Optimization and approximation on systems of geometric objects

van Leeuwen, E.J.

Chapter 10

Geometric Unique and Membership Coverage Problems

We present the first study on the approximability of geometric versions of the unique coverage problem and the minimum membership set cover problem. We prove that Unique Coverage (and thus Budgeted Low-Coverage as well) remains NP-hard on unit disks and give constant-factor approximation algorithms for both problems on unit disks. The results extend to unit squares. We then show that Budgeted Low-Coverage has an eptas on fat objects of bounded ply, but prove that without the bounded ply assumption, the problem is as hard to approximate as in its general setting.

We then consider the geometric version of Minimum Membership Set Cover. We prove that approximating the problem within ratio less than 2 is NP-hard on unit disks and unit squares, and give a 5-approximation algorithm on unit squares that runs in polynomial time if the optimal objective value is bounded by a constant.

10.1 Unique Coverage

We consider the complexity and approximability of Unique Coverage on unit disks and on unit squares. We first introduce some notation. Suppose that we are given a set $\mathcal{S}$ of geometric objects in the plane and a set of points $\mathcal{P} \subseteq \mathbb{R}^2$. Then for any $X \subseteq \mathcal{S}$, $cov(X)$ respectively $uc(X)$ denotes the set of points in $\mathcal{P}$ covered respectively covered uniquely by X. If S is a square in the plane, $uc_S(X)$ is defined as the set of points of $\mathcal{P}$ that lie inside S and are uniquely covered by X.

Theorem 10.1.1 *Unique Coverage on unit disks and unit squares is NP-hard.*

Proof: We reduce from Independent Set on planar graphs of maximum degree 3, which is known to be NP-hard [114]. The starting point of the construction is similar to one used to prove NP-hardness of Independent Set on unit disk graphs [67]. For a planar graph G of degree 3, create a rectilinear embedding of G. This is an embedding of G onto a unit grid, such that each vertex is mapped to a unique corner of the grid and each edge is mapped to a path in the grid, where all such paths are disjoint, except possibly at their ends.

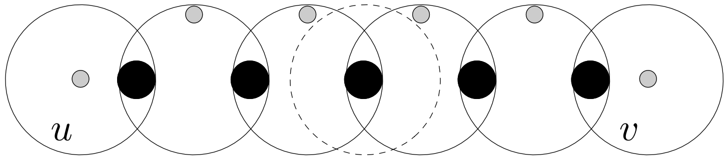

Figure 10.1: The edge gadget for some $(u, v) \in E(G)$, shown together with the vertex disks for u and v. If the path in the rectilinear embedding corresponding to (u, v) consists of more than one straight line segment, we can easily adapt the gadget.

Valiant [255, Theorem 2] has shown that such an embedding exists having area $O(|V(G)|^2)$. Now replace each vertex by a disk with a point in its center and each edge by a gadget, shown in Figure 10.1.

The set of (solid) disks connecting u and v for any edge $(u, v) \in E(G)$ must have even cardinality and is denoted by $\mathcal{D}^{(u,v)}$. A single (dashed) disk contains the middle black blob. The small gray points correspond to a single point. All disks of the construction, except the middle disk of each edge gadget (drawn dashed in Figure 10.1), contain a unique gray point. Each big black blob corresponds to a collection of $t = |V(G)| + \sum_{(u,v)\in E(G)} |\mathcal{D}^{(u,v)}|$ points.

We claim that in the constructed instance of Unique Coverage, a set of unit disks can uniquely cover at least

$$k' := k + \sum_{(u,v)\in E(G)} \left(\tfrac{1}{2}|\mathcal{D}^{(u,v)}| + t \cdot \left(|\mathcal{D}^{(u,v)}| + 1 \right) \right)$$

points if and only if G has an independent set of cardinality at least k. For the 'only if' part, note that any set of disks uniquely covering at least k' points must uniquely cover all black blobs. This can only be done using exactly $\frac{1}{2}|\mathcal{D}^{(u,v)}|$ disks of each edge gadget (which uniquely cover $\frac{1}{2}\sum_{(u,v)\in E(G)} |\mathcal{D}^{(u,v)}|$ gray points in total) and at least k vertex disks. By the construction of the edge gadgets, these vertex disks form an independent set, which indeed has cardinality at least k. We can now easily verify the 'if' part of the claim.

