$.

The following is similar in spirit to the Clarkson technique (a similar but simpler argument was used by Chan and Har-Peled [CH11]).

Lemma 3.4. *The expected number of k-conflicts realized in Υ is $\mathcal{M} = O(F_k(\mathcal{E}(X)))$, where Υ is a random sample of X such that each vertex $v \in X$ is in Υ independently at random with probability $x_v/2$.*

Proof: Each k-conflict h that is realized corresponds to a hyperedge f with capacity k such that $h = f \cap \Upsilon$. Additionally, two realized conflicts that are distinct correspond to different hyperedges. Therefore the number of k-conflicts that are realized in Υ is at most the number of hyperedges f such that the capacity of f is k and $|f \cap \Upsilon| = k+1$. It follows from the definition of $F_k(\cdot)$ that the number of k-conflicts is at most $F_k(|\Upsilon|)$. Therefore it suffices to upper bound the expected value of $F_k(|\Upsilon|)$.

Note that $\mathbf{E}[|\Upsilon|] = \mathcal{E}(X)/2$. We have

$$\begin{aligned} \mathbf{E}\Big[F_k(|\Upsilon|)\Big] &\leq \sum_{t=0}^{\infty} \mathbf{Pr}\left[|\Upsilon| \geq t\frac{\mathcal{E}(X)}{2}\right] F_k\left((t+1)\frac{\mathcal{E}(X)}{2}\right) \leq \sum_{t=0}^{\infty} 2^{-(t+1)/2} F_k\left((t+1)\frac{\mathcal{E}(X)}{2}\right) \\ &\leq \sum_{t=0}^{\infty} 2^{-(t+1)/2} c^{O(\log t)} F_k\big(\mathcal{E}(X)\big) = O\Big(F_k\big(\mathcal{E}(X)\big)\Big), \end{aligned}$$

since G has the bounded growth property (see Definition 2.2), and by the Chernoff inequality (we use here implicitly that $\mathcal{E}(X) \geq 1$). ∎

Lemma 3.5. *For each k-conflict h, the probability that h is realized in Υ is $\geq \rho(h)/2(2e)^k$. Therefore the expected number of k-conflicts realized in Υ is $\mathcal{M} = \Omega\left(\frac{1}{(2e)^k} \sum_{h \in \mathcal{H}_k, h \subseteq X} \rho(h)\right)$.*

Proof: Let $f \in \mathsf{E}$ be a hyperedge with capacity k that generated the conflict h. Since x is a feasible solution for the LP, we have that $\sum_{v\in f\setminus h} x_v \leq \sum_{v\in f} x_v \leq \#(f) = k$. Clearly, the conflict h is realized if we pick all the vertices of h, and none of the vertices of $f \setminus h$, and the probability of that event is

$$\prod_{v\in h} \frac{x_v}{2} \prod_{v\in f\setminus h} \left(1 - \frac{x_v}{2}\right) \geq \frac{1}{2^{k+1}} \prod_{v\in h} x_v \prod_{v\in f\setminus h} \exp(-x_v)$$
$$= \frac{\rho(h)}{2^{k+1}} \cdot \exp\left(-\sum_{v\in f\setminus h} x_v\right) \geq \frac{\rho(h)}{2(2e)^k},$$

where the first line we used the inequality $1 - x_v/2 \geq \exp(-x_v)$, which holds since $x_v \leq 1$. ■

Putting the above two lemmas together, we get the following.

Lemma 3.6. *For any non-negative integer k we have* $\displaystyle\sum_{h\in\mathcal{H}_k, h\subseteq X} \rho(h) = O\Big((2e)^k F_k\big(\mathcal{E}(X)\big)\Big)$.

Running Example 3.7. Continuing the discussion from Running Example 2.3, we have that the expected number of k-conflicts that are being realized by a random sample (sampling more or less according to the LP values) is $\sum_{h\in\mathcal{H}_k} \rho(h) = O\Big((2e)^k k\mathcal{E}\Big)$. This is a hefty quantity, but the key observation is that if we sample according to the LP values scaled down by a large enough constant, then the probability of such a conflict to be realized drops exponentially with k. In particular, for a sufficiently large constant, the expected number of realized k-conflicts in such a sample is going to be $\leq \mathcal{E}/\left(10 \cdot 2^k\right)$. Intuitively, this implies that such conflicts can only cause the algorithm to drop few vertices during the rounding stage, thus guaranteeing a good solution.

3.2.3 Resistance is futile, if you pick the right vertex

In the following, we consider a subset X of the vertices and we show that there exists a vertex $v \in X$ whose Δ-resistance $\eta_\Delta(v, X)$ is at most $1/4$. Recall that $\mathcal{H}_k$ is the set of all k-conflicts involving vertices in V. We can rewrite the Δ-resistance of v in X as

$$\eta_\Delta(v, X) = \frac{\Delta}{x_v} \sum_{h\in\mathcal{H}, h\subseteq X, v\in h} \rho_\Delta(h) = \frac{1}{x_v} \sum_{k=\chi}^{\infty} \frac{1}{\Delta^k} \sum_{h\in\mathcal{H}_k, h\subseteq X, v\in h} \rho(h).$$

The function $F_k(\cdot)$ in Lemma 3.6 above has bounded growth (see Definition 2.2) and we get the following corollary.

Corollary 3.8. *We have* $\sum_{h\in\mathcal{H}_k, h\subseteq X} \rho(h) = O\Big(2^{c'k}\mathcal{E}(X)\,\gamma\big(\mathcal{E}(X)\big)\Big)$, *where c' is a constant.*

Lemma 3.9. *Suppose that the hypergraph G satisfies the bounded growth property (see Definition 2.2). Then, for any set $X \subseteq \mathsf{V}$, there exists a vertex $v \in X$ such that $\eta_\Delta(v, X) \leq 1/4$.*

Proof: Let $T = \sum_{v\in X} x_v \eta_\Delta(v, X)$. The quantity $T/\mathcal{E}(X)$ is the weighted average of the resistances of the vertices in X, where the weight of a vertex v is $x_v/\mathcal{E}(X)$. Therefore it suffices to show that

$T \leq \mathcal{E}(X)/4$, since the minimum resistance is at most the weighted average. Since $\rho_\Delta(h) = \rho(h)/\Delta^{|h|}$, by Eq. $(3)_{\text{p11}}$ we have that

$$T = \sum_{v \in X} x_v \left(\frac{\Delta}{x_v} \sum_{h \in \mathcal{H}, h \subseteq X, v \in h} \rho_\Delta(h) \right) = \sum_{v \in X} \sum_{\substack{h \in \mathcal{H} \\ h \subseteq X \\ v \in h}} \frac{\rho(h)}{\Delta^{|h|-1}} = \sum_{v \in X} \sum_{k=\chi}^{\infty} \sum_{\substack{h \in \mathcal{H}_k \\ h \subseteq X \\ v \in h}} \frac{\rho(h)}{\Delta^{(k+1)-1}}$$

$$= \sum_{k=\chi}^{\infty} \frac{1}{\Delta^k} \sum_{v \in X} \sum_{\substack{h \in \mathcal{H}_k \\ h \subseteq X \\ v \in h}} \rho(h) = \sum_{k=\chi}^{\infty} \frac{k+1}{\Delta^k} \sum_{\substack{h \in \mathcal{H}_k \\ h \subseteq X}} \rho(h) = \sum_{k=\chi}^{\infty} \frac{k+1}{\Delta^k} O\Big(2^{c'k} \mathcal{E}(X) \gamma(\mathcal{E}(X))\Big)$$

$$\leq \mathcal{E}(X) \cdot \underbrace{\beta \sum_{k=\chi}^{\infty} \left(\frac{2^{c'}}{\Delta} \right)^k (k+1) \gamma\big(\mathcal{E}(X)\big)}_{:=S},$$

by Corollary 3.8, where β is some constant. We remind the reader that $\Delta = \alpha\, \gamma(\mathcal{E})^{1/\chi}$, see Eq. $(2)_{\text{p10}}$, and hence we have

$$S = \sum_{k=\chi}^{\infty} \beta \left(\frac{2^{c'}}{\alpha} \right)^k (k+1) \left(\frac{1}{\gamma(\mathcal{E})} \right)^{k/\chi} \gamma(\mathcal{E}(X)) \leq \sum_{k=\chi}^{\infty} \beta \left(\frac{2^{c'}}{\alpha} \right)^k (k+1) \left(\frac{1}{\gamma(\mathcal{E})} \right)^{k/\chi} \gamma(\mathcal{E}) \leq \frac{1}{4}.$$

In the second to last inequality, we have used the fact that $\gamma(\cdot)$ is non-decreasing. The last inequality follows if we pick α to be a sufficiently large constant. Therefore $T \leq \mathcal{E}(X)/4$, and the lemma follows. ■

3.3 Improving the running time

Section 3.1 describes an algorithm that constructs an ordering of the vertices by repeatedly finding the vertex of least resistance with respect to the set of remaining vertices. Computing the resistance of a vertex by brute force takes $O\big(n^{\overline{\chi}+O(1)}\big)$ time, where $\overline{\chi}$ is the maximum capacity of an edge in E. However, for our analysis to go through, we only need to find a vertex that is safe with respect to the set of remaining vertices; informally, a vertex v is safe if the probability that it participates in a conflict with a random sample of the remaining vertices is smaller than some constant (that is strictly smaller than one), where each remaining vertex u is included in the sample with probability x_u/Δ. In this section we show that there is a sampling algorithm that finds a safe vertex with high probability and its running time is polynomial in the maximum capacity $\overline{\chi}$.

Lemma 3.10. *Computing a good ordering of the vertices can be done in polynomial time. Namely, the algorithm of Section 3.1 can be implemented in polynomial time.*

Proof: To get the same quality of approximation we do not need to take the vertex of least resistance in each round (of computing the ordering), but merely a vertex that is "safe". More precisely, let X be the current set of vertices, let v be a vertex of this set, and let Υ be a random sample of X in which each vertex u is included with probability x_u/Δ (also we force v to be in Υ). We say that v is ***violated*** in Υ if v is contained in a hyperedge f such that the number of vertices of f that are in Υ is larger than its capacity $\#(f)$. Let $\mu(v, X)$ denote the probability that v is violated in Υ. Note that $\mu(v, X)$ is a (conservative) upper bound on the probability that v is rejected by our rounding algorithm if we started with an ordering in which $X \setminus \{v\}$ is the set of all vertices that come before v. Therefore, in order for our rounding to succeed, in each round we only need to find a vertex v for which the probability $\mu(v, X)$ is low, where X is the set of all vertices that still need to be ordered at the beginning of the round. (The argument of Lemma 3.9 implies, for any set X, that there is a vertex v for which $\mu(v, X) \leq 1/4$.)

Now we are ready to describe how to construct an ordering for our algorithm. Let X be the set of vertices that still need to be ordered. As we will see shortly, for each vertex $v \in X$, we can compute an estimate $\overline{\mu}(v, X)$ of the probability $\mu(v, X)$. We pick the vertex v with minimum estimated probability $\overline{\mu}(v, X)$, make v the last vertex (in the ordering of X), and recursively order $X \setminus \{v\}$.

We can compute the estimates $\overline{\mu}(v, X)$ in polynomial time as follows. Fix a vertex v. Let $\psi = c_1 \lceil \log n \rceil$, where c_1 is a sufficiently large constant. We pick ψ independent random samples of X (again, forcing v to be in each of these samples); in each random sample, each vertex u is included with probability x_u/Δ. Let Z_v be the number of random samples that rejects v because of a conflict. We set $\overline{\mu}(v, X) = Z_v/\psi$ to be the fraction of the samples in which the v is being rejected. Clearly, $\mathbf{E}[Z_v] = \mu(v, X)\psi$. Our scheme fails if v has high resistance (say larger than 1/3), but our estimate claims that it has low resistance; that is, the bad event is that $\mu(v, X) \geq 1/3$ but $\overline{\mu}(v, X) \leq 1/4$. By the Chernoff inequality,

$$\mathbf{Pr}\left[\overline{\mu}(v, X) \leq \frac{1}{4}\right] = \mathbf{Pr}\left[Z_v \leq \left(1 - \tfrac{1}{4}\right) \mu(v, X)\psi\right] \leq \exp\left(-\frac{\mu(v, X)\psi}{32}\right) \leq \exp\left(-\frac{\psi}{96}\right),$$

which can be made polynomially small by picking c_1 to be sufficiently large.

