Polynomial-Time Approximation Schemes for Packing and Piercing Fat Objects

Timothy M. Chan*
Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1, Canada

October 15, 2001

Abstract

We consider two problems: given a collection of n fat objects in a fixed dimension,

1. (*packing*) find the maximum subcollection of pairwise disjoint objects, and
2. (*piercing*) find the minimum point set that intersects every object.

Recently, Erlebach, Jansen, and Seidel gave a polynomial-time approximation scheme (PTAS) for the packing problem, based on a shifted hierarchical subdivision method. Using shifted quadtrees, we describe a similar algorithm for packing but with a smaller time bound.

Erlebach *et al.*'s algorithm requires polynomial space. We describe a different algorithm, based on geometric separators, that requires only linear space. This algorithm can also be applied to piercing, yielding the first PTAS for that problem.

Abbreviated title. Packing and Piercing Fat Objects.

Keywords. Computational geometry, approximation algorithms, maximum independent set, hitting set, quadtrees, dynamic programming, separator theorems.

1 Introduction

In this paper, we study two related NP-hard geometric optimization problems and prove approximability results for the case when the objects involved are *fat* (see Section 2 for a precise definition).

Packing. Let $d \geq 2$ be a constant integer and $\varepsilon > 0$ be an arbitrarily small constant. In our first problem, we are given a collection $\mathcal{C}$ of n objects in d-dimensional space $\mathbb{R}^d$ and we want to find the largest subcollection in which no two objects intersect. The cardinality of the optimal solution will be denoted by $\text{pack}(\mathcal{C})$.

This problem may be viewed as a geometric version of the set packing problem [15]. (The nomenclacture is perhaps misleading to the geometry community, as the objects have fixed position in space and are not allowed to translate or rotate; nevertheless, for historical reasons, we will still use the term "packing" here.) Viewed alternatively, the problem is equivalent to finding the

*Work supported in part by an NSERC Research Grant. E-mail: `tmchan@uwaterloo.ca`.

maximum independent set in the intersection graph of the objects. The case where the objects are axis-aligned unit squares or unit circles in the plane is already NP-hard [14, 19], so it is desirable to find good approximation algorithms. Although the independent set problem in general graphs is hard to approximate, even to within a factor of $n^{1-\varepsilon}$ [16], the geometric restriction lets us obtain much better approximability results.

The problem, particularly in the rectangular case, has generated much interest from several groups of researchers motivated by different applications, such as VLSI design [17], data mining [7, 21], and map labeling [1]. For instance, in one (simplified) formulation of the map labeling problem, the goal is to maximize the number of rectangular labels that can be placed without overlapping, given a collection of possible label placements in the plane.

The case of unit squares or disks (or in higher dimensions, unit hypercubes or balls) was investigated in a seminal work by Hochbaum and Maass [17], who gave a polynomial-time approximation scheme (PTAS) based on a simple shifted grid strategy; more precisely, a solution within a $1+\varepsilon$ factor of the maximum can be found in $n^{O(1/\varepsilon^d)}$ time and linear space. The strategy generalizes to any collection of fat objects of roughly the same size (i.e., when the maximum-to-minimum size ratio is bounded by a constant). By applying dynamic programming along one of the dimensions, we can reduce the time bound to $n^{O(1/\varepsilon^{d-1})}$ and further generalize to objects, not necessarily fat, but whose projection to the first $d-1$ coordinates are fat and of roughly the same size, as essentially shown by Agarwal *et al.* [1] (they are interested in two-dimensional unit-height rectangles, which naturally occur in the map labeling application mentioned above).

The packing problem seems difficult if we do not impose any fatness constraints on the objects. It is open whether the case of arbitrary axis-aligned rectangles in the plane (or more generally, axis-aligned boxes in a fixed dimension) admits a polynomial-time algorithm with any constant approximation factor: Agarwal *et al.* [1] described a straightforward divide-and-conquer algorithm with $O(n\log^{d-1} n)$ running time and a $\log_2^{d-1} n$ approximation factor guarantee (their result was stated in the planar case, and Nielsen [26] and Khanna *et al.* [21] independently reported on similar claims as well); more recently, Berman *et al.* [7] showed how to achieve a $\log_b^{d-1} n$ approximation factor in $n^{O(b^{d-1})}$ time for an arbitrarily large constant b.