A similar argument can be given to demonstrate NP-hardness on unit squares. Moreover, the ply of the above construction is 3 and thus the NP-hardness extends to this case. $\square$

10.1.1 Approximation Algorithm on Unit Disks

Let $\mathcal{D}$ be a set of equally sized disks and $\mathcal{P}$ a set of points, both in $\mathbb{R}^2$. By scaling, we may assume that all disks in $\mathcal{D}$ have radius $1/2$. We aim to find a set $C \subseteq \mathcal{D}$ maximizing the number of uniquely covered points of $\mathcal{P}$. We apply the shifting technique (see Chapter 6) in a novel way.

Lemma 10.1.2 *Suppose for points in a square of size $\delta \times \delta$ ($0 < \delta \leq 1$), the unique coverage problem on unit disks has a polynomial time $1/c$-approximation algorithm. Then there is a polynomial time $(1/c) \cdot \delta^2/(1+\delta)^2$-approximation algorithm for the general unique coverage problem on unit disks.*

Proof: Let OPT be an optimal solution to the unique coverage problem on unit disks for some set of disks $\mathcal{D}$ and set of points $\mathcal{P}$. Pick two numbers a, b uniformly at random from $[0, 1+\delta)$. Consider the set of squares $\mathcal{S} = \{[a+h+h\delta, a+h+(h+1)\delta) \times [b+v+v\delta, b+v+(v+1)\delta) \mid h, v \in \mathbb{Z}\}$. Each square has size $\delta \times \delta$. As the squares in $\mathcal{S}$ have pairwise distance greater than 1, no unit disk can cover a point in more than one square of this set. Hence we may consider these squares to be 'independent'.

The probability that a point of $\mathcal{P}$ is in a square of $\mathcal{S}$ is $\delta^2/(1+\delta)^2$. Hence

$$\mathbf{E}\left[\sum_{S \in \mathcal{S}} |uc_S(OPT)|\right] = |uc(OPT)| \cdot \delta^2/(1+\delta)^2.$$

By assumption, we can find a $1/c$-approximation of the unique coverage problem on unit disks for each of the squares in $\mathcal{S}$ in polynomial time. Let $\mathsf{size}(S)$ denote the number of points uniquely covered by the solution of the algorithm for a particular square $S \in \mathcal{S}$. Then $\mathsf{size}(S) \geq 1/c \cdot |uc_S(OPT)|$. As we can assume that the solutions produced by the $1/c$-approximation algorithm contain only disks intersecting the square it was invoked on, it follows that

$$\begin{aligned} \mathbf{E}\left[\sum_{S \in \mathcal{S}} \mathsf{size}(S)\right] &\geq (1/c) \cdot \mathbf{E}\left[\sum_{S \in \mathcal{S}} |uc_S(OPT)|\right] \\ &= (1/c) \cdot |uc(OPT)| \cdot \delta^2/(1+\delta)^2. \end{aligned}$$

This approach can be derandomized. Choices of a, b for which the same set of points is in the squares of $\mathcal{S}$ give an approximation of the same quality. Hence it suffices to look at the $\mathrm{O}(|\mathcal{P}|^2)$ values of a, b for which a square boundary hits a point of $\mathcal{P}$ and thus we can consider all relevant values in polynomial time. The solution with the highest $\sum_{S \in \mathcal{S}} \mathsf{size}(S)$ is a $(1/c) \cdot \delta^2/(1+\delta)^2$-approximation of the optimum. □

The proof of the next theorem uses some ideas from Ambühl et al. [13].