Thus, our estimates would identify correctly a vertex of low resistance, with high probability, and therefore our rounding algorithm achieves the required approximation with high probability as well. ■

3.4 The result

Putting the above together (specifically, Corollary 3.3, Lemma 3.9, and Lemma 3.10) we get the following.

Theorem 3.11. *Let $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ be a hypergraph with a weight function $w(\cdot)$ on the vertices and a capacity function $\#(\cdot)$ on the edges, such that $|\mathsf{E}|$ is polynomial in $|\mathsf{V}|$ and G satisfies the bounded growth property (see Definition 2.2). Then we can compute in polynomial time a subset $X \subseteq \mathsf{V}$ of vertices such that no hyperedge f contains more than its capacity $\#(f)$ vertices of X, and such that in expectation, the total weight of the output set is $\Omega\left(\mathrm{opt}/\gamma(\mathcal{E})^{1/\chi}\right)$, where opt is the weight of the optimal solution, $\mathcal{E}$ is the energy of the LP, χ is the minimum capacity of the given instance, and $\gamma(\cdot)$ is the growth function of Definition 2.2.*

Consider an integer constant $\phi > 0$, and observe that one can always relax the capacity constraints of a given instance of HGraphPacking by replacing all capacities smaller than ϕ by this value. Theorem 3.11 thus implies the following.

Corollary 3.12. *Given an instance of HGraphPacking, with the bounded growth property, one can compute in polynomial time a $\left(O\left(\gamma(\mathcal{E})^{1/\phi}\right), \phi\right)$-approximation to the optimal solution.*

3.5 Contention resolution schemes

It turns out that our rounding scheme falls into a more general technique known as contention resolution schemes. These schemes work by first selecting the elements according to some fractional solution, and then filtering out a valid subset from the selected set. Informally, a contention resolution scheme requires that a selected element has constant probability to survive the second stage purge. Given such a rounding scheme, one can use them to solve more general optimization problems.

Definition 3.13. *Let $\mathsf{V} = \{v_1, \ldots, v_n\}$ be a finite ground set of size n. A function $g : 2^{\mathsf{V}} \to \mathbb{R}$ is* **submodular** *if $g(X) + g(Y) \geq g(X \cap Y) + g(X \cup Y)$ for any two subsets X, Y of V.*

A family $\mathcal{I} \subseteq 2^{\mathsf{V}}$ of subsets of V is **downward-closed** *if $Y \in \mathcal{I}$ and $X \subseteq Y$ then $X \in \mathcal{I}$. For a subset $X \subseteq \mathsf{V}$, its* **characteristic** *vector $\chi(X) \in \{0,1\}^n$ has 1 in the ith coordinate if $v_i \in X$, and 0 otherwise. The* **hull** *of $\mathcal{I}$, denoted by $P_{\mathcal{I}}$, is the convex hull of all the characteristic vectors in $\mathcal{I}$.*

For example, the problem of maximizing $g(X)$ subject to the constraint that $X \in \mathcal{I}$ generalizes the hypergraph packing problem if $g(X) = \sum_{v \in X} w_v$, and the family $\mathcal{I}$ is the family of all subsets $X \subseteq \mathsf{V}$ such that, for each hyperedge f, $|f \cap X| \leq \#(f)$. In this case, the set $P_{\mathcal{I}}$ is the convex hull of the characteristic vectors of these subsets. In particular, the feasible region of Hypergraph-LP is superset of $P_{\mathcal{I}}$[6], which contains no integral point which is not also in $P_{\mathcal{I}}$, and as such it is sufficient for our purposes use the feasible region of Hypergraph-LP as our $P_{\mathcal{I}}$. This substitution is needed since we do not have efficient access to $P_{\mathcal{I}}$ but we do have efficient access to Hypergraph-LP.

Given $x \in P_{\mathcal{I}} \subseteq [0,1]^n$ and a constant $b \in (0,1]$, we construct a random set $\mathcal{R}(bx)$ by picking each $v_i \in \mathsf{V}$ independently at random with probability bx_i. It is unlikely that this set is in $\mathcal{I}$, and we apply some cleanup algorithm on $\mathcal{R}(bx)$ to get a subset $\mathcal{A} \subseteq \mathcal{R}(bx)$ such that $\mathcal{A} \in \mathcal{I}$.

Definition 3.14 ([CVZ11]). *A* **(b,c)-balanced contention resolution scheme** *for $P_{\mathcal{I}}$ is a scheme where for any $x \in P_{\mathcal{I}}$, the scheme selects a subset $\mathcal{A} \subseteq \mathcal{R}(bx)$ such that (i) $\mathcal{A} \in \mathcal{I}$, and (ii) $\mathbf{Pr}\Big[v_i \in \mathcal{A} \,\Big|\, v_i \in \mathcal{R}(bx)$ c, for every $v_i \in \mathsf{V}$.*

The scheme is **monotone** *if $\mathbf{Pr}\Big[v_i \in \mathcal{A} \,\Big|\, \mathcal{R}(bx) = R_1\Big] \geq \mathbf{Pr}\Big[v_i \in \mathcal{A} \,\Big|\, \mathcal{R}(bx) = R_2\Big]$ for any $v_i \in R_1 \subseteq R_2$.*

It is easy to verify that the rounding scheme described in Section 3.1 is a monotone $(\Delta, 3/4)$-balanced CR scheme on $x \in P_{\mathcal{I}}$, where $P_{\mathcal{I}}$ is the set of all feasible solutions to the Hypergraph-LP relaxation.

In the following, we require that $P_{\mathcal{I}}$ is (i) a polytope, (ii) down-monotone (i.e., if $x \in P_{\mathcal{I}}$ then the segment $x0$ is contained in $P_{\mathcal{I}}$), and (iii) solvable – the maximum of any linear function defined over $P_{\mathcal{I}}$ can be computed in polynomial time.[7]

Theorem 3.15 ([CVZ11]). *Let $g : 2^{\mathsf{V}} \to \mathbb{R}^+$ be a non-negative submodular function. Assume that we are given a polynomial time monotone (b,c)-balanced CR scheme defined over $P_{\mathcal{I}}$. Then one can compute, in randomized polynomial time, a set $J \in \mathcal{I}$, such that $\mathbf{E}[g(J)] \geq \alpha g(\mathrm{opt})$, where opt is the optimal solution; that is $\mathrm{opt} = \arg\max_{Y \in \mathcal{I}} g(Y)$. Here $\alpha = bc/c_1$, where $c_1 > 0$ is some absolute constant.*

Corollary 3.16. *Let $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ be a hypergraph. Let $w : 2^{\mathsf{V}} \to \mathbb{R}^+$ be a weight function on the vertices that is non-negative and submodular. Let $\#(\cdot)$ be a capacity function on the edges such that $|\mathsf{E}|$ is polynomial in $|\mathsf{V}|$ and G satisfies the bounded growth property (see Definition 2.2). Then we can compute in polynomial time a subset $X \subseteq \mathsf{V}$ of vertices such that no hyperedge f contains more than its capacity $\#(f)$ vertices of X. Furthermore, in expectation, the total weight of the output set is $\Omega\big(\mathrm{opt}/\gamma(\mathcal{E})^{1/\chi}\big)$, where opt is the weight of the optimal solution, and χ is the minimum capacity of the given instance.*

3.5.1 Applications

We sketch a few general applications here – they are discussed in more detail (for more some specific cases) in Section 4 below.

[6] They are not equal since the feasible region of Hypergraph-LP may contain non-integral vertices.

[7] Here, and in the rest of the paper, we assume that solving an LP takes polynomial time. This is reasonable as the LPs being considered involve "small" integer numbers.

Set cover with bounded depth. Consider the standard geometric set cover problem, where we have a set P of points in the plane, and a set $\mathcal{D}$ of regions, say disks. We would like to cover as many points as possible, while no point $\mathsf{p} \in \mathsf{P}$ is covered more than its capacity. An edge in E is the subset of disks around a given point. The hypergraph here is $\mathsf{G} = (\mathcal{D}, \mathsf{E})$, and the payoff function is $g(f) = \left|\left(\cup_{\mathsf{d} \in f} \mathsf{d}\right) \cap \mathsf{P}\right|$. This function is submodular – indeed, consider two subsets $X, Y \subseteq \mathcal{D}$, and observe that for any point $\mathsf{p} \in \mathsf{P} \cap (\bigcup X \cup \bigcup Y)$, it is either covered only by one of these subsets of disks, or alternatively, it is covered by both. In any case, the inequality $g(X) + g(Y) \geq g(X \cap Y) + g(X \cup Y)$ holds[8]. This falls into the framework of Corollary 3.16, and we can get a good approximation in this case.

Partial coverage with bounded depth. We can extend the above. For example, if a point is not covered, we can still allow it to get some partial coverage if it is close enough to the boundary of the covered region. So, for a point $\mathsf{p} \in \mathsf{P}$, let $\psi_{\mathsf{p}}(r)$ be a monotone decreasing function. For a set of disks $X \subseteq \mathcal{D}$, we define the payoff for a point $\mathsf{p} \in \mathsf{P}$, to be $\psi_{\mathsf{p}}(r)$ where r is the distance of p to the union of disks in X. As done above, it is enough to verify that the submodularity condition holds for a single point, which indeed holds. Again, Corollary 3.16 applies, and we can get a good approximation in this case. Note, that this is an extension of the previous problem. See Corollary 4.4 for the exact result.

4 Applications

Using our main result (Theorem 3.11), we get several approximation algorithms for the packing problems mentioned in the introduction. We present some of these results here.

4.1 Packing regions with low union complexity

Let $\mathcal{D}$ be a set of n weighted regions in the plane, and let the maximum union complexity of $m \leq n$ objects of $\mathcal{D}$ be $\mathsf{U}(m) = m\mathsf{u}(m)$. We assume that (i) $\mathsf{U}(n)/n = \mathsf{u}(n)$ is a non-decreasing function, and (ii) there exists a constant c, such that $\mathsf{U}(xr) \leq c\,\mathsf{U}(r)$, for any r and $1 \leq x \leq 2$. We are also given a set of points P, where each point $\mathsf{p} \in \mathsf{P}$ is assigned a positive integer $\#(\mathsf{p})$ which is the capacity of p.

We are interested in solving PackRegions (Problem 1.1) for $\mathcal{D}$ and P. Consider the hypergraph G obtained by creating a vertex for each region and a hyperedge for each subset of regions containing a given point of P. Here, $F_k(t)$ is bounded by the number of faces in the arrangement of t regions of depth exactly $k+1$. The number of such faces can be bounded by the union complexity by a standard application of the Clarkson technique [Cla88, CS89].

Lemma 4.1. *Consider a set of regions in the plane such that the boundary of every pair intersects a constant number of times. The number of faces of depth at most $k+1$ in the arrangement of any subset of these regions of size t is $O(k^2\mathsf{U}(t/k))$.*

Plugging this bound into Theorem 3.11 yields the following result.

Theorem 4.2. *Let $\mathcal{D}$ be a set of m weighted regions in the plane such that the union complexity of any t of them is $\mathsf{U}(t) = t\mathsf{u}(t)$. Let P be a set of n points in the plane, where there is a capacity $\#(\mathsf{p})$ associated with each point $\mathsf{p} \in \mathsf{P}$. There is a polynomial time algorithm that computes a subset $\mathcal{A} \subseteq \mathcal{D}$ of regions*

[8] Somewhat confusingly, this inequality might be strict – indeed, imagine a point of P covered by exactly two different disks, one belonging to X and the other belonging to Y. Clearly, this point would contribute two on the left side, but only one on the right side (i.e., the point is in the geometric intersection of these disks, but not in the intersection of the sets).

such that no point $\mathsf{p} \in \mathsf{P}$ is contained in more than $\#(\mathsf{p})$ regions of $\mathcal{A}$. Furthermore, in expectation, the total weight of the output set is $\Omega\Big(\mathrm{opt}/\mathsf{u}(\mathcal{E})^{1/\chi}\Big)$, where opt is the weight of the optimal solution, $\mathcal{E}$ is the energy of the LP solution and χ is the minimum capacity of the given instance.

Alternatively, for any integer constant ϕ, one can get an $\Big(O\Big(\mathsf{u}(\mathcal{E})^{1/\phi}\Big), \phi\Big)$-approximation to the optimal solution for the given instance.