Here, we focus on the case where objects are fat but have possibly varying sizes (for example, arbitrary squares or bounded-aspect-ratio rectangles). Erlebach, Jansen, and Seidel [12] recently showed that a PTAS is possible, using dynamic programming with a shifted hierarchical subdivision strategy. For $d=2$, their algorithm requires $n^{O(1/\varepsilon^4)}$ time and space (the authors did not report running time for $d \geq 3$). We describe a similar PTAS that takes only $n^{O(1/\varepsilon^{d-1})}$ time and space. Our algorithm uses instead shifted quadtrees, which are more well-studied and, we believe, intuitively more appealing. Our algorithm contains an interesting application of known shifting lemmas from [8, 9] that were originally established in a different context.

The large space complexity of either dynamic-programming-based algorithm is undoubtedly a concern to practitioners. We describe another algorithm that takes $n^{O(1/\varepsilon^d)}$ time but only linear space. This algorithm uses an entirely different idea: namely, divide-and-conquer with geometric separators, specifically, those of Smith and Wormald [30]. Our extension of their separator theorem here could be of independent interest. Before, it was unclear that separator-based approaches could work when we have highly overlapping objects; in fact, Hunt *et al.* [18] (p. 243) believed otherwise, but our result dispels their belief.

The case of arbitrary-size fat objects studied here is important, because as pointed out by Hunt *et al.* [18] and Erlebach *et al.* [12], it not only generalizes the earlier PTAS result for the equal-size case obtained by Hochbaum and Maass [17], but at the same time generalizes previous PTAS results for planar graphs. Indeed, by the Koebe's theorem [27], every planar graph can be realized as the intersection graph of a set of disks (of possibly varying size) in the plane; more precisely, a graph is planar iff it is the "contact" graph of a set of interior-disjoint disks. So, Erlebach *et al.*'s and our PTASs generalize the known PTASs for independent sets in planar graphs by Lipton and Tarjan [23] and Baker [5]. One can draw analogies between our first algorithm with Baker's (which uses dynamic programming), and our second algorithm and Lipton and Tarjan's (which uses the original planar separator theorem).

Piercing. We also examine a geometric analogue of the hitting set problem [15]: choose the smallest number of points in $\mathbb{R}^d$ such that each of the n given objects contains at least one of the chosen points. The cardinality of the optimal solution will be denoted by $\text{pierce}(\mathcal{C})$.

The problem is basic and has applications in facility location. For example, given a collection of demand regions, we may want to minimize the number of facilities to be built at selected points so that all the demands are served. A demand region may consist of all points within a certain distance from a point site (a disk if the Euclidean metric used), and thus is usually fat. The regions may have different sizes if these sites have different weights (i.e., radii). A large body of work on this and the related "p-center" problem appeared in the computational geometry literature (see the references in [2]) but primarily addressed the case where we are to choose only a constant number p of points.

When a large number of chosen points is possible, the problem is NP-hard, even if the objects are axis-aligned unit squares or unit disks in the plane [14]. Therefore, we again investigate efficient approximation algorithms.

The piercing problem is related to the packing problem. Clearly, $\text{pierce}(\mathcal{C}) \geq \text{pack}(\mathcal{C})$. For intervals in one dimension, the two numbers are well-known to be equal. For fat objects in any fixed dimension, the two numbers are within a constant factor of each other (see Section 2), and for general axis-aligned boxes, they are within a factor of $\log^{d-1} n$ [26].

Many approximation results for packing carry over to piercing. For example, Hochbaum and Maass' method [17] yield PTAS for piercing unit squares or disks, and more generally near-equal-size fat objects in any fixed dimension; the running time is $n^{O(1/\varepsilon^d)}$. For general axis-aligned rectangles or boxes, there is again a straightforward $O(n \log^{d-1} n)$ algorithm with a $\log_2^{d-1} n$ approximation factor [26]. Some piercing results have no analogy, however; for example, the standard greedy algorithm for hitting set [10, 13, 29] imply a $\ln n$-factor approximation in polynomial time for arbitrary boxes (and in fact, more general objects) in any fixed dimension.

Using the separator approach, we obtain a PTAS for the piercing problem for fat objects of arbitrary size; it runs in $n^{O(1/\varepsilon^d)}$ time and requires linear space. To our knowledge, this is the first PTAS for this case; previously, only a constant-factor approximation result was known, with a simple algorithm described by Efrat *et al.* [11] (see Section 2). Our result thus extends Hochbaum and Maass' by enlarging the class of objects with PTASs.