Theorem 10.1.3 *Unique Coverage on unit disks has a $1/18$-approximation algorithm running in $\mathrm{O}(|\mathcal{P}|^3 |\mathcal{D}|^8)$ time.*

Proof: We prove that there is a polynomial-time $1/2$-approximation algorithm for Unique Coverage on unit disks for size $1/2 \times 1/2$ squares. Together with Lemma 10.1.2, this proves the theorem.

Consider a size $1/2 \times 1/2$ square S containing a set of points $\mathcal{P}_S$ and intersected by a set of disks $\mathcal{D}_S$ of radius $1/2$. We may assume that no disk covers all points of $\mathcal{P}_S$, for such a disk would constitute an optimal solution. Construct a mapping of disks in $\mathcal{D}_S$ to one of the four boundaries of S. The mapping assigns a disk to the boundary of S it overlaps most, breaking ties arbitrarily. If we can solve the unique coverage problem on unit disks optimally for both pairs of opposing boundaries of S, the best solution gives a 1/2-approximation of the optimum for Unique Coverage on unit disks for S.

Let OPT_S be an optimal solution to the unique coverage problem on $\mathcal{D}_S$ and $\mathcal{P}_S$. Consider two opposing boundaries b, b' of S (say b is the top boundary, b' the bottom boundary) and the sets of disks $\mathcal{D}^b, \mathcal{D}^{b'}$ assigned to them. Observe that for any disk $d \in \mathcal{D}^b$ the projection of $d \cap S$ onto b is equal to $d \cap b$, or d would overlap another boundary of S more and the mapping would assign it to this boundary. Hence any disk in $\mathcal{D}^b \cap OPT_S$ meets the lower envelope of $\mathcal{D}^b \cap OPT_S$ (for a set of disks, we say that a disk d from the set *meets* or *is on* the lower envelope of the set if part of that envelope is formed by the boundary of d), or any point of $\mathcal{P}_S$ it covers would already be covered by a disk on the lower envelope. Furthermore, for any disk d_2 on the lower envelope of $\mathcal{D}^b \cap OPT_S$, all points of $\mathcal{P}_S$ uniquely covered by d_2 are in $d_2 - d_1 - d_3$, where d_1 (d_3) is the disk lying directly to the left (right) of d_2 on the lower envelope of $\mathcal{D}^b \cap OPT_S$. The same properties hold for disks in $\mathcal{D}^{b'}$ and the upper envelope of $\mathcal{D}^{b'} \cap OPT_S$.

We now design a sweep-line algorithm with a vertical sweep-line, moving from left to right and stopping at each point of $\mathcal{P}_S$. Index the points of $\mathcal{P}_S$ in order of nondecreasing x-coordinate, with u_1 being the leftmost and u_m the rightmost point. By the above arguments, it suffices to consider three disks of both boundaries at each position of the sweep-line. Therefore when the sweep-line is at u_i, we say that a triple $\mathsf{d} = (d_1, d_2, d_3)$ of disks in $\mathcal{D}^b \cup \{b\}$ is *proper* if

- if $d_i = d_j$ for some $1 \le i < j \le 3$, then $d_i = d_j = b$;
- d_1, d_2, and d_3 appear in this order on the lower envelope of $d_1 \cup d_2 \cup d_3$;
- the intersection point of the sweep-line and d_2 lies on this envelope.

We use the 'disk' b to model the boundary, i.e. the situation when no disk intersects the sweep-line. If d is proper, then a proper triple d' is a *predecessor* of d if $\mathsf{d} = \mathsf{d}'$ or $\mathsf{d} = (d'_2, d'_3, d'_4)$ for some $d'_4 \in \mathcal{D}^b \cup \{b\}$. These notions can be defined analogously for triples $\mathsf{e} = (e_1, e_2, e_3)$ with respect to b'.