The following results follow from the theorem above.

Corollary 4.3. *We get approximation algorithms for* PackRegions *with the following guarantees:*
(A) $O(1)$-approximation for fat triangles of similar size, disks, or pseudo-disks.
(B) An $O(\log^* n)$-approximation for (arbitrary) fat triangles.
(C) For any integer constant $\phi > 0$, an $\big(O\big(\mathcal{E}^{1/\phi}\big), \phi\big)$-approximation, for a set of regions in the plane such that the boundaries of any pair of them intersects a constant number of times.

Proof: (A) The union complexity of pseudo-disks and fat triangles of similar size is linear; that is, $\mathsf{U}(t) = O(t)$. Therefore we get an $O(1)$-approximation for PackRegions if the regions are fat triangles of similar size, disks, or pseudo-disks.

(B) The union complexity of fat triangles is $O(n \log^* n)$ [AdBES14].

(C) In this case, $\mathsf{U}(t) = O(t^2)$ and $\mathsf{u}(t) = O(t)$. Therefore, for any integer constant $\phi > 0$, we get an $\big(O\big(\mathcal{E}^{1/\phi}\big), \phi\big)$-approximation for instances of PackRegions on such regions. ■

Plugging the above into the problems of Section 3.5.1, we get the following.

Corollary 4.4. *Let P be a set of n points in the plane with capacities, and let $\mathcal{D}$ be a set of objects with linear union complexity (e.g., disks, pseudo-disks, and fat triangles of similar size). Furthermore, assume that for each point $\mathsf{p} \in \mathsf{P}$ there is a monotone decreasing coverage function $\psi_{\mathsf{p}}(r)$, which returns the amount of coverage p gets if its distance from the selected set of regions is r (if p is covered than $r = 0$). Specifically, given a set of disks $X \subseteq \mathcal{D}$, the coverage X provides is $g(X) = \sum_{\mathsf{p}\in\mathsf{P}} \psi_{\mathsf{p}}\Big(\mathsf{d}_{\mathsf{p}}\big(\bigcup_{\mathsf{d}\in X} \mathsf{d}\big)\Big)$, where $\mathsf{d}_{\mathsf{p}}(S)$ denotes the distance of p from the set $S \subseteq \mathbb{R}^2$.*

Then, one can compute, in polynomial time, $O(1)$ approximation to the subset of disks of $\mathcal{D}$ that maximizes coverage, while not violating the capacity constraints.

4.2 Packing halfspaces, rays and disks

Problem 4.5. (A) **PackHalfspaces***: Given a weighted set of halfspaces $\mathcal{S}$ and a set of points P with capacities in $\mathbb{R}^3$, find a maximum weight subset $\mathcal{A}$ of $\mathcal{S}$ so that, for each point p, the number of halfspaces of $\mathcal{A}$ that contains p is at most $\#(\mathsf{p})$.*

(B) **PackRaysInPlanes***: Given a weighted set of vertical rays $\mathcal{R}$ and a set of planes $\mathcal{H}$ with capacities in $\mathbb{R}^3$, find a maximum weight subset $\mathcal{A}$ of $\mathcal{R}$ so that, for each plane h, the number of rays of $\mathcal{A}$ that intersect h is at most $\#(\mathsf{h})$.*

(C) **PackPointsInDisks***: Given a set $\mathcal{D}$ of disks with capacities and a weighted set P of points, find a maximum weight subset $\mathcal{A}$ of the points so that each disk $\mathsf{d} \in \mathcal{D}$ contains at most $\#(\mathsf{d})$ points of $\mathcal{A}$.*

Since the union complexity of halfspaces in three dimensions is linear, we get the following from Theorem 3.11 (and the 3d analogue of Lemma 4.1).

Corollary 4.6. *One can compute, in polynomial time, a constant factor approximation to the optimal solution of the* PackHalfspaces *problem.*

Standard point/plane duality implies that the same result holds for the dual problem. Namely, a point (a, b, c) gets mapped to the plane $z = ax + by - c$, and a plane $z = ax + by + c$ gets mapped to the point $(a, b, -c)$. Also, a point lies below a given plane if and only if the dual point of the plane lies below the dual plane of the point. As such, the dual of an instance of PackHalfspaces is an instance of PackRaysInPlanes (and vice versa).

Thus, Corollary 4.6 implies the following.

Corollary 4.7. *One can compute, in polynomial time, a constant factor approximation to the optimal solution of the* PackRaysInPlanes *problem.*

Finally, observe that an instance of PackPointsInDisks can be lifted into an instance of PackRaysInPlanes, by the standard lifting $f(x, y) = (x, y, x^2 + y^2)$, which maps points and disks in the plane to halfspaces and points in three dimensions [dBCvKO08].

Corollary 4.8. *One can compute, in polynomial time, a constant factor approximation to the optimal solution to the* PackPointsInDisks *problem.*

4.3 Axis Parallel Rectangles/Boxes

4.3.1 Packing rectangles (2d)

Problem 4.9. (PackRectsInPoints) *Given a weighted set* $\mathcal{B}$ *of axis-parallel rectangles in the plane, and a point set* P *with capacities, find a maximum weight subset* $\mathcal{A} \subseteq \mathcal{B}$*, such that, for any* $\mathsf{p} \in \mathsf{P}$*, the number of rectangles of* $\mathcal{A}$ *containing* p *is at most* $\#(\mathsf{p})$*.*

Note that the union complexity of a set of rectangles can be quadratic. Hence we cannot simply use Theorem 4.2 to get a meaningful approximation. However, by using the standard approach for approximating the independent set of rectangles, one can get a reasonable approximation as the following lemma demonstrates.

Lemma 4.10. *Given an instance* $(\mathcal{B}, \mathsf{P})$ *of* PackRectsInPoints *with* m *rectangles, one can compute, in polynomial time, a subset* $\mathcal{A} \subseteq \mathcal{B}$ *of total weight* $\Omega(\mathrm{opt}/\log m)$ *such that no capacity constraint of* P *is violated, where* opt *is the weight of the optimal solution.*

Proof: It is straightforward to verify that a set of rectangles that intersect a common line have linear union complexity. Therefore it follows from Theorem 4.2 that we can get a constant factor approximation for such instances. Given an arbitrary set of axis-parallel rectangles, we can reduce it to the case in which all rectangles intersect a common line as follows.

We next modify an idea of Agarwal *et al.* [AKS98]. We construct an interval tree on $\mathcal{B}$. Let $\mathsf{b} \in \mathcal{B}$ be the median rectangle of $\mathcal{B}$ when sorted by left edges. Let ℓ denote the vertical line which passes through the left edge of b, and let $\mathcal{B}_\ell$ denote the set of rectangles it intersects. We associate $\mathcal{B}_\ell$ with the root of our tree, and then recursively build left and right subtrees for the rectangles in $\mathcal{B} \setminus \mathcal{B}_\ell$ that lie to the left or right of ℓ, respectively. The recursion bottoms out once every rectangle has been stabbed by a line. Clearly the depth of the tree is $O(\log m)$, since each time we choose the median line and only recursively continue on those rectangles that do not intersect it. Therefore there exists a level of the range tree that has a solution of weight $\Omega(\mathrm{opt}/\log m)$. The algorithm now considers each level of the tree separately, and for each node at the given level, it constructs an approximate solution for the rectangles associated with the node using the constant factor approximation algorithm guaranteed by Theorem 4.2. Next, the algorithm considers the union of all these solutions to form the solution for this level. Since two rectangles associated with two different nodes at the same depth in the range tree do not intersect, the resulting set is a valid solution.

The algorithm returns the best solution found among all the levels. Clearly, its weight is at least $\Omega(\mathrm{opt}/\log m)$. ■

We leave the problem of improving the approximation to $O(\log \mathrm{opt})$ in Lemma 4.10 as an open problem for further research.

4.3.2 Packing axis-parallel boxes (3d)

The union complexity of axis-parallel boxes in $\mathbb{R}^3$ that contain a common point is also linear, and therefore a similar approach as above will enable us to solve the following problem.

Problem 4.11. (**PackBoxesInPoints**) *Given a weighted set $\mathcal{B}$ of axis-parallel boxes in $\mathbb{R}^3$, and a point set P with capacities, find a maximum weight subset $\mathcal{A} \subseteq \mathcal{B}$, such that, for any $\mathsf{p} \in \mathsf{P}$, the number of boxes of $\mathcal{A}$ containing p is at most $\#(\mathsf{p})$.*

Lemma 4.12. *Given an instance $(\mathcal{B}, \mathsf{P})$ of PackBoxesInPoints with m boxes, one can compute, in polynomial time, a subset $\mathcal{A} \subseteq \mathcal{B}$ of total weight $\Omega(\mathrm{opt}/\log^3 m)$ such that no capacity constraint of P is violated, where opt is the weight of the optimal solution.*

Proof: We build a multi-layer interval tree on the boxes. On the top layer, we build a balanced tree on the x-axis projection of the boxes, where a node $v_{x'}$ stores all boxes intersecting the plane $x = x'$. Next, we build for each such node a secondary interval tree on the y-axis projections, and for each node on this secondary data structure we build a third layer data structure on the z-axis projections. All the boxes are stored in the nodes of the third layer data structure.

First, observe that the boxes stored in a node on the third layer, $v_{x'y'z'}$, all contain the point (x', y', z'). Now, the union complexity of axis parallel boxes all sharing a common point is linear. As such, we can apply Theorem 3.11 to compute a packing that does not violate the capacities and is a constant factor approximation to the optimal (on this restricted set of boxes). Now, for each third layer tree, we find the level that contains the best possible combined solution (taking the union of the solutions at a level is valid since there is no point in common to two boxes that are stored in two different nodes at the same level). This assigns values for each node in the secondary tree. Again, we choose for each secondary tree the level with the maximum total weight solution. We assign this value to the corresponding node in the first layer data structure. Again, we choose the level with the highest possible value. This corresponds to a valid solution that complies with the capacity constraints.

As for the quality of approximation, observe that for a tree at a given layer, at least a logarithmic factor of the remaining weight of the optimal solution is contained in some level of the tree. As such, each time we go down one layer in this data structure we lose at most a logarithmic factor of the optimal solution, and hence the quality of approximation of this algorithm is $\Omega(\log^3 m)$. ■

Remark 4.13. (A) It is natural to ask if this result can be extended to higher dimensions. However, it is easy to see that the union complexity of n axis-parallel boxes in four dimensions that all contain a common point can be quadratic. As such, this approach would fail miserably. We leave the problem of getting a better approximation for this case as an open problem for further research.

(B) The continuous version seems to be considerably easier. For the weighted case (with unit capacities; that is, the independent set variant) an $O(\log^{d-1} m/\log\log m)$ approximation is known [CH09] by solving the two dimensional case, and then using the above interval tree technique to apply it for higher dimensions. For the unweighted continuous case, an $O(\log^{d-2} m \ \log\log m)$ approximation is known [CC09].

The discrete version is different than the continuous version because, for example, if considering boxes in $\mathbb{R}^3$ that all intersect the xy-plane, the induced two dimensional instance fails to encode the capacity constraints, as they rises from points in three dimensions that do not lie on this plane (while in the continuous case, it is enough to solve the induced problem in this plane).

4.3.3 Packing points into rectangles

Problem 4.14. (**PackPntsInRects**) *Given a weighted set* P *of points and a set* $\mathcal{B}$ *of axis-parallel rectangles with capacities in the plane, find a maximum weight subset* $\mathcal{A} \subseteq \mathsf{P}$*, such that, for any* $\mathsf{b} \in \mathcal{B}$*, the number of points of* $\mathcal{A}$ *contained in* b *is at most* $\#(\mathsf{b})$*.*

We first observe that the hypergraph that arises from an instance $\mathsf{G} = (\mathsf{P}, \mathcal{B})$ of PackPntsInRects might not have the bounded growth property for any reasonable growth function. To see this consider the following example.