2 Preliminaries

In the sequel, all boxes (including squares and hypercubes) are implicitly assumed to be axis-aligned.

Given an object S, define the *center* and *size* of S to be the center and side length of its smallest enclosing hypercube. There are a number of different definitions of fatness in the geometry literature (e.g., see the many references in [6, 11]). The one that we find the most appropriate for our purpose is the following:

Definition. A collection $\mathcal{C}$ of objects is *fat* if for any r and size-r box R, we can choose a constant number c of points such that every object that intersects R and has size at least r contains one of the chosen points.

A collection of closed hypercubes in $\mathbb{R}^d$ is fat according to this definition, with $c = 2^d$ (just choose the vertices of R). It is not difficult to see that a collection of balls or boxes with bounded aspect ratios satisfies the property as well, with a different constant c.

We will in addition make computational assumptions about $\mathcal{C}$, that certain reasonable operations can be accomplished in polynomial time, such as determining the center and size of an object, deciding if two objects intersect, and constructing the arrangement of the objects. This implies, in particular, that a polynomial-time algorithm exists for both the packing and piercing problem if $\text{pack}(\mathcal{C})$ or $\text{pierce}(\mathcal{C})$ is bounded by a constant (using linear space).

The following simply incremental algorithm, described by Efrat *et al.* [11] (a special case described by Marathe *et al.* [24] essentially used the same idea), yields a constant-factor approximation simultaneously for the packing and piercing problem if we are given a collection $\mathcal{C}$ of fat objects of different sizes: select the object S of smallest size, create a cluster consisting of all objects intersecting S (including S itself), remove all objects in the cluster, and repeat until no objects remain.

Let t be the number of clusters. For packing, we can simply return the t selected objects. For piercing, we can return a solution with ct points, because by our fatness definition, we can choose a c-point set to intersect all objects in a cluster. Since $t \leq \text{pack}(\mathcal{C}) \leq \text{pierce}(\mathcal{C}) \leq ct$, the approximation factor is at most c for both problems.

For example, for squares in the plane, this algorithm achieves an approximation factor 4. The question we tackle is whether this number can be reduced if we perform more work. What we will show is that the number can (in theory) be reduced to any constant arbitrarily to 1 in polynomial time for any collection of fat objects in any fixed dimension.

3 A Shifted-Quadtree Algorithm for Packing

The first algorithm we present for packing bears close resemblance with Erlebach *et al.*'s recently published algorithm [12], although ours is faster in terms of dependence on ε and uses a simpler recursive subdivision strategy—the quadtree. (Our algorithm was actually discovered independently of Erlebach *et al.*'s.)

Given a collection $\mathcal{A}$ of objects and a box R, let $\mathcal{A}_R$, $\mathcal{A}_{\overline{R}}$, and $\mathcal{A}_{|\partial R}$ denote the subcollection of objects in $\mathcal{A}$ that are inside R, are outside R, and intersect the boundary of R respectively. (In this paper, "inside/outside" always means "completely inside/outside.")

An *r-grid interval* refers to an interval of the form $[ri, r(i+1))$ for an integer i, and an *r-grid cell* in $\mathbb{R}^d$ refers to the Cartesian product of d r-grid intervals (a hypercube of size r). Hochbaum and Maass' algorithm [17] for equal-size objects use a single r-grid, but to deal with different-size objects, we have to consider r-grid cells for $r = 2^{-\ell}$ over all integers ℓ, called *quadtree cells* (the terminology is due to the hierarchical structure formed in the two-dimensional case). Our key definition is the following:

Definition. An object with center p and size r is *k-aligned* if it is inside a quadtree cell of size at most kr.

For the objects that are k-aligned for a constant k, we can actually solve the packing problem *exactly* in polynomial time by divide-and-conquer with dynamic programming.

Lemma 3.1 *If all objects in $\mathcal{C}$ are fat and k-aligned, then the packing problem can be solved in $n^{O(k^{d-1})}$ time and space.*

Proof: Let R be a quadtree cell of size r. Take an object S that intersects R. By the k-alignedness definition, we know that if S has size $< r/k$, then S is inside some quadtree cell of size $< r$, and is thus inside R. Therefore, the objects in $\mathcal{C}_{|\partial R}$ must have size $\geq r/k$, but by the fatness definition, since R's boundary can be covered by $2dk^{d-1}$ boxes of size r/k, any disjoint subcollection of $\mathcal{C}_{|\partial R}$ can have cardinality at most $K = 2cdk^{d-1}$—a constant, if k is constant. This is the main property that enables a polynomial-time algorithm, because we can simply "guess" which constant number of objects intersect the boundary in the optimal solution, and then apply divide-and-conquer to the objects inside.