Now define for any point u_i and any pair of proper triples d and e a function h such that

$$h_i(\mathsf{d}, \mathsf{e}) = \begin{cases} \iota_i(\mathsf{d}, \mathsf{e}) & \text{if } i = 1; \\ \iota_i(\mathsf{d}, \mathsf{e}) + \max h_{i-1}(\mathsf{d}', \mathsf{e}') & \text{if } i > 1, \end{cases}$$

where $\iota_i(\mathsf{d}, \mathsf{e})$ is 1 if u_i is in $d_2 - d_1 - d_3$ or in $e_2 - e_1 - e_3$ but not in both, and 0 otherwise. The maximum is over all proper triples d' and e' that are predecessors of d and e respectively.

The maximum value of h_m over all proper triples for u_m is the optimal solution for this pair of opposing boundaries of S. Because h can be computed in $O(|\mathcal{P}|\,|\mathcal{D}|^8)$ time, the theorem follows from Lemma 10.1.2. □

10.1.2 Budgets and Satisfactions

The above algorithm extends easily to the case where points have associated profits and we aim to maximize the total profit of uniquely covered points. However for Budgeted Low-Coverage on unit disks, we need to change the approach. In the following, let $p_{\max}$ and $c_{\max}$ denote the maximum profit and the maximum cost respectively. We may assume that the budget is at least the maximum cost and that an optimal solution attains profit at least $s_1 \cdot p_{\max}$.

Theorem 10.1.4 *If $s_z > 0$ and $s_{z+1} = 0$ for some fixed z and the profits and satisfactions are integers, then Budgeted Low-Coverage on unit disks has a $1/18$-approximation algorithm running in $\mathrm{O}(|\mathcal{P}|^3 \cdot (s_1 \cdot p_{\max})^2 \cdot |\mathcal{D}|^{4z+4})$ time.*

Proof: Recall the proof of Lemma 10.1.2. Let $\mathcal{S}$ be the set of squares under consideration. We use similar ideas as in Theorem 10.1.3 to compute for both pairs of opposing boundaries of each square $S \in \mathcal{S}$ and for each $r = 0, \ldots, |\mathcal{P}| \cdot s_1 \cdot p_{\max}$ a set of disks such that the total satisfaction-modulated profit of covered points is at least r and the total cost is minimized.

Instead of triples, we consider tuples of $2z+1$ disks. Similar to Theorem 10.1.3, we can define the notions of proper and predecessor tuples. We then define for any $r = 0, \ldots, |\mathcal{P}| \cdot s_1 \cdot p_{\max}$, any point u_i, and any pair of proper tuples $\mathsf{d} = (d_1, \ldots, d_{2z+1})$, $\mathsf{e} = (e_1, \ldots, e_{2z+1})$ a function h such that $h_i(\mathsf{d}, \mathsf{e}, r)$ equals

$$\begin{cases} \min \Big\{ \vartheta(d_{z+1}, d'_{z+1}) \cdot c(d_{z+1}) \\ \qquad +\vartheta(e_{z+1}, e'_{z+1}) \cdot c(e_{z+1}) \\ \qquad +h_{i-1}\Big(\mathsf{d}', \mathsf{e}', r - s_{\gamma_i(\mathsf{d},\mathsf{e})} \cdot p(u_i)\Big) \Big\} & \text{if } i > 1; \\ \displaystyle\sum_{i=1}^{z+1} (c(d_i) + c(e_i)) & \text{if } i = 1 \text{ and } s_{\gamma_i(\mathsf{d},\mathsf{e})} \cdot p(u_i) \geq r; \\ \infty & \text{otherwise.} \end{cases}$$

The minimum is over all proper triples d' and e' that are predecessors of d and e respectively. The value of $\gamma_i(\mathsf{d}, \mathsf{e})$ is equal to the number of disks in $\{d_1, \ldots, d_{2z+1}, e_1, \ldots, e_{2z+1}\}$ containing u_i. The indicator function $\vartheta(\cdot, \cdot)$ is 1 if its parameters are distinct and 0 otherwise.