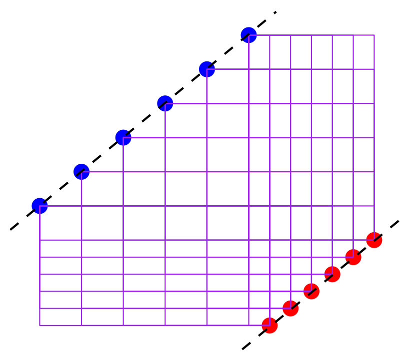

Figure 3

Consider two parallel lines in the plane with positive slope. Place $n/2$ points on each line such that all the points on the top line lie above and to the left of all the points on the bottom line. Let the set of rectangles for this instance of PackPntsInRects be all the rectangles which have a point on the top line as their upper left corner and a point on the bottom line as their lower right corner. In this case any subset of $O(t)$ points from the top line and $O(t)$ points from the bottom line induce a set of $O(t^2)$ hyperedges, each of size 2. Therefore, $F_1(t) = \Omega(t^2)$, and hence Theorem 3.11 only gives an $O(\mathcal{E})$ approximation.

Since we cannot hope to apply our main result to the case of PackPntsInRects, we will instead seek a bi-criteria approximation. Our algorithm here is inspired by the work of Ezra *et al.* [AES10] on ε-nets for rectangles. Before tackling this problem, we will first consider an easier variant, which will be useful later in obtaining a bi-criteria approximation. In the following, we call a set of rectangles such that all their (say) bottom edges lies on a common line a ***baseline***.

Problem 4.15. (**PackPntsInBaseline**) *Given a weighted set* P *of points and a set* $\mathcal{B}$ *of baseline rectangles with capacities in the plane, find a maximum weight subset* $\mathcal{A} \subseteq \mathsf{P}$*, such that, for any* $\mathsf{b} \in \mathcal{B}$*, the number of points of* $\mathcal{A}$ *contained in* b *is at most* $\#(\mathsf{b})$*.*

Lemma 4.16. *Let* P *be a set of* n *points in the plane all placed above the* x*-axis. Let* $F_k(n)$ *be the maximum number of different subsets of* P *of size* k *that are realized by intersecting* P *with a rectangle whose bottom edge lies on the* x*-axis. We have that* $F_k(n) = O(nk^2)$*.*

Proof: Consider a rectangle b with its bottom edge lying on the x-axis, and which contains k points of P. Lower its top edge till it passes through a point of P, and let p denote this point. Similarly, move its left and right edges till they pass through points of P. Let b' be this new canonical rectangle. Now, let i_{left} (resp. i_{right}) be the number of points of P inside b' that are to the left (resp. right) of p. Clearly, $(\mathsf{p}, i_{\text{left}}, i_{\text{right}})$ uniquely identifies this canonical rectangle. This implies the claim as $\mathsf{p} \in \mathsf{P}$, $i_{\text{left}} \leq k$ and $i_{\text{right}} \leq k$, and hence the numbers of such triples is $O(|\mathsf{P}|\, k^2)$. ■

Lemma 4.17. *Given an instance of* PackPntsInBaseline*, one can compute, in polynomial time, an $O(1)$-approximation to the optimal solution.*

Proof: Consider the associated hypergraph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$. By Lemma 4.16, this hypergraph has the bounded growth property with $F_k(t) = O(tk^2)$ (here $\gamma(t) = 1$). Therefore, the algorithm of Theorem 3.11 provides the required approximation. ■

Lemma 4.18. *Given a set P of n points in the plane, and a parameter k, one can compute a set $\mathcal{D}$ of $O(k^2 n \log n)$ axis-parallel rectangles, such that for any axis-parallel rectangle b, if $|\mathsf{b} \cap \mathsf{P}| \leq k$, then there exists two rectangles $\mathsf{b}_1, \mathsf{b}_2 \in \mathcal{D}$ such that $(\mathsf{b}_1 \cup \mathsf{b}_2) \cap \mathsf{P} = \mathsf{b} \cap \mathsf{P}$.*

Furthermore, consider the graph where two points of P are connected if they belong to the same rectangle in $\mathcal{D}$. Then the number of edges in this graph is $O(nk \log n)$.

Proof: Find a horizontal line ℓ that splits P equally, and compute all the baseline rectangles that contain at most k points of P (that is, compute both the rectangle above and below the line). By Lemma 4.16, the number of such rectangles is $O(nk^2)$. Now, recursively compute the rectangle set for the points above ℓ, and for the points below ℓ. Clearly, the number of rectangles generated is $O(k^2 n \log n)$, and let $\mathcal{D}$ denote the resulting set of rectangles.

Now, consider any axis-parallel rectangle b such that $|\mathsf{b} \cap \mathsf{P}| \leq k$. If it is does not intersect ℓ then by induction it has the desired property. Otherwise, if b intersects ℓ, then it can be decomposed into two baseline rectangles, each one of them contains at most k points of P. By construction, for each of these rectangles there is a rectangle in $\mathcal{D}$ that contains exactly the same set of points.

As for the second claim, we apply a similar argument. Consider an edge pq in this graph that arise because of a top baseline rectangle of ℓ. Furthermore, assume that p is higher than q and to its right. Clearly, there are at most k such edges emanating from p, as the baseline rectangle having p as its top right corner and having its left edge through q contains at most k points, and each such rectangle corresponds to a unique edge. As such, we get that the number of edges in the graph is $E(n) = O(nk) + 2E(n/2) = O(nk \log n)$. ■

Remark 4.19. A slightly more careful analysis shows that the number of rectangles in the set computed by Lemma 4.18 that contain exactly k points is $O(nk \log n)$. This will not be needed for our analysis.

Remark 4.20. Consider an instance $\mathsf{G} = (\mathsf{P}, \mathcal{B})$ of PackPntsInRects. Let $\mathsf{G}' = (\mathsf{P}, \mathcal{D})$ be a modified instance of PackPntsInRects where $\mathcal{D}$ is obtained from $\mathcal{B}$ by replacing each rectangle by two new rectangles whose union covers that same set of points. Lemma 4.18 guarantees that this can be done such that $|\mathcal{D}| = O(n^3 \log n)$. One might be tempted to believe that we can plug G' into Theorem 3.11 in order to get a bi-criteria approximation for G. Unfortunately, this does not work since (as the following example shows) the hypergraph does not have the bounded growth property for any meaningful growth function.

Consider two parallel lines in the plane with positive slope. Place $\Theta(\log n)$ points of P on each line (or close to the line) such that the points on (or close to) the top line all lie above and to the left of

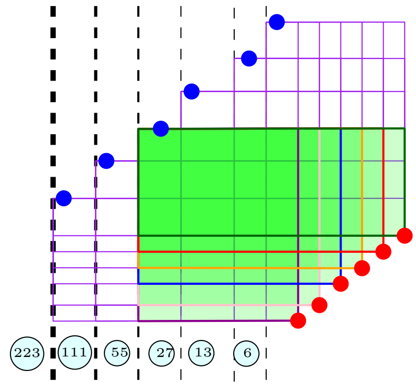

Figure 4: No bounded growth.

those on the bottom line. The remaining points of P will all lie below the points on the diagonals. Let T be the interval tree of P (using vertical split lines). Specifically, the remaining points of P will be placed such that each point on the top line lies in a different level of T, and all the points on the bottom line lie in the same node as the rightmost point on the top line. More specifically, the leftmost point on the top line will correspond to the root and the points in order from the left to right on the top line will correspond to continually walking down in the tree. This is depicted in Figure 4, where the notation ⓤ represents a cluster of u points close together. Now let X be the subset of P which consists of the two set of $\Theta(\log n)$ points on the diagonal lines. Consider the intersection sub-hypergraph induced by X. Suppose that $\mathcal{B}$ has rectangle for every pair of points in P that can be obtained as the intersection of a rectangle with P. Then any pair of points from the top and bottom diagonals will correspond to a hyperedge in this induced sub-hypergraph. Therefore, $F_1(\log n) = \Omega(\log^2 n)$, and hence Theorem 3.11 only gives an $O(\mathcal{E})$ approximation.

Since (as the above remark demonstrates) we cannot directly apply Lemma 4.18, our approach will be more roundabout. We first show how to solve the independent set variant of our problem (i.e., unit capacities). Next, we slice the rectangles of the given instance with non-uniform capacities case into subrectangles with unit capacities, and plug it into the above algorithm to get a meaningful approximation.

Lemma 4.21. *Given an instance of $\mathsf{G} = (\mathsf{V}, \mathcal{B})$ of* PackPntsInRects *with unit capacities, one can compute a subset $X \subseteq \mathsf{V}$, such that the total weight of X is $\Omega(\mathrm{opt}/\log \mathcal{E})$ and each rectangle of $\mathcal{B}$ contains at most 2 points of P, where $n = |\mathsf{P}|$.*

Proof: We first use Lemma 2.5 to sparsify the given instance. We now have a set of $\mathsf{P} \subseteq \mathsf{V}$ of $t = \Theta(\mathcal{E} \log \mathcal{E})$ points, and an associated fractional solution, such that none of the constraints are violated. The value of the fractional solution on G_P is $\Omega(\mathrm{opt})$, and as such we restrict our search for a solution to P.

Furthermore, we can assume that the value assigned to each point of P by this fractional solution is exactly $1/M$ (we replicate a point i times if it is assigned value i/M), where $M = O(\log \mathcal{E})$. Note, that none of the rectangles of $\mathcal{B}$ contains more than M points of P. In particular, by Lemma 4.18, one can build a set of rectangles $\mathcal{D}$ of size $O(M^2 t \log t)$, such that every rectangle of $\mathcal{B}$ can be covered by the union of two rectangles of $\mathcal{D}$; formally, for every $\mathsf{b} \in \mathcal{B}$ there exists $\mathsf{b}_1, \mathsf{b}_2 \in \mathcal{D}$ such that $\mathsf{b} \cap \mathsf{P} = (\mathsf{b}_1 \cup \mathsf{b}_2) \cap \mathsf{P}$. We build a conflict graph G over P connecting two points if (i) they are both contained in a rectangle of $\mathcal{B}$, and (ii) there is a rectangle of $\mathcal{D}$ that contains them both. By Lemma 4.18 this graph has at most $O(Mt \log t) = O(\mathcal{E} \log^3 \mathcal{E})$ edges and $t = \Theta(\mathcal{E} \log \mathcal{E})$ vertices.

We further add edges to G making a clique out of each group of duplicated points that arose from a single given point of P (this is needed since when duplicating the points we perturbed them in order to maintain the implicit general position assumptions of Lemma 4.18, and one needs to guarantee that at most one of these copies is picked to the independent set). Now for a point $\mathsf{p} \in \mathsf{P}$ with LP value x_{p}, the number of duplicated points is $x_{\mathsf{p}} M$. Hence the number of edges added for these cliques is

$$\sum_{\mathsf{p} \in \mathsf{P}} \binom{x_{\mathsf{p}} M}{2} \leq \sum_{\mathsf{p} \in \mathsf{P}} x_{\mathsf{p}}^2 M^2 \leq M^2 \sum_{\mathsf{p} \in \mathsf{P}} x_{\mathsf{p}} \leq M^2 \mathcal{E} = O\left(\mathcal{E} \log^2 \mathcal{E}\right),$$

and hence the number of edges in G overall is $O(\mathcal{E} \log^3 \mathcal{E})$.

It is easy to verify that G has average degree $O(\log^2 \mathcal{E})$, and the total weight of the vertices is $\Theta(\mathrm{opt} \log \mathcal{E})$, as such, by Turán's theorem, one can compute an independent set of vertices in this graph of weight $\Omega(w(\mathsf{P})/(\text{average degree} + 1)) = \Omega(\mathrm{opt}/\log \mathcal{E})$.

Now, it is easy to verify that any rectangle in $\mathcal{B}$ contains at most two points of this independent set. ∎

Theorem 4.22. *Given an instance of $(\mathsf{V}, \mathcal{B})$ of* PackPntsInRects *(with arbitrary capacities), one can compute in polynomial time a subset $X \subseteq \mathsf{V}$ that is an $(O(\log \mathcal{E}), 2)$-approximation to the optimal solution.*

Proof: Compute a fractional solution to the given instance. Split each rectangle b with capacity $\#(\mathsf{b})$ into $\lceil \#(\mathsf{b})/3 \rceil$ rectangles, each one containing at most value 4 from the fractional solution (this can be done by sweeping the rectangle from left to right, and splitting it whenever the fractional solution inside the current portion exceeds 3). Consider now a unit capacity instance on the same point set but with these new rectangles. We use Lemma 4.21 in order to get an $(O(\log \mathcal{E}), 2)$-approximation for this new instance.