More precisely, given quadtree cell R and a disjoint subcollection $\mathcal{B} \subseteq \mathcal{C}_{|\partial R}$, define pack$[R, \mathcal{B}]$ to be the maximum cardinality of a subcollection $\mathcal{A} \subseteq \mathcal{C}_R$ such that $\mathcal{A} \cup \mathcal{B}$ is disjoint. We describe a recursive formula to compute this value.

First take the smallest quadtree cell $\hat{R} \subseteq R$ that contains all the objects' centers in R (this idea of "shrinking" is standard, e.g., see [4]); we can compute $\hat{R}$ in linear time (if the real-RAM model is used, the support of operations like bitwise exclusive-or, floors, and logarithms is assumed). Next split $\hat{R}$ into 2^d quadtree subcells $\{R_i\}_{i=1}^{2^d}$.

We can relate the values pack$[R, \cdot]$ to the values of pack$[R_i, \cdot]$ over these subcells R_i as follows. Any subcollection $\mathcal{A} \subseteq \mathcal{C}_R$ can be decomposed as the union $\bigcup_i \mathcal{A}_i \cup \mathcal{B}'$, where $\mathcal{A}_i$ consists of the objects inside R_i, and $\mathcal{B}'$ consists of the objects that intersect the boundary of some R_i but not the boundary of R. (This is because by construction, every object inside R intersects $\hat{R}$, and thus, at least one of the R_i's.) Now, $\mathcal{A} \cup \mathcal{B}$ is disjoint iff for every i, $\mathcal{A}_i \cup \mathcal{B}' \cup \mathcal{B}$ is disjoint iff for every i, $\mathcal{A}_i \cup (\mathcal{B}' \cup \mathcal{B})_{|\partial R_i}$ is disjoint, assuming that $\mathcal{A}_i$ and $\mathcal{B}' \cup \mathcal{B}$ themselves are disjoint. The formula below can thus be verified:

$$\text{pack}[R, \mathcal{B}] \;=\; \max_{\mathcal{B}'} \left(\sum_{i=1}^{2^d} \text{pack}[R_i, (\mathcal{B}' \cup \mathcal{B})_{|\partial R_i}] \;+\; |\mathcal{B}'| \right),$$

where the maximum is over all subcollection $\mathcal{B}' \subseteq \bigcup_i \mathcal{C}_{|\partial R_i} \setminus \mathcal{C}_{|\partial R}$ such that $\mathcal{B}' \cup \mathcal{B}$ is disjoint.

Since $|\mathcal{B}| \leq K$ and $|\mathcal{B}'| \leq 2^d K$, the number of choices for $\mathcal{B}$ and $\mathcal{B}'$ in the above is bounded by $n^{O(K)}$. Since the number of centers in each R_i is strictly less than that in R (and the problem becomes trivial if no centers remain), the number of quadtree cells R generated by this recursion is $O(n)$. So by dynamic programming, we can evaluate the values pack$[R, \mathcal{B}]$ bottom-up, and thus solve the packing problem, in $n^{O(K)}$ time. $\square$

Hochbaum and Maass observed that a $(1+\varepsilon)$-approximation can be obtained simply by shifting the grid (or equivalently, the objects) by various vectors, then solving the exact problem on the objects that are aligned (in our terminology) for each vector, and returning the best solution found. We adopt the same strategy here. The key is the following lemma, a generalization of Lemma 3.3 in [9], originally stated in a different terminology concerning *centrality* of points (roughly speaking, an "aligned" object is the same as an object whose center is "central" in a grid cell of the appropriate size). By scaling, we may assume that all objects are inside $[0,1)^d$.