We are then interested in h_m for all $r = 0, \ldots, |\mathcal{P}| \cdot s_1 \cdot p_{\max}$. These values can clearly be computed in $\mathrm{O}(|\mathcal{P}|^2 \cdot s_1 \cdot p_{\max} \cdot |\mathcal{D}|^{4z+4})$ time.

For fixed S and r, let $\mathsf{cost}(S, r)$ denote the minimum cost over both pairs of opposing boundaries. Then $\max\{\sum_{S \in \mathcal{S}} r_S \mid \sum_{S \in \mathcal{S}} \mathsf{cost}(S, r_S) \leq B\}$ gives a $1/2$-approximation of $\sum_{S \in \mathcal{S}} sp_S(OPT)$, where $sp_S(OPT)$ is the satisfaction-modulated profit accrued by OPT in square S. This maximum is just an instance of Multiple-Choice Knapsack [60, 120], which we have to solve in a way that avoids having B in the running time. We first compute for each $r = 0, \ldots, |\mathcal{P}| \cdot s_1 \cdot p_{\max}$ a function

$$g(r) = \min \left\{ \textstyle\sum_{S \in \mathcal{S}} \mathsf{cost}(S, r_S) \mid \sum_{S \in \mathcal{S}} r_S = r \right\}$$

and then choose the largest r such that $g(r) \le B$. Using dynamic programming, this can be done in $O(|\mathcal{P}|^3 \cdot (s_1 \cdot p_{\max})^2)$ time, as the number of nonempty squares of $\mathcal{S}$ is at most $|\mathcal{P}|$. Applying the shifting technique as in Lemma 10.1.2 yields a 1/18-approximation in $O(|\mathcal{P}|^3 \cdot (s_1 \cdot p_{\max})^2 \cdot |\mathcal{D}|^{4z+4})$ time. □

Theorem 10.1.5 *If $s_z > 0$ and $s_{z+1} = 0$ for some fixed z, then for any $\epsilon > 0$, there is a $O(|\mathcal{P}|^7 \cdot |\mathcal{D}|^{4z+4}/\epsilon^4)$ time $(1-\epsilon)/18$-approximation algorithm for Budgeted Low-Coverage on unit disks.*

Proof: We extend the algorithm of Theorem 10.1.4 to deal with noninteger profits and satisfactions and to achieve polynomial running-time by applying scaling. Scale the profits by $\frac{|\mathcal{P}|}{\epsilon \cdot p_{\max}}$, i.e. define $p'(u) = \left\lfloor \frac{|\mathcal{P}| \cdot p(u)}{\epsilon \cdot p_{\max}} \right\rfloor$, and scale the satisfactions by $\frac{|\mathcal{P}|}{\epsilon \cdot s_1}$, i.e. define $s'_i = \left\lfloor \frac{|\mathcal{P}| \cdot s_i}{\epsilon \cdot s_1} \right\rfloor$. We first give an auxiliary inequality. For any $u \in \mathcal{P}$ and any i,

$$
\begin{aligned}
&s_i \cdot p(u) - \left(\frac{\epsilon \cdot s_1}{|\mathcal{P}|} \cdot \frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|}\right) \cdot s'_i \cdot p'(u) \\
&\quad= s_i \cdot p(u) - \frac{\epsilon \cdot s_1}{|\mathcal{P}|} \cdot \left\lfloor \frac{|\mathcal{P}| \cdot s_i}{\epsilon \cdot s_1} \right\rfloor \cdot \frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|} \cdot \left\lfloor \frac{|\mathcal{P}| \cdot p(u)}{\epsilon \cdot p_{\max}} \right\rfloor \\
&\quad\le s_i \cdot p(u) - \frac{\epsilon \cdot s_1}{|\mathcal{P}|} \cdot \left(\frac{|\mathcal{P}| \cdot s_i}{\epsilon \cdot s_1} - 1\right) \cdot \frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|} \cdot \left(\frac{|\mathcal{P}| \cdot p(u)}{\epsilon \cdot p_{\max}} - 1\right) \\
&\quad= s_i \cdot p(u) - \left(s_i - \frac{\epsilon \cdot s_1}{|\mathcal{P}|}\right) \cdot \left(p(u) - \frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|}\right) \\
&\quad\le 2 \cdot \frac{\epsilon \cdot s_1 \cdot p_{\max}}{|\mathcal{P}|}.
\end{aligned}
$$