We now show that this solution we obtained for the unit capacity instance is also an $(O(\log \mathcal{E}), 2)$-approximation to the original instance. First observe that the LP value on this new instance is $\Omega(\mathrm{opt})$ (where opt is the LP value of the original instance) since scaling down the fractional solution to original instance by a factor of 4 would be a valid solution to the LP for the new instance (since these newly created rectangles each contained at most 4 from the fractional solution), and hence the weight of the approximation is $\Omega(\mathrm{opt}/\log \mathcal{E})$. Furthermore, we know every rectangle $\mathsf{b} \in \mathcal{B}$ contains at most $2 \lceil \#(\mathsf{b})/3 \rceil \leq \max(2, \#(\mathsf{b}))$ points from this solution, since in the new instance each rectangle from $\mathcal{B}$ was replaced with $\lceil \#(\mathsf{b})/3 \rceil$ rectangles each of which contains at most two points from the computed solution. (Note that the inequality holds since $\#(\mathsf{b})$ has integral value.) ∎

Remark 4.23. There are now improved bounds on the size of relative approximation for points and rectangles [Ezr13]. This suggests that further improvements to the results in this section are possible, and we leave this as open problem for further research.

5 Packing points into fat triangles

In this section, we give a bi-criteria approximation for packing points into a set of α-fat triangles. More precisely, we consider the following problem.

Problem 5.1. (PackPntsInFatTriangs) *Given a weighted set P of points and a set $\mathcal{T}$ of α-fat triangles in the plane such that each triangle $\triangle$ has a capacity $\#(\triangle)$, find a maximum weight subset $\mathcal{A} \subseteq \mathsf{P}$, such that, for each $\triangle \in \mathcal{T}$, the number of points of $\mathcal{A}$ contained in $\triangle$ is at most $\#(\triangle)$.*

The approximation algorithm uses the following building blocks:

(A) We prove that, for a given point set, there exists a small number of canonical sets such that for any fat triangle that covers at most k points, there exists a constant number of these canonical sets whose union covers exactly the same points. Showing this result is quite technical and requires non-trivial modifications of the work of Aronov *et al.* [AES10] (in particular, their work does not imply this result). This is delegated to Section 5.4, see Theorem 5.6 for the exact result.

(B) An algorithm for approximating the unit capacity case. This follows by an algorithm similar to the one in Lemma 4.21, see Lemma 5.2 for details. Note that this uses the result from (A) to get the required approximation.

(C) A partition scheme that shows that a fat triangle (with a measure defined over it) can be "partitioned" into $O(k)$ triangles such that any triangle in this partition has measure at most $1/k$; see Lemma 5.3.

Putting these components together yields the approximation algorithm; see Theorem 5.5 for details.

5.1 The unit capacity case

Lemma 5.2. *Given an instance* $\mathsf{G} = (\mathsf{V}, \mathcal{T})$ *of* PackPntsInFatTriangs *with unit capacities, one can compute a subset* $X \subseteq \mathsf{V}$ *such that the total weight of* X *is* $\Omega\left(\mathrm{opt}/\log^6 \mathcal{E}\right)$ *and each triangle of* $\mathcal{T}$ *contains at most* 9 *points of* P.

Proof: We follow the proof of Lemma 4.21. We first use Lemma 2.5 to sparsify the given instance. We now have a set of $\mathsf{P} \subseteq \mathsf{V}$ of $t = \Theta(\mathcal{E} \log \mathcal{E})$ points and a corresponding fractional solution that is feasible. The value of the fractional solution on G_P is $\Omega(\mathrm{opt})$, and as such we restrict our search for a solution to P.

Furthermore, we can assume that the value assigned to each point of P by this fractional solution is exactly $1/M$ — we replicate a point i times if it is assigned value i/M — where $M = O(\log \mathcal{E})$. Note that none of the triangles of $\mathcal{T}$ contains more than M points of P. In particular, by Theorem 5.6, one can construct a set $\mathcal{Z}$ of regions of size $O\left(M^3 t \log^2 t\right)$ such that, for every triangle of $\triangle \in \mathcal{T}$, there exists a subset $\{z_1, \ldots, z_k\} \subseteq \mathcal{Z}$ of at most 9 regions (i.e., $k \leq 9$) such that $\mathsf{P} \cap \triangle = \mathsf{P} \cap \left(\cup_{i=1}^k z_i\right)$. We build a conflict graph G over P connecting two points if (i) they are both contained in a triangle of $\mathcal{T}$, and (ii) there is a set of $\mathcal{Z}$ that contains both of them. Since the number of sets in $\mathcal{Z}$ is $O\left(M^3 t \log^2 t\right)$, and each such set has size at most M, it follows that the number of edges in the resulting graph G is $O\left(M^5 t \log^2 t\right)$, and the number of vertices is $t = \Theta(\mathcal{E} \log \mathcal{E})$ vertices. As in the proof of Lemma 4.21, we also add edges between replicated points (since these edges do not affect our analysis, we ignore them for the sake of simplicity of exposition).

The graph G has average degree $O\left(M^5 \log^2 t\right) = O\left(\log^7 \mathcal{E}\right)$, and the total weight of the vertices is $\Theta(\mathrm{opt} \log \mathcal{E})$. Therefore, by Turán's theorem, one can compute an independent set of vertices in this graph of weight $\Omega(w(\mathsf{P}) / (\text{average degree} + 1)) = \Omega(\mathrm{opt}/\log^6 \mathcal{E})$.

Finally, it is easy to verify that any triangle in $\mathcal{T}$ contains at most 9 points of this independent set. ■

5.2 Covering a measure on a fat triangle

At this point we would like to use Lemma 5.2 in order to get a bi-criteria approximation for the case in which the capacities are arbitrary, as we did in Theorem 4.22. However, doing so directly proves more challenging for fat triangles than axis parallel rectangles. This is because our general procedure requires that, given an object x of a given type such that the total fractional value of the points in x is non-zero, we need to be able to decompose x into $O(\#(x))$ smaller objects of the same type such that, for each smaller object, the fractional value of the points in the object is only a constant. This can easily be done for axis parallel rectangles by using vertical splitting lines (as was done in Theorem 4.22), but it is more challenging for fat triangles. However, the following lemma shows that such a decomposition is still possible for fat triangles.

Lemma 5.3. *Let μ be a measure defined over the plane, and consider a fat triangle $\triangle$. Then, for any integer k, one can cover $\triangle$ by at most $18k$ fat triangles, such that the measure of each of these triangles is at most $\mu(\triangle)/k$.*

Proof: To simplify the presentation we assume that $\mu(\triangle) = 1$. We recursively build a tree on $\triangle$ by partitioning the original triangle $\triangle$ into 4 similar triangles as shown in Figure 5. Each node of this tree corresponds to a triangle from this recursive construction. We stop the recursive partition for a node v as soon as the measure of the triangle $\triangle_v$ associated with it is at most $1/k$.

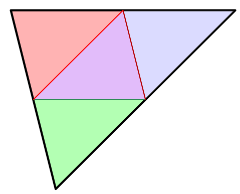

Figure 5: Partition.

Once we have the tree, we select a set S of nodes of the tree as follows. We find the lowest node v in the tree such that the measure of its corresponding triangle is at least $1/k$. We add the node v to S and we treat the measure inside the triangle corresponding to v as being 0. We repeat this process until the measure left uncovered is smaller than $1/k$, at which point we take the lowest node covering the remaining measure and we add it to S. We also add the root of the tree to S. Note that the set S contains at most $k + 1$ nodes: since the initial measure is one, we added at most k nodes to S that are not the root.

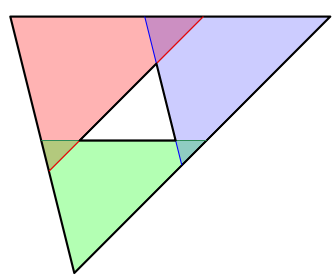

Figure 6: Triangle with hole.

We also add all the nodes in this tree that are the least common ancestor (LCA) of a pair of nodes in S. Let S' be the resulting set of nodes. Now we can show that, for any tree T and any subset R of nodes of T, the set of all LCA nodes of all of the pairs of nodes in R has size at most $|R| - 1$[9]. Therefore there are at most $|S| - 1$ nodes in $S' \setminus S$ and thus the size of S' is at most $2\,|S| \leq 2(k + 1)$. Let $\mathcal{T}$ be the set of triangles induced by the triangles corresponding to the children of the nodes of S' (i.e., every node of S' gives rise to four triangles). Consider the partition of the original triangle formed by $\mathcal{T}$. It is

[9] Let u, v be the pair of nodes in R whose LCA has maximum depth. Let z be the LCA of u and v, and let $R' = (R \setminus \{u, v\}) \cup \{z\}$. The pairs in R and the pairs in R' have the same set of LCA nodes and thus the number of LCA nodes of a set of size r satisfies the recurrence $f(r) \leq f(r - 1) + 1$ and $f(2) = 1$.

easy to verify that every face in this arrangement has measure at most $1/k$, and the number of faces of this arrangement, denoted by n, is at most $8k$ (observe that since the triangles arise out of a recursive partition, a pair of such triangles is either disjoint or contained in each other). Furthermore, each face of this arrangement is either a triangle (which is a scaled and rotated copy of the original triangle), or the difference of two triangles where one contains the other (having this property is why we took the children of every LCA). We will refer to a face which is the difference of two triangles as an annulus face. Let n' be the number of annulus faces in this arrangement. Clearly, $n' \leq |S'| = 2k$, as one can charge an annulus face to the node of S' that induced the hole in this face.

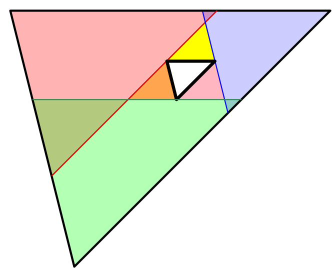

Figure 7: Fixing the hole.

We claim that an annulus face can be covered by the union of six translated and rotated copies of the original triangle. The easy case, is when the hole is a scaled and translated reflection of the outside triangle, see Figure 6, where three triangles are sufficient. The other case is when the hole is a scaled translated copy of the outer triangle, which by attaching to the hole three translated and rotated copies of the hole, gets reduced to the other case, see Figure 7. (The property used here implicitly is that the outer triangle of the annulus can be partitioned into translated and rotated copies of the hole triangle, as the hole arises out of a recursive partition of the outer triangle.)

As such, there are $n - n'$ triangular faces in this arrangement and n' annulus faces. Thus, the original triangle can be covered by $(n - n') + 6n' = n + 5n' \leq 8k + 10k \leq 18k$ translated and scaled copies of the original triangle that cover it completely, and no triangle in this collection has measure that exceeds $1/k$. ■

Remark 5.4. Given a weighted set of n points defining the measure inside the given fat triangle, the cover of Lemma 5.3 can be computed in $O(n \log n)$ time. This requires using known techniques used in constructing compressed quadtrees, see [Har11] for details.

5.3 The result

Theorem 5.5. *Given an instance of* $(\mathsf{V}, \mathcal{T})$ *of* PackPntsInFatTriangs *(with arbitrary capacities), one can compute, in polynomial time, a subset* $X \subseteq \mathsf{V}$ *that is* $\left(O\left(\log^6 \mathcal{E}\right), 9\right)$*-approximation to the optimal solution.*

Proof: Compute a fractional solution to the given instance. For any triangle $\triangle$ in the plane, we denote by $\mathcal{E}(\triangle) = \sum_{\mathsf{p} \in \mathsf{V} \cap \triangle} x_{\mathsf{p}}$ the total mass of the fractional solution inside $\triangle$. Next, we get a constant capacity instance out of $(\mathsf{V}, \mathcal{T})$ by replacing each triangle of $\mathcal{T}$ by a "few" triangles covering it, such that the total mass of the fractional solution inside each of these new triangles is at most $c = 4 \cdot 18 \cdot 9$. Formally, consider a triangle $\triangle \in \mathcal{T}$, and let $k = \lceil \#(\triangle)/c \rceil$. If $\#(\triangle) \leq c$ then there is nothing to do (as $\mathcal{E}(\triangle) \leq \#(\triangle) \leq c$), so we assume that $\#(\triangle) > c$. Applying the algorithmic version of Lemma 5.3, see Remark 5.4, we cover $\triangle$ with at most

$$18k = 18 \left\lceil \frac{\#(\triangle)}{c} \right\rceil = 18 \left\lceil \frac{\#(\triangle)}{4 \cdot 18 \cdot 9} \right\rceil \leq \left\lceil \frac{\#(\triangle)}{2 \cdot 9} \right\rceil$$

triangles, where the total mass of the fractional solution inside each of them is at most $\mathcal{E}(\triangle)/k \leq \#(\triangle)/\lceil \#(\triangle)/c \rceil \leq c$.