Lemma 3.2 *Fix an odd number $k > d$. Let $v^{(j)} = (j/k, \ldots, j/k) \in \mathbb{R}^d$. For any object S inside $[0,1)^d$, the shifted object $S + v^{(j)}$ is $(2k)$-aligned for all but at most d indices $j \in \{0, 1, \ldots, k\}$.*

Proof: Say S has center $p = (p_1, \ldots, p_d)$ and size r. If $r > 1/k$, then the lemma is trivial since $S + v^{(j)}$ is contained in $[0,2)^d$, which has size less than $2kr$. Otherwise, find a natural number ℓ such that $2^{-\ell} < 2kr \leq 2^{-\ell+1}$. Suppose $S + v^{(j)}$ is not $(2k)$-aligned. By definition, $S + v^{(j)}$—and thus, the hypercube with center $p + v^{(j)}$ and size r—cannot be inside any $2^{-\ell}$-grid cell. So, for some $i \in \{1, \ldots, d\}$, the interval with center $p_i + j/k$ and length r is not inside any $2^{-\ell}$-grid interval. Algebraically, this means that

$$(p_i + j/k + r/2) \bmod 2^{-\ell} \; < \; r.$$

(Recall the notation for real numbers x, y: $x \bmod y = x - \lfloor x/y \rfloor y$.) Multiplying both sides by $2^\ell k$, setting $z_i = 2^\ell k(p_i + r/2)$, and observing that $2^\ell kr \leq 1$, we have

$$(2^\ell j + z_i) \bmod k \; < \; 1, \quad \text{i.e.,} \quad 2^\ell j \equiv -\lfloor z_i \rfloor \pmod{k}.$$

For each i, there is at most one choice for j, since 2^ℓ and k are relatively prime. So, there can be at most d choices for j. □

Putting these observations together, we obtain a PTAS.

Theorem 3.3 *Given a collection $\mathcal{C}$ of n fat objects in $\mathbb{R}^d$, we can find a $(1+\varepsilon)$-factor approximation to the packing problem in $n^{O(1/\varepsilon^{d-1})}$ time and space.*

Proof: For each $j \in \{0, 1, \ldots, k-1\}$, apply Lemma 3.1 to find the largest disjoint subcollection $\mathcal{A}^{(j)}$ among the objects $S \subseteq \mathcal{C}$ with $S + v^{(j)}$ being $(2k)$-aligned. Return the solution $\mathcal{A}^{(j)}$ of the largest size.

To analyze the approximation factor of this algorithm, let $\mathcal{A}^*$ be the largest disjoint subcollection of $\mathcal{C}$. Then

$$\sum_{j=0}^{k-1} |\mathcal{A}^{(j)}| \; \geq \; \sum_{j=0}^{k-1} |\{S \in \mathcal{A}^* \; : \; S + v^{(j)} \text{ is } (2k)\text{-aligned}\}| \; \geq \; (k-d)|\mathcal{A}^*|,$$

by Lemma 3.2. So, for some j, we have $|\mathcal{A}^{(j)}| \geq (1 - d/k)|\mathcal{A}^*|$. Choosing k near d/ε establishes the result. □

Alternatively, we can just take one random vector $v \in [0,1)^d$ for the shift: it is not difficult to show that for a fixed object S, $S + v$ is $(2k)$-aligned with probability at least $1 - d/k$ (e.g., see [8] in terms of centrality), hence a randomized algorithm with expected approximation factor $1+\varepsilon$ follows.

Deterministic and randomized shifting of quadtrees have been used in approximate solutions for nearest neighbor queries [8, 9] (for which Lemma 3.2 was established), traveling salesman tours [3, 28], and other problems. Our example here is interesting in that we are not approximating distances but rather a combinatorial quantity.

A nice feature of the above algorithm is that it extends to the weighted packing problem (maximizing the sum of the weights of the chosen objects). In some applications, it is natural to give larger-size objects larger weights. However, not all algorithms can be adapted for the weighted problem (e.g., consider the constant-factor method in Section 2).

4 A Separator-Based Algorithm

Several obstacles prevent us from adapting the preceding algorithm to the piercing problem for arbitrary-size fat objects (unlike in the same-size case, where Hochbaum and Maass' grid method [17] adapts easily). First of all, the divide-and-conquer approach in Lemma 3.1 works for packing because with a subcollection of constant cardinality, we can characterize what any feasible solution looks like along the boundary of a quadtree cell; however, in the piercing problem, many points could still be used to intersect the objects along the boundary in the global optimal solution. Secondly, unlike in Theorem 3.3, taking a global shift and eliminating all the unaligned objects do not seem to guarantee a $(1+\varepsilon)$-factor approximation for piercing.

To deal with the first obstacle, a natural idea is simply to ignore the objects along the boundary, since they only affect the optimal value by at most a constant term (actually a sublinear term with the change suggested below), and we are settling for approximations anyways. In order to make this term negligible, we cannot use the naive split in Lemma 3.1 but need a more "balanced" split (balanced versions of quadtrees are well-studied, e.g., see [4]). To overcome the second obstacle, we have to use a different shift at each divide step, instead of a single global shift, so that the objects eliminated indeed do not greatly affect the optimal value.