Now for a set of disks $\mathcal{D}'$, let $sp(\mathcal{D}')$ denote the satisfaction-modulated profit achieved by $\mathcal{D}'$ under the original profits and satisfaction and let $s'p'(\mathcal{D}')$ be the satisfaction-modulated profit achieved by $\mathcal{D}'$ under the scaled profits and scaled satisfactions.

Apply the algorithm of Theorem 10.1.4 to obtain a set of disks C that is a 1/18-approximation of the optimum under the scaled profits and satisfactions. Let OPT denote a set of disks giving an optimum satisfaction-modulated profit under the budget and the original profits and satisfactions. Then

$$
\begin{aligned}
sp(C) &\ge \frac{\epsilon \cdot s_1}{|\mathcal{P}|} \cdot \frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|} \cdot s'p'(C) \\
&\ge \frac{\epsilon \cdot s_1}{|\mathcal{P}|} \cdot \frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|} \cdot \tfrac{1}{18} \cdot s'p'(OPT) \\
&\ge \tfrac{1}{18} \cdot \left(sp(OPT) - 2 \cdot \frac{\epsilon \cdot s_1 \cdot p_{\max}}{|\mathcal{P}|} \cdot |\mathcal{P}|\right) \\
&\ge \tfrac{1}{18} \cdot (sp(OPT) - 2 \cdot \epsilon \cdot sp(OPT)).
\end{aligned}
$$

Note that $s'_1, p'_{\max} \le |\mathcal{P}|/\epsilon$. The theorem follows from Theorem 10.1.4. □

If the satisfactions are different for each point, but all still nonincreasing, a similar algorithm may be used. It remains an interesting open problem if Budgeted Low-Coverage on unit disks is approximable for arbitrary z and/or satisfactions that are not nonincreasing.

10.1.3 Approximation Algorithm on Unit Squares

As with the geometric set cover problem, it seems that Unique Coverage is easier to approximate on unit squares than on unit disks. On unit disks, we had to consider instances on size $1/2 \times 1/2$ squares to be able to restrict the attention to the upper and lower envelopes. On unit squares however, this is no longer necessary. We use some ideas from Mihalák [209].

Theorem 10.1.6 *If $s_z > 0$ and $s_{z+1} = 0$ for some fixed z, then for any $\epsilon > 0$, there is a $\mathrm{O}(|\mathcal{P}|^7 \cdot |\mathcal{D}|^{4z+4}/\epsilon^4)$-time $(1-\epsilon)/2$-approximation algorithm for Budgeted Low-Coverage on unit squares.*

Proof: First assume that the profits and satisfactions are integer. Partition the plane into horizontal slabs of height 1. This induces a partition of the points of $\mathcal{P}$ as well. We claim that for any slab S and for each $r = 0, \ldots, |\mathcal{P}| \cdot s_1 \cdot p_{\max}$, we can find a set of squares such that the total satisfaction-modulated profit of covered points in S is at least r and the total cost is minimized.