Now, consider the generated instance with these new triangles, where each such triangle has capacity one. To this end, scale down the solution of the LP by a factor of c. Clearly, we now have a uniform capacity instance with an associated (valid) fractional solution having value $\Omega(\mathrm{opt})$ (where opt is the optimal LP value for the original instance). Furthermore, any solution to this unit capacity instance, would correspond to a solution to the original instance (since we covered each original triangle with at most $18k \leq \left\lceil \frac{\#(\triangle)}{2 \cdot 9} \right\rceil \leq \#(\triangle)$ new unit capacity triangles). Plugging this instance into Lemma 5.2 yields the required approximation. Specifically, every triangle $\triangle \in \mathcal{T}$ contains at most $9 \lceil \#(\triangle)/(2 \cdot 9) \rceil \leq \max(9, \#(\triangle))$ points of the computed set of points. ∎

5.4 Canonical decomposition for fat triangles

In this section, we show that given a set P of n points in the plane, and a parameter k, one can compute a set $\mathcal{S}$ of $O\big(k^3 n \log^2 n\big)$ regions, such that for any α-fat triangle $\triangle$, if $|\triangle \cap \mathsf{P}| \leq k$, then there exists (at most) 9 regions in $\mathcal{S}$ whose union has the same intersection with P as $\triangle$ does.

Our construction follows closely the argumentation of Aronov *et al.* [AES10]. However, our construction is (somewhat) different and (arguably) simpler since we are considering a "dual" problem to theirs. In particular, since modifying Aronov *et al.* [AES10] to get our result is not obvious, we present it here in detail.

5.4.1 Initial setup

To construct the set of regions, $\mathcal{S}$, we will use an approach similar to that of Lemma 4.18. As was observed in [AES10], we can restrict our attention to axis aligned right triangles whose hypotenuse differs by no more than (say) one degree from $-45°$, as measured from the positive x-axis (i.e., it is near isosceles and faces to the right). In the following, let $\triangle$ be an arbitrary such triangle that contains at most k points.

We first construct a two level interval tree on P, where the first level partitions the points based on their x-coordinate, and the second level based on their y-coordinate (and the splitting line for each node goes through the median point). Let v be the highest node in the first level of the interval tree whose corresponding split line, ℓ, intersects $\triangle$. Let $\triangle_{\leftarrow}$ and $\triangle_{\rightarrow}$ denote the portion of $\triangle$ to the left or right of ℓ, respectively. Also, let u be the highest node in the second level tree rooted at the left child of v whose corresponding split line, $\wp$, intersects $\triangle_{\leftarrow}$, and let $\triangle_{\nwarrow}$ and $\triangle_{\swarrow}$ denote the portion of $\triangle_{\leftarrow}$ above or below $\wp$, respectively. (Note that we may assume that there exists split lines ℓ and $\wp$ that intersect $\triangle$ and $\triangle_{\leftarrow}$, respectively, since such regions that contain no points can be skipped). In the following, let p denote the point of intersection between ℓ and $\wp$. See Figure 8.

We now construct sets of canonical regions, $\mathcal{T}_{\rightarrow}$, $\mathcal{T}_{\nwarrow}$, and $\mathcal{T}_{\swarrow}$ such that for any choice of $\triangle$ there exists constant number of regions $r_1, \ldots, r_m$ in $\mathcal{T}_{\rightarrow} \cup \mathcal{T}_{\nwarrow} \cup \mathcal{T}_{\swarrow}$, such that $\triangle \cap \mathsf{P} = \bigcup_i (r_i \cap \mathsf{P})$, and $m \leq 9$.

We achieve this by showing that in each case (i.e., $\triangle_{\nwarrow}$, $\triangle_{\swarrow}$ and $\triangle_{\rightarrow}$) the region r under consideration can be transformed into a polygonal region with a constant number of points of P (or orientations)

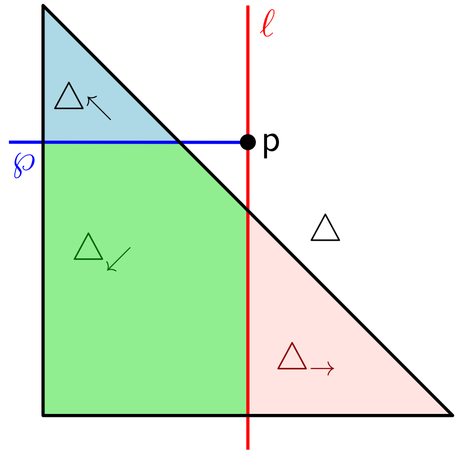

Figure 8: Decomposing △.

defining its bounding edges, and whose intersection with P is the same as r [10]. We then show that the number of such regions needed for a particular choice ℓ and $\wp$ is $O(nk^3)$.

In the following, let P_v be the subset of points of P stored in the subtree rooted v, and let $\mathsf{P}_{u,v}$ be the set of points stored in the subtree rooted at u.

5.4.2 Handling the right portion of the triangle ($\triangle_{\rightarrow}$)

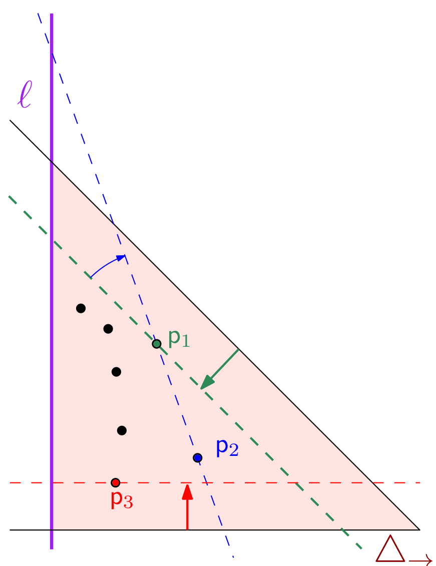

Figure 9: Handling $\triangle_{\rightarrow}$.

For any $\triangle$, we know that $\triangle_{\rightarrow}$ will be a homothet of $\triangle$, whose vertical edge lies on ℓ, see Figure 8. We now transform $\triangle_{\rightarrow}$ uniquely such that two points of P lie on its hypotenuse (or one point and the hypotenuse is at an angle of $-46°$) and one point of P lies on its bottom edge. Start by translating the

[10] For each point on the bounding edges, we will need to specify whether it is inside or outside the canonical region. This can be encoded by a string of length c, where c is some constant bounding the number of defining boundary points. Hence we can specify the inclusion or exclusion of the boundary points while only increasing the number of canonical regions by a factor of $2^c = O(1)$, and hence we will not need to worry about such issues.

hypotenuse towards the lower left corner of $\triangle_{\rightarrow}$ (while clipping it to $\triangle_{\rightarrow}$) until it hits a point, p_1. Next rotate the hypotenuse clockwise around p_1 until it hits a second point p_2, or its orientation is $-46°$ (as we rotate we modify its length so that one endpoint of the hypotenuse stays on ℓ and the other on the base of $\triangle_{\rightarrow}$). Next translate the base of $\triangle_{\rightarrow}$ straight upwards (while clipping it and the hypotenuse as to maintain a right triangle) until it hits a third point p_3 (which may be the same as the rightmost point out of p_1 and p_2). Observe that the resulting region has the same intersection with P as $\triangle_{\rightarrow}$ (except maybe for the points on the boundary). See Figure 9.

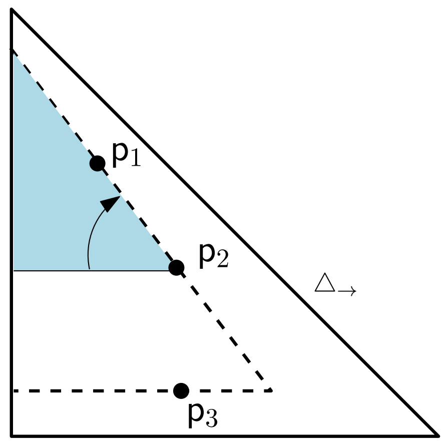

Figure 10

We now bound the number of such resulting regions. Assume that p_2 lies to the right of p_1 (the other case is handled similarly). There are $n_v = |\mathsf{P}_v|$ possible choices for p_2. Now consider the horizontal line segment that connects ℓ and p_2. Rotate this segment clockwise around p_2 (while increasing its length so that the other endpoint stays on ℓ) until it hits p_1, see Figure 10. We know that all the points we hit in this sweeping process lie in the computed region, and hence we can only have swept over k points before reaching p_1 (i.e., given p_2 there are at most k choices for p_1. If p_2 does not exist we start with the triangle formed by ℓ and a horizontal and $-46°$ line through p_1). Now imagine translating the horizontal segment connecting p_2 and ℓ straight downward till we hit p_3 (while increasing its length so that its right endpoint stays on the hypotenuse defined by p_1 and p_2). Again we know that all the points we hit in this sweeping process must be in our canonical region, and hence we can only have swept over k points before reaching p_3 (i.e., given p_2 and p_1, there are at most k choices for p_3).

Hence there are $O(n_v k^2)$ such canonical regions for the node v. Since $\mathsf{P}_{v_i} \cap \mathsf{P}_{v_j} = \emptyset$ for any v_i, v_j at the same level in the top layer tree, summing across a given level gives $O(nk^2)$ canonical regions, where $n = |\mathsf{P}|$. Thus, summing over all nodes in the top layer tree gives $O(nk^2 \log n)$ such canonical regions overall.

5.4.3 Handling the top left portion of the triangle ($\triangle_{\nwarrow}$)

Here we must consider two cases, based on the possible locations of p. If $\mathsf{p} \notin \triangle_{\nwarrow}$ (see Figure 8), we have a homothet of $\triangle$ whose bottom edge lies on $\wp$, and therefore we can argue as in the $\triangle_{\rightarrow}$ case, that this gives rise to $O(n_{u,v} k^2)$ different canonical regions, where $n_{u,v} = |\mathsf{P}_{u,v}|$. Summing over all possible nodes u and v gives $O\big(nk^2 \log^2 n\big)$ such canonical regions overall.

Now suppose that $\mathsf{p} \in \triangle_{\nwarrow}$, see Figure 11. In this case, we can extend $\triangle_{\nwarrow}$ to get a homothet of $\triangle$ whose right side was cut off by l in order to get $\triangle_{\nwarrow}$ ($\triangle'$ in Figure 11). Clearly, we have that $\mathsf{P}_{u,v} \cap \triangle_{\nwarrow} = \mathsf{P}_{u,v} \cap \triangle'$. We can now generate a canonical region for $\triangle'$ in a similar fashion as the $\triangle_{\rightarrow}$

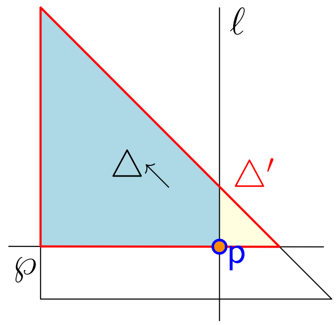

Figure 11

case, since it is just a homothet of $\triangle$ with its base lying on $\wp$, and then we can cut off the portion to the right of l. This would imply that we can generate $O(n_{u,v}k^2)$ such canonical regions for the nodes u and v, and so overall there are $O\big(nk^2 \log^2 n\big)$ such canonical regions.

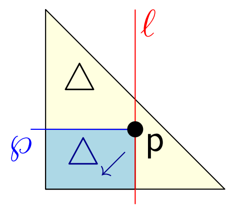

Figure 12

5.4.4 Handling the bottom left portion of the triangle ($\triangle_{\swarrow}$)

Again we consider two cases, based on the possible locations of p. If $\mathsf{p} \in \triangle_{\swarrow}$ (see Figure 12), then $\triangle_{\swarrow}$ is an axis parallel rectangle such that one of its sides lies on ℓ (and another side lies on $\wp$). Hence by the proof of Lemma 4.18, in this case $\triangle_{\swarrow}$ gives rise to $O(k^2 n \log n)$ canonical regions overall.