As it turns out, the divide-and-conquer algorithm that results from applying the above ideas can more conveniently be expressed in terms of *separators*. Smith and Wormald [30] described a geometric separator theorem, which will be our starting point. The most basic version states:

Theorem 4.1 (Smith and Wormald's separator theorem) *Given a collection of n disjoint fat objects in $\mathbb{R}^d$, there exists a box R such that at most $2n/3$ objects are inside R, at most $2n/3$ objects are outside R, and at most $O(n^{1-1/d})$ objects intersect the boundary of R.*

Smith and Wormald in fact considered various forms of "fatness" and extensions to "disjointness" in the above theorem (for example, the theorem holds even if the objects are not disjoint as long as each point can lie in at most a constant number of objects).

To appreciate the power of this simple theorem (with a simple proof), take a planar graph G of n vertices and realize it as the contact graph of n disks in the plane by Koebe's theorem. Theorem 4.1 immediately implies that G can be decomposed into two components with a fraction of the vertices by the removal of $O(\sqrt{n})$ vertices—ignoring constant factors, this is Lipton and Tarjan's famous separator theorem for planar graphs [22]. As another special case, Theorem 4.1 also includes Miller *et al.*'s higher-dimensional analogue to the planar separator theorem, the so-called "sphere separator" theorem [25] (although the separator R is stated as a box above, other types of objects have also been considered in Smith and Wormald's paper).

We are unable to apply Theorem 4.1 directly to solve our problems because of two reasons: 1. our input collection of objects $\mathcal{C}$ is not disjoint but can be highly overlapping, and 2. the above separator bound depends on n and is not sensitive to the output size. (It was the first reason that led Hunt *et al.* [18] to believe that separator-based approaches are doomed.) It turns out that both difficulties can be simultaneously resolved by modifying the proof of the theorem.

To state our result in an abstract framework, consider a *measure* $\mu(\cdot)$ that maps a collection of objects to a nonnegative number and satisfies the following axioms:

(A1) If $\mathcal{A} \subseteq \mathcal{B}$, then $\mu(\mathcal{A}) \leq \mu(\mathcal{B})$.

(A2) $\mu(\mathcal{A} \cup \mathcal{B}) \leq \mu(\mathcal{A}) + \mu(\mathcal{B})$.

(A3) If no pair of objects in $\mathcal{A} \times \mathcal{B}$ intersects, then $\mu(\mathcal{A} \cup \mathcal{B}) = \mu(\mathcal{A}) + \mu(\mathcal{B})$.

(A4) Given any r and size-r box R, if every object in $\mathcal{A}$ intersects R and has size at least r, then $\mu(\mathcal{A}) \leq c$ for a constant c.

(A5) A constant-factor approximation to $\mu(\mathcal{A})$ can be computed in time $|\mathcal{A}|^{O(1)}$. If $\mu(\mathcal{A}) \leq b$, then $\mu(\mathcal{A})$ can be computed exactly in time $|\mathcal{A}|^{O(b)}$ and linear space.

If we are dealing with disjoint collections, then the cardinality of the collection is certainly a measure that obeys these axioms. In general, it is easy to see that both pack$(\cdot)$ and pierce$(\cdot)$ satisfy (A1)–(A3), and assuming fatness of the objects, (A4) as well. By the discussion in Section 2, the computational axiom (A5) also holds.

We apply Smith and Wormald's technique to obtain the following extension to Theorem 4.1. There are some minor changes to the proof in [30]: we use a 2^d-way split instead of a 2-way split purely for convenience, so that all the boxes encountered are perfect hypercubes (the same comment applies to the description of our algorithm in Section 3); we also prefer an expansion rather than a shift, which is chosen deterministically rather than randomly so as to avoid a probabilistic argument. We describe a simpler counting argument, by bounding the measure of "large objects" ($\mathcal{C}'$) and "small objects" ($\mathcal{C} \setminus \mathcal{C}'$) separately.