To see this, consider a slab S and the set of points $\mathcal{P}_S$ contained in it. Any square covering a point of $\mathcal{P}_S$ must intersect either the top or the bottom boundary of S. Let $\mathcal{D}^t$ denote the set of squares intersecting the top boundary of S and $\mathcal{D}^b$ the set of squares intersecting the bottom boundary. Observe that in any optimal solution OPT, any square in $OPT \cap \mathcal{D}^t$ must intersect the lower envelope of $OPT \cap D^t$. Moreover, all points yielding a profit that are covered by a square $d_{z+1} \in OPT \cap \mathcal{D}^t$ are in $\bigcup_{i \in \{2,\ldots,2z\}} d_i - d_1 - d_{2z+1}$, where $d_1, \ldots, d_z$ and $d_{z+2}, \ldots, d_{2z+1}$ are the squares respectively to the left and to the right of d_{z+1} on the lower envelope of $OPT \cap \mathcal{D}^t$. Similar observations can be made about $C \cap \mathcal{D}^b$ and its upper envelope. Then we can just apply the algorithm of Theorem 10.1.3 and Theorem 10.1.4. This takes $\mathrm{O}(|\mathcal{P}|^2 \cdot s_1 \cdot p_{\max} \cdot |\mathcal{D}|^{4z+4})$ time.

Now let $\mathcal{S}^1$ denote the set of slabs whose bottom boundary is $y = i$ for some even integer i and let $\mathcal{S}^2$ denote the set of remaining slabs. Use the Multiple-Choice Knapsack algorithm of Theorem 10.1.4 to compute for both $\mathcal{S}^1$ and $\mathcal{S}^2$ a set of squares such that the total satisfaction-modulated profit of covered points is maximal and the total cost is at most B. The most profitable of these two solutions then gives a 1/2-approximation algorithm. This takes $\mathrm{O}(|\mathcal{P}|^3 \cdot (s_1 \cdot p_{\max})^2)$ time, as the number of nonempty slabs is at most $|\mathcal{P}|$.

Finally, we apply scaling as in Theorem 10.1.5 to deal with noninteger profits and satisfactions and to obtain a polynomial running time. $\square$

10.2 Unique Coverage on Disks of Bounded Ply

If the disks have arbitrary size, but have bounded ply, we can improve on the algorithm of the previous section and give an eptas for Unique Coverage on disks. Recall that the *ply* of a set of disks is the maximum over all points in the plane of the number of disks strictly containing this point. To obtain the eptas, we apply the shifting technique, which we used before in the case of disks of bounded ply (see Chapter 7). However, the complexity of the budgeted unique coverage problem on disks forces major changes in this approach. In particular, it is nontrivial to enforce the global budget constraint; we handle this by creating dynamic programming tables that are additionally indexed by profit values and contain the cheapest cost for achieving a certain profit in a given square. The best choice of profit values for disjoint squares can then be addressed as a multiple-choice knapsack problem. Furthermore, relating the profit of the algorithm's solution to that of a modified optimal solution (Lemma 10.2.5) is significantly more difficult than for the problems studied in Chapter 7. In the proofs below we focus on those aspects that are different from the approach in Chapter 7.

The setup of the algorithm is as follows. Let $\mathcal{D}$ be a set of disks and $\mathcal{P}$ a set of points. By scaling, we may assume that all disks have radius at least 1/2. Partition the disks into levels, where a disk with radius r has level $j \in \mathbb{Z}_{\geq 0}$ if $2^{j-1} \leq r < 2^j$. Then we can define $\mathcal{D}_{=j}$ as the set of disks having level exactly j, $\mathcal{D}_{\geq j}$ as the set of disks having level at least j, etc. We use l to denote the level of the largest disk.