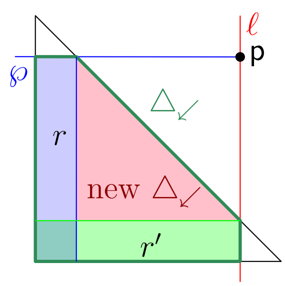

Figure 13

Now we consider (what is by far) the hardest case, when $\mathsf{p} \notin \triangle_{\swarrow}$. In order to handle this case we will need to break up $\triangle_{\swarrow}$ as follows. Observe that $\triangle_{\swarrow}$ is a rectangular region whose upper right corner was cut off by the hypotenuse of $\triangle$. First, we reduce $\triangle_{\swarrow}$ into a homothet of $\triangle$, by removing rectangles r and r' from the left and bottom parts of $\triangle_{\swarrow}$, respectively (see Figure 13). This can be done since we already observed that by the proof of Lemma 4.18 we can construct a set of $O\big(nk^2 \log^2 n\big)$ canonical rectangles such that any rectangle (with a side on one of the split lines) has the same intersection with P as one of the canonical rectangles. For simplicity we continue to refer to the remaining part of $\triangle_{\swarrow}$ as just $\triangle_{\swarrow}$.

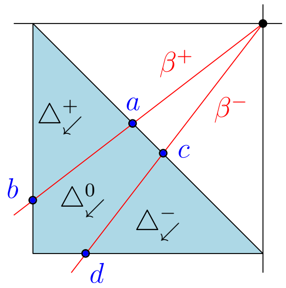

Figure 14

We now break up $\triangle_{\swarrow}$ into three regions. Let β^+ and β^- denote the rays emanating from p at angles $-140°$ and $-130°$, respectively (again, as measured clockwise from the positive x-axis). These two lines split $\triangle_{\swarrow}$ into three regions, which we will denote in their counterclockwise order as $\triangle^+_{\swarrow}$, $\triangle^0_{\swarrow}$ and $\triangle^-_{\swarrow}$ (see Figure 14). Let a and b denote the intersection of β^+ with the hypotenuse and the left edge of $\triangle_{\swarrow}$, respectively. Similarly, let c and d denote the intersection of β^- with the hypotenuse and bottom edge of $\triangle_{\swarrow}$, respectively.

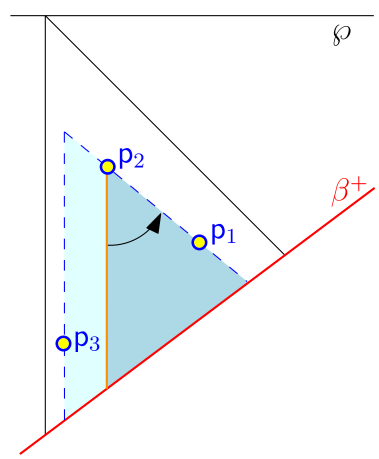

Figure 15

Handling the top and bottom parts of $\triangle_{\swarrow}$ (i.e., $\triangle^+_{\swarrow}$ and $\triangle^-_{\swarrow}$). We now construct the canonical regions for $\triangle^+_{\swarrow}$. The construction is nearly identical to that for $\triangle_{\rightarrow}$ and is included for the sake of completeness. The construction for $\triangle^-_{\swarrow}$ is omitted as it is symmetric to the $\triangle^+_{\swarrow}$ case.

Start by translating the part of the boundary that intersects the hypotenuse of $\triangle$ towards the lower left corner of $\triangle^+_{\swarrow}$ (while clipping it to $\triangle^+_{\swarrow}$) until it hits a point, p_1. Next rotate this edge counterclockwise around p_1 until it hits a second point p_2, or its orientation is $-44°$ (as we rotate we modify its length so that one endpoint stays on β^+ and the other on the boundary $\triangle^+_{\swarrow}$). Next translate the vertical edge of $\triangle^+_{\swarrow}$ to the right (while clipping it to $\triangle^+_{\swarrow}$) until it hits a third point, p_3.

As for the number of such resulting regions, assume that p_2 lies to the left of p_1 (the other case is handled similarly). There are $n_{u,v} = |\mathsf{P}_{u,v}|$ possible choices for p_2. Now consider the vertical line segment that connects β^+ and p_2. Imagine rotating this segment counterclockwise around p_2 (while increasing its length so that the other endpoint stays on β^+) until it hits p_1. We know that all the points we hit in this sweeping process must be in our canonical region, and hence we can only have swept over k points before reaching p_1 (if p_2 does not exist we start with the triangle formed by β^+ and a vertical and $-44°$

line through p_1). Now imagine translating the vertical segment connecting p_2 and β^+ to the left until we hit p_3 (while increasing its length so that its top endpoint stays on the line defined by p_1 and p_2 and its bottom endpoint on β^+). Again we know that all the points we hit in this sweeping process must be in our canonical region, and hence we can only have swept over k points before reaching p_3. Hence there are $O(n_{u,v}k^2)$ such canonical regions for a pair nodes u and v. Thus overall there are $O\big(nk^2 \log^2 n\big)$ such canonical regions.

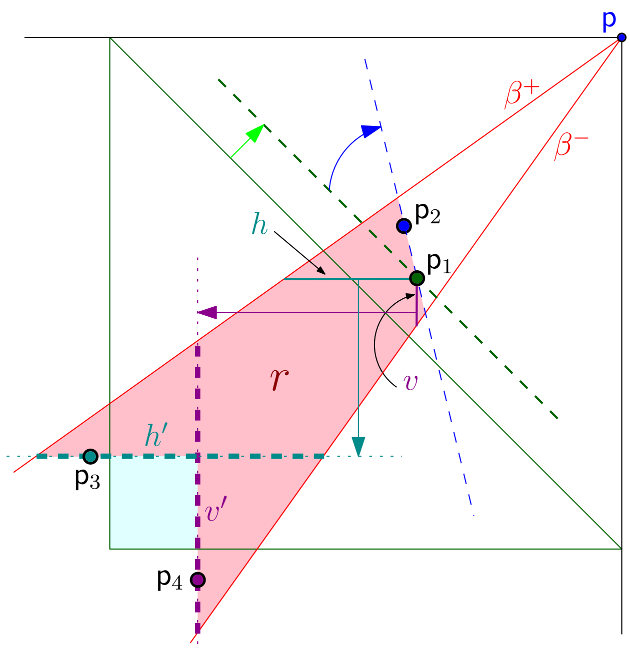

Figure 16: Handling the middle.

Handling the middle part of $\triangle_{\swarrow}$ (i.e., $\triangle^0_{\swarrow}$). Let $\mathsf{P}^0_{u,v}$ denote the subset of $\mathsf{P}_{u,v}$ that lies in between β^+ and β^-. Consider the segment that is the intersection of the hypotenuse of $\triangle$ with $\triangle^0_{\swarrow}$. Translate this segment towards p (while clipping it to $\triangle^0_{\swarrow}$) until it hits a point p_1. Then rotate this segment clockwise around p_1 until it hits a point p_2, or it becomes vertical. Without loss of generality, assume that p_1 lies to the right of p_2. Let the horizontal (resp. vertical) line connecting p_1 and β^+ (resp. β^-) be called h (resp. ν). Translate h downwards (resp. ν to the left), while enlarging it so that one endpoint stays on β^+ (resp. β^-), until either it hits the lowest (resp. furthest to the left) point of $\triangle^0_{\swarrow}$ or a point outside of $\triangle^0_{\swarrow}$. Let this point be denoted p_3 (resp. p_4), and let h' (resp. ν') be the final translation of h (resp. ν). See Figure 16.

Consider the region, r, bounded by the portion of h' to the left of p_4, the portion of ν' below p_3, β^+, β^-, and the line going through p_1 and p_2 (this is the red shaded region in Figure 16). First observe that if both p_3 and p_4 lie outside of $\triangle^0_{\swarrow}$ then r will not cover all the points in $\triangle^0_{\swarrow} \cap P^0_{u,v}$. Namely, the points lying in the rectangle defined by h', ν', and the vertical and horizontal edges of $\triangle^0_{\swarrow}$ might not be covered by r (see Figure 16). However, we already constructed a set of $O(k^2 n \log n)$ canonical rectangles, which we know contains two canonical rectangles that cover these points, and as such we do not have to worry about covering these points. Clearly, all the points of $\triangle^0_{\swarrow} \cap P^0_{u,v}$ either lie in this rectangle or in r. Next observe that there are no points of $\mathsf{P}^0_{u,v}$ that lie in r that are not in $\triangle_{\swarrow} \cap P^0_{u,v}$. This follows by a tedious but straightforward argument since (i) the hypotenuse was within one degree of $-45°$, (ii) $h \cap \beta^+$ lies to the right of b, and (iii) $\nu \cap \beta^-$ lies above d.

We now bound the number of canonical regions of type r. There are $\left|P^0_{u,v}\right|$ possible choices for p_1 (which again we assume is to the right of p_2). Now consider rotating h clockwise around p_1 until we hit p_2. We know from above that all the points we sweep past in this process must be contained in $\triangle^0_{\swarrow} \cap P^0_{u,v}$ and so given p_1 there are at most k possible choices for p_2. Now consider translating h downward (resp. ν to the left) until we hit p_3 (resp. p_4). Again, from above we know that all the points we sweep over in this process must be contained in $\triangle^0_{\swarrow} \cap P^0_{u,v}$ and so given p_1 and p_2, there are at most k possible choices for p_3 (resp. p_4). Hence there are $O\left(k^3 \left|P^0_{u,v}\right|\right)$ such canonical regions for a given pair of nodes u and v, and so overall there are $O\left(k^3 n \log^2 n\right)$ such canonical regions.

5.4.5 Putting things together

Summing the above bounds over all choices of the nodes u and v results overall in $O\left(k^3 n \log^2 n\right)$ canonical regions. Furthermore, for any choice of $\triangle$, we showed above that there exists a set of at most 9 of theses canonical regions whose (union of) intersections with P is the same as that of $\triangle$. We thus get the following result.

Theorem 5.6. *Given a set P of n points in the plane, a parameter k, and a constant $\alpha > 0$, one can compute, in polynomial time, a set $\mathcal{S}$ of $O\left(k^3 n \log^2 n\right)$ regions, such that for any α-fat triangle $\triangle$, if $|\triangle \cap \mathsf{P}| \leq k$, then there exists (at most) 9 regions in $\mathcal{S}$ whose union has the same intersection with P as $\triangle$ does.*

6 PTAS for Unweighted Disks and Points

In this section, we consider instances of the PackRegions problem in which the regions are disks with unit weights and all points have unit capacities. This is the discrete independent set problem for disks (the independence constraints are defined by the points). Chan and Har-Peled provided a constant factor approximation for the weighted version of this problem [CH11]. We now outline a PTAS for such instances based on the local search technique. The algorithm and proof are an extension of those of Chan and Har-Peled [CH11, CH09], and Mustafa and Ray [MR10].

The algorithm. Since all of the regions have unit weight, we may assume that no region is completely contained in another. We say that a subset L of $\mathcal{D}$ is ***b-locally optimal*** if L is a pointwise independent set and one cannot obtain a larger pointwise independent set by removing $\ell \leq b$ regions of L and inserting $\ell + 1$ regions of $\mathcal{D} \setminus \mathsf{L}$.

Our algorithm constructs a b-locally optimal solution using local search, where b is some suitable constant. We start with $\mathsf{L} \leftarrow \emptyset$. We consider each subset $X \subseteq \mathcal{D} \setminus \mathsf{L}$ of size at most $b+1$: if X is a pointwise independent set and the set $Y \subseteq \mathsf{L}$ of regions pointwise intersecting the objects of X has size at most $|X| - 1$, we set $\mathsf{L} \leftarrow (\mathsf{L} \setminus Y) \cup X$. This is continued till no further improvement is possible.

Running time. Every such swap increases the size of L by at least one, and as such it can happen at most $n = |\mathcal{D}|$ times. Therefore the running time is bounded by $O\left(n^{b+3} b \,|\mathsf{P}|\right)$, since there are $\binom{n}{b+1}$ subsets X to consider and for each such subset X it takes $O(nb\,|\mathsf{P}|)$ time to compute Y.