Theorem 4.2 *Given a measure μ satisfying* (A1)–(A4) *and a collection $\mathcal{C}$ of n objects in $\mathbb{R}^d$ with $\mu(\mathcal{C})$ sufficiently large, there exists a box R such that $\mu(\mathcal{C}_R), \mu(\mathcal{C}_{\overline{R}}) \geq \alpha\mu(\mathcal{C})$, and $\mu(\mathcal{C}_{|\partial R}) = O(\mu(\mathcal{C})^{1-1/d})$, where $\alpha > 0$ is some fixed constant. Moreover, if* (A5) *is satisfied, such a box can be found in polynomial time and linear space.*

Proof: Define the measure of a box R to be the measure of the subcollection of all objects in $\mathcal{C}$ with center in R. Consider the smallest hypercube R_0 with measure $\geq \frac{1}{2^d+1}\mu(\mathcal{C})$. Say R_0 has center p and size r. Every hypercube of size r has measure at most $\frac{1}{2^d+1}\mu(\mathcal{C}) + O(1)$ (otherwise we can shrink it slightly to get a better R_0).

Given $0 \leq t \leq 1$, let R_t be the hypercube with center p and size $(1+t)r$. Since R_t can be covered by 2^d hypercubes of size r, R_t has measure between $\frac{1}{2^d+1}\mu(\mathcal{C})$ and $\frac{2^d}{2^d+1}\mu(\mathcal{C}) + O(1)$. With α near $\frac{1}{2^d+1}$, it suffices to show that $\mu(\mathcal{C}_{|\partial R_t}) = O(\mu(\mathcal{C})^{1-1/d})$ for some choice of t.

Fix a parameter k to be chosen later. Let $\mathcal{C}'$ be the objects in $\mathcal{C}$ with size at least $r/2k$. By (A4), since R_t's boundary can be covered by $2d(4k)^{d-1}$ boxes of size $r/2k$, we have $\mu(\mathcal{C}'_{|\partial R_t}) \leq 2cd(4k)^{d-1}$

for any $t \leq 1$. On the other hand, no pair of objects in $(\mathcal{C} \setminus \mathcal{C}')_{|\partial R_s} \times (\mathcal{C} \setminus \mathcal{C}')_{|\partial R_t}$ intersects if $|s - t| \geq 1/k$, and by (A3),

$$\sum_{j=0}^{k-1} \mu((\mathcal{C} \setminus \mathcal{C}')_{|\partial R_{j/k}}) \;\leq\; \mu(\mathcal{C}).$$

So, for some $t = j/k$, we have $\mu((\mathcal{C} \setminus \mathcal{C}')_{|\partial R_t}) \leq \mu(\mathcal{C})/k$, implying that

$$\mu(\mathcal{C}_{|\partial R_t}) \;\leq\; \mu(\mathcal{C})/k \,+\, O(k^{d-1}).$$

Setting k near $\mu(\mathcal{C})^{1/d}$ establishes the separator bound.

As for the computational cost, there are $O(n^{2d})$ choices for R_0 (as it suffices to consider boxes whose coordinates are from the coordinates of the n center points) and k choices for t (namely, $t = 0, 1/k, 2/k, \ldots, (k-1)/k$), and a naive algorithm can try each combination and verify the bounds (a more efficient approach should be possible, though). Since we only have access to constant-factor approximations to $\mu(\mathcal{C})$, $\mu(\mathcal{C}_R)$, and $\mu(\mathcal{C}_{\overline{R}})$ by (A5), the value of α needs to be decreased by a constant factor. $\square$

Equipped with our separator theorem, we can immediately derive a PTAS for both packing and piercing fat objects, in a manner similar to how Lipton and Tarjan [23] derived their PTAS for maximum independent sets in planar graphs with their separator theorem. (It is a straightforward matter to output the approximately optimal disjoint subcollection or piercing point set in the algorithm below.)

Corollary 4.3 *Given a measure μ satisfying* (A1)–(A5) *and a collection $\mathcal{C}$ of n objects in $\mathbb{R}^d$, we can find a $(1+\varepsilon)$-factor approximation to $\mu(\mathcal{C})$ in $n^{O(1/\varepsilon^d)}$ time and $O(n)$ space.*

Proof: Fix a parameter b to be determined later. By (A5), we can tell whether $\mu(\mathcal{C}) \leq b$ and if so, compute $\mu(\mathcal{C})$ exactly in $n^{O(b)}$ time. If $\mu(\mathcal{C}) > b$, apply Theorem 4.2, compute $\mu(\mathcal{C}_R)$ and $\mu(\mathcal{C}_{\overline{R}})$ recursively, and return the sum.