For each level j, partition the plane using a grid induced by horizontal lines $y = hk2^j$ and vertical lines $x = vk2^j$ $(h, v \in \mathbb{Z})$, where $k \geq 5$ is an odd integer determined later. The squares of this partition for level j are called *j-squares*. Any j-square is contained in precisely one $j+1$-square, while each $j+1$-square contains exactly four j-squares, denoted $S_1, \ldots, S_4$ and called *siblings*. For a j-square S, let $\mathcal{D}^S$ denote the set of disks intersecting S and let $\mathcal{D}^{\mathrm{b}(S)}$ denote the set of disks intersecting the boundary of S. As a shorthand, let $\mathcal{D}^{\mathrm{i}(S)} = \mathcal{D}^S - \mathcal{D}^{\mathrm{b}(S)}$ be the set of disks fully contained inside S and $\mathcal{D}^{+(S)} = \bigcup_{i=1}^4 \mathcal{D}^{\mathrm{b}(S_i)} - \mathcal{D}^{\mathrm{b}(S)}$ be the set of disks intersecting the boundary of some S_i, but not the boundary of S. Combinations such as $\mathcal{D}^{\mathrm{b}(S)}_{>j}$ are self-explanatory. Let $\mathcal{P}_S$ denote the set of points contained in a j-square S and let $j(S)$ denote the level of a square S.

We prove the following auxiliary result.

Theorem 10.2.1 *Let $\mathcal{D}$ be a set of disks of ply γ, $k \geq 5$ an odd integer, and OPT a subset of $\mathcal{D}$ such that $p(uc(OPT))$ is maximum under $c(OPT) \leq B$. Then in time $\mathrm{O}((k^2|\mathcal{D}| + |\mathcal{P}|) \cdot k\,|\mathcal{D}| \cdot 2^{32k\gamma/\pi}(|\mathcal{P}| \cdot p_{\max})^2)$, we can find a set $\mathcal{D}' \subseteq \mathcal{D}$ such that $c(\mathcal{D}') \leq B$ and $p(uc(\mathcal{D}')) \geq p(uc(\bigcup_S OPT^{\mathrm{i}(S)}_{=j(S)}))$.*

To prove this theorem, we apply dynamic programming on the squares. Define for each j-square S, each set $W \subseteq \mathcal{D}^{\mathrm{b}(S)}_{>j}$, and each $r \in \{0, \ldots, |\mathcal{P}| \cdot p_{\max}\}$ the function $\mathsf{cost}(S,W,r)$ as

$$\mathsf{cost}(S,W,r)=\begin{cases}\min\left\{c(T)\,\middle|\,p(uc_S(T\cup W))\geq r;\ T\subseteq \mathcal{D}^{\mathrm{i}(S)}_{\geq j}\right\} & \text{if } j=0;\\ \min\Big\{c(U)+\sum_{i=1}^{4}\mathsf{cost}\big(S_i,(U\cup W)^{\mathrm{b}(S_i)},r_i\big)\,\Big| & \\ \qquad\qquad \sum_{i=1}^{4} r_i=r;\ U\subseteq \mathcal{D}^{+(S)}_{>j-1}\Big\} & \text{if } j>0.\end{cases}$$

Here the minimum over an empty set is ∞. Let $\mathsf{sol}(S,W,r)$ denote the subset of $\mathcal{D}$ attaining $\mathsf{cost}(S,W,r)$ if $\mathsf{cost}(S,W,r)\neq\infty$, or $\emptyset$ otherwise. Note that we would actually only need to define $\mathsf{cost}(S,W,r)$ and $\mathsf{sol}(S,W,r)$ for subsets W of $\mathcal{D}^{\mathrm{b}(S)}_{>j}\cap\bigcup_{S'\supset S}\mathcal{D}^{\mathrm{i}(S')}_{=j(S')}$. This will turn up in the analysis of the approximation ratio of the algorithm, but it is not very important in the analysis of the (worst-case) running time.

10.2.1 Properties of the cost- and sol-Functions

We start by proving the function cost is indeed close to the optimum. Define

$$up(S)=\bigcup\nolimits_{S'\supset S} OPT^{\mathrm{i}(S')}_{=j(S')}\qquad\text{and}\qquad down(S)=\bigcup