Analysis. Let opt be the maximum pointwise independent set, and let L be the b-locally optimal solution returned by our algorithm. If we can show that the pointwise intersection graph of opt $\cup\, \mathsf{L}$ is planar then the analysis in [CH11] will directly imply that $|\mathsf{L}| \geq \left(1 - O(1)/\sqrt{b}\right) |\text{opt}|$.

We map the disks in opt and L to sets of points Q_{opt} and Q_{L} in $\mathbb{R}^3$, respectively, and we map the points in P to a set of halfspaces H_{P}, by using the lifting of disks to planes and points to rays, and then dualizing the problem (see Section 4.2). Mustafa and Ray prove that a range space defined by a set of points and halfspaces in $\mathbb{R}^3$ has the ***locality condition***, which is defined as follows.

Definition 6.1 ([MR10]). *A range space $R = (\mathsf{P}, \mathcal{D})$ satisfies the* **locality condition** *if for any two disjoint subsets $R, B \subseteq \mathsf{P}$, it is possible to construct a planar bipartite graph $G = (R, B, E)$ with all edges going between R and B such that for any $D \in \mathcal{D}$, if $D \cap R \neq \emptyset$ and $D \cap B \neq \emptyset$, then there exist two vertices $u \in D \cap R$ and $v \in D \cap B$ such that $(u, v) \in E$.*

Since opt and L are both pointwise independent sets, we know each point in P can intersect at most one disk from opt and at most one disk from L. Hence each halfspace in H_{P} can contain at most one point from Q_{opt} and at most one point from Q_{L}. Since points and halfspaces in $\mathbb{R}^3$ have the locality condition, setting $R = \mathsf{L}$ and $B = \mathrm{opt}$ immediately implies that there is a planar graph on the vertex set $\mathsf{L} \cup \mathrm{opt}$ such that any vertex from L and any vertex from opt that are in the same halfspace are adjacent. In particular, the intersection graph is planar.

Theorem 6.2. *Given a set of n unweighted disks and a set of m points in the plane (with unit capacities), any b-locally optimal pointwise independent set has size $\geq \left(1 - c_2/\sqrt{b}\right)\mathrm{opt}$, where opt is the size of the maximum pointwise independent set of the disks, and c_2 is a constant. In particular, one can compute a discrete independent set of size $\geq (1-\varepsilon)\mathrm{opt}$, in time $mn^{O(1/\varepsilon^2)}$.*

Corollary 6.3. *There is a PTAS for the following problems:*
(A) *Instances of* PackHalfspaces *in which each halfspace has unit weight, and each point has unit capacity.*
(B) *Instances of* PackRegions *in which each region is a unit-weight disk, and each point has unit capacity.*
(C) *Instances of* PackPoints *in which each region is a unit-capacity disk, and each point has unit weight.*

7 Hardness of approximation

7.1 Packing same size fat triangles into points

Here we show that PackRegions (Problem 1.2) does not have a PTAS, even if the regions have unit weight and their union complexity is linear. We show that the problem is APX-hard using a reduction from the maximum bounded 3-dimensional matching problem. Since maximum bounded 3-dimensional matching is APX-complete [Kan91], this will imply the claim (unless P = NP).

Theorem 7.1. *Unless P = NP there is no PTAS for* PackRegions *(Problem 1.2) even if the regions are unweighted, in the plane, and have linear union complexity. In particular, this holds if the regions are fat triangles of similar size. (See Corollary 4.3 (A) for the matching approximation algorithm.)*

Proof: Let $T \subseteq A \times B \times C$ be the input triples for an instance of maximum bounded 3-dimensional matching, where A, B, and C are disjoint subsets of some ground set X (for simplicity we assume $X = A \cup B \cup C$). For each element $x \in X$ we make a representative point v_x and place it arbitrarily on the unit circle in the plane and give it unit capacity. Let V_A, V_B, and V_C be the sets of representatives for A, B, and C (respectively). A triple in T thus corresponds to a triangle with one vertex in each of

V_A, V_B, and V_C. Clearly, finding a maximum packing of these triangles into these points is an instance of PackRegions. Moreover, a maximum packing here corresponds to a maximum set of triangles (triples) such that each point (element of X) is covered by at most one triangle. Therefore a PTAS for this problem translates to a PTAS for the maximum bounded 3-dimensional matching problem. (Note that this does not imply that there is no PTAS for other specific types of regions.)

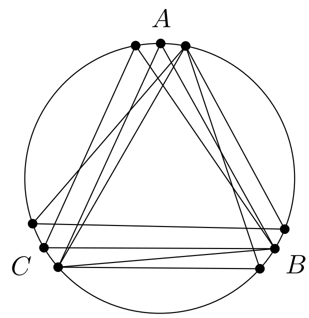

Figure 17: Packing triangles.

Now we show that we can make the triangles fat and of similar size, and hence there is no PTAS even in the case of linear union complexity. Let the range of a set of representative points be the angle around the circle between the farthest two points of the set, and let the center of a set be the midpoint on the circle between the farthest two points of the set. Instead of placing the points arbitrarily, we will place the points so that the range of each of V_A, V_B, or V_C is less than (say) five degrees. Moreover, we place the points so that the centers of V_A, V_B, and V_C are 120 degrees apart. In this case the triangles all have roughly the same size and are nearly equilateral. It is known that such a set of triangles has linear union complexity [MMP+94]. Hence, by the above reduction, even in this case where the regions are restricted to have linear union complexity (and even more specifically when they are restricted to be fat triangles of roughly the same size), we cannot get a PTAS. ■

7.2 Packing points into fat triangles

Lemma 7.2. *There is an approximation-preserving reduction from the* Independent Set *problem in general graphs to the* PackPoints *problem. In particular, for instances of the problem* PackPoints *in which the regions are fat triangles with unit capacities and the points are unweighted, no approximation better than $\Omega(n^{1-\varepsilon})$ is possible in polynomial time, for any constant $\varepsilon > 0$, unless $\mathsf{P} = \mathsf{NP}$.*

Proof: Consider an instance of the Independent Set problem, namely a graph $G = (V, E)$. Let $n = |V|$. Place n distinct points on the unit circle (arbitrarily) and map every vertex of V to a unique point of the resulting set of points P. For every edge $uv \in E$, consider the segment $\mathsf{p}_u\mathsf{p}_v$, where p_u and p_v are the points corresponding to u and v in P. We construct a fat triangle containing $\mathsf{p}_u\mathsf{p}_v$ by connecting p_u, p_v, and a third vertex in the interior of the unit disk, and this can be done so the resulting triangle has fatness at most 2. We add this triangle to our set of regions $\mathcal{D}$, and assign it capacity one.

Clearly, solving the resulting instance $(\mathsf{P}, \mathcal{D})$ of PackPoints is equivalent to solving the Independent Set problem for G. The claim now follows from the hardness results known for the Independent Set problem [Has99]. ■

8 Conclusions

In this paper, we presented a general framework for approximating geometric packing problems with non-uniform constraints. We then applied this framework in a systematic fashion to get improved algorithms for specific instances of this problem, many of which required additional non-trivial ideas. There are several special cases of this problem for which we currently do not know any useful approximation; for example, the special case of packing axis-parallel boxes into points, in which the boxes are in four dimensions is still open (see [Cha12] for recent relevant work). Making some progress on these special cases is an interesting direction for future work.

Acknowledgments

The authors thank Timothy Chan, Chandra Chekuri, and Esther Ezra for several useful discussions. In addition, the authors thank the anonymous referees for their detailed comments.

References

[AdBES14] B. Aronov, M. de Berg, E. Ezra, and M. Sharir. Improved bounds for the union of locally fat objects in the plane. *SIAM J. Comput.*, 43(2):543–572, 2014.

[AES10] B. Aronov, E. Ezra, and M. Sharir. Small-size ε-nets for axis-parallel rectangles and boxes. *SIAM J. Comput.*, 39(7):3248–3282, 2010.

[AKS98] P. K. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Comput. Geom. Theory Appl.*, 11:209–218, 1998.

[Cap99] P. Cappanera. A survey on obnoxious facility location problems. Technical report, University of Pisa, 1999.

[CC07] M. Chlebík and J. Chlebíková. The complexity of combinatorial optimization problems on d-dimensional boxes. *SIAM J. Discrete Math.*, 21(1):158–169, 2007.

[CC09] P. Chalermsook and J. Chuzhoy. Maximum independent set of rectangles. In *Proc. 20th ACM-SIAM Sympos. Discrete Algorithms*, pages 892–901, 2009.

[CCH09] C. Chekuri, K. L. Clarkson., and S. Har-Peled. On the set multi-cover problem in geometric settings. In *Proc. 25th Annu. ACM Sympos. Comput. Geom.*, pages 341–350, 2009.

[CH09] T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. In *Proc. 25th Annu. ACM Sympos. Comput. Geom.*, pages 333–340, 2009.

[CH11] T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *CoRR*, abs/1103.1431, 2011.

[Cha03] T. M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46(2):178–189, 2003.

[Cha12] T. M. Chan. Conflict-free coloring of points with respect to rectangles and approximation algorithms for discrete independent set. In Tamal K. Dey and Sue Whitesides, editors, *Proc. 28th Annu. Sympos. Comput. Geom.* (SoCG), pages 293–302. ACM, 2012.

[Cla88] K. L. Clarkson. Applications of random sampling in computational geometry, II. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 1–11, 1988.

[CS89] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[CVZ11] C. Chekuri, J. Vondrák, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *CoRR*, abs/1105.4593, 2011. Preliminary version in STOC 2011.

[dBCvKO08] M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edition, 2008.

[EAS11] E. Ezra, B. Aronov, and M. Sharir. Improved bound for the union of fat triangles. In *Proc. 16th ACM-SIAM Sympos. Discrete Algorithms*, pages 1778–1785, 2011.

[EHR11] A. Ene, S. Har-Peled, and B. Raichel. Geometric packing under non-uniform constraints. *CoRR*, abs/1107.2949, 2011.

[EJS05] T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM J. Comput.*, 34(6):1302–1323, 2005.

[Ezr13] E. Ezra. Small-size relative (p, ε)–approximations for well-behaved range spaces. In Guilherme Dias da Fonseca, Thomas Lewiner, Luis Mariano Peñaranda, Timothy M. Chan, and Rolf Klein, editors, *Proc. 29th Annu. Sympos. Comput. Geom.* (SoCG), pages 233–242. ACM, 2013.

[FP11] J. Fox and J. Pach. Computing the independence number of intersection graphs. In *Proc. 22nd ACM-SIAM Sympos. Discrete Algorithms*, pages 1161–1165, 2011.

[GC11] E. Grant and T. M. Chan. Exact algorithms and APX-hardness results for geometric set cover. In *Proc. 23rd Canad. Conf. Comput. Geom.*, 2011.

[Har09] S. Har-Peled. Being fat and friendly is not enough. *CoRR*, abs/0908.2369, 2009.

[Har11] S. Har-Peled. *Geometric Approximation Algorithms*. Amer. Math. Soc., 2011.

[Has99] J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, pages 105–142, 1999.

[HS11] S. Har-Peled and M. Sharir. Relative (p, ε)-approximations in geometry. *Discrete Comput. Geom.*, 45(3):462–496, 2011.

[Kan91] V. Kann. Maximum bounded 3-dimensional matching is max SNP-complete. *Inform. Process. Lett.*, 37:27–35, January 1991.

[MMP+94] J. Matoušek, N. Miller, J. Pach, M. Sharir, S. Sifrony, and E. Welzl. Fat triangles determine linearly many holes. *SIAM J. Comput.*, 23(1):154–169, 1994.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[MR10] N. H. Mustafa and S. Ray. Improved results on geometric hitting set problems. *Discrete Comput. Geom.*, 44(4):883–895, 2010.

[SMC+07] P. G. Szabó, M. Cs. Markót, T. Csendes, E. Specht, L. G. Casado, and I. Garcãa. *New Approaches to Circle Packing in a Square: With Program Codes (Springer Optimization and Its Applications)*. Springer-Verlag New York, Inc., 2007.

[Sri01] A. Srinivasan. New approaches to covering and packing problems. In *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 567–576, 2001.

[Tam91] A. Tamir. Obnoxious facility location on graphs. *SIAM J. Discrete Math.*, 4(4):550–567, 1991.

[Var10] K. Varadarajan. Weighted geometric set cover via quasi-uniform sampling. In *Proc. 42nd Annu. ACM Sympos. Theory Comput.*, pages 641–648, 2010.

[WS11] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.