Let us analyze the additive error $E(m)$ for this algorithm on an instance with $\mu(\mathcal{C}) = m$. Since $\mu(\mathcal{C}_R) + \mu(\mathcal{C}_{\overline{R}}) \;\leq\; \mu(\mathcal{C}) \;\leq\; \mu(\mathcal{C}_R) + \mu(\mathcal{C}_{\overline{R}}) + O(\mu(\mathcal{C})^{1-1/d})$, we have the recurrence

$$E(m) \;=\; \begin{cases} 0 & \text{if } m \leq b \\ E(m_1) + E(m_2) + O(m^{1-1/d}) & \text{otherwise} \end{cases},$$

where $m_1 + m_2 \leq m$ and $m_1, m_2 \geq \alpha m$. The solution to this recurrence is $E(m) = O(m/b^{1/d})$, so the approximation factor of the algorithm is $1 + O(1/b^{1/d})$. Setting b near $1/\varepsilon^d$ establishes the result. $\square$

Comparing with our quadtree algorithm, our separator algorithm is clearly more space-efficient. Unlike the quadtree algorithm, the separator algorithm can also handle certain extensions of the packing problem, for example, when objects may *slide*—a variant considered in the map-labeling context [20]—assuming that the region swept by sliding has roughly the same size as the object itself. However, the separator algorithm cannot handle the weighted packing problem. Erlebach *et al.* [12] have already adapted their algorithm to solve the vertex cover problem on intersection graphs of fat objects; it remains to explore whether our techniques could help in approximately solving other NP-hard problems on these graphs.

Finally, we remark that our study here is largely theoretical. Constant factors in the exponents are unoptimized and quite high, and much work remains in order to develop truely practical approximation algorithms for these problems.

References

[1] 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.

[2] P. K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surveys*, 30:412–458, 1998.

[3] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45:753–782, 1998.

[4] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *J. ACM*, 45:891–923, 1998.

[5] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41:153–180, 1994.

[6] M. de Berg, M. Katz, A. F. van der Stappen, and J. Vleugels. Realistic input models for geometric algorithms. In *Proc. 13th ACM Sympos. Comput. Geom.*, pages 294–303, 1997.

[7] P. Berman, B. DasGupta, S. Muthukrishnan, and S. Ramaswami. Improved approximation algorithms for rectangle tiling and packing. In *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 427–436, 2001.

[8] M. Bern. Approximate closest-point queries in high dimensions. *Inform. Process. Lett.*, 45:95–99, 1993.

[9] T. M. Chan. Approximate nearest neighbor queries revisited. *Discrete Comput. Geom.*, 20:359–373, 1998.

[10] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, Mass., 1990.

[11] A. Efrat, M. J. Katz, F. Nielsen, and M. Sharir. Dynamic data structures for fat objects and their applications. *Comput. Geom. Theory Appl.*, 15:215–227, 2000.

[12] T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric graphs. In *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 671–679, 2001.

[13] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45:634–652, 1998.

[14] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Inform. Process. Lett.*, 12:133–137, 1981.

[15] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, 1979.

[16] J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[17] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32:130–136, 1985.

[18] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms*, 26:238–274, 1998.

[19] H. Imai and T. Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *J. Algorithms*, 4:310–323, 1983.

[20] M. van Kreveld, T. Strijk, and A. Wolff. Point set labeling with sliding labels. In *Proc. 14th ACM Sympos. Comput. Geom.*, pages 337–346, 1998.

[21] S. Khanna, S. Muthukrishnan, and M. Paterson. Approximating rectangle tiling and packing. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 384–393, 1998.

[22] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Applied Math.*, 32:177–189, 1979.

[23] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9:615–627, 1980.

[24] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

[25] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *J. ACM*, 44:1–29, 1997.

[26] F. Nielsen. Fast stabbing of boxes in high dimensions. *Theoret. Comput. Sci.*, 246:53–72, 2000.

[27] J. Pach and P. K. Agarwal. *Combinatorial Geometry.* Wiley-Interscience, New York, 1995.

[28] S. B. Rao and W. D. Smith. Improved approximation schemes for geometrical graphs via 'spanners' and 'banyans'. In *Proc. 31st ACM Sympos. Theory Comput.*, pages 540–550, 1999.

[29] P. Slavík. A tight analysis of the greedy algorithm for set cover. In *Proc. 28th ACM Sympos. Theory Comput.*, pages 435–439, 1996.

[30] W. D. Smith and N. C. Wormald. Geometric separator theorems and applications. In *Proc. 39th IEEE Sympos. Found. Comput. Sci.*, pages 232–243, 1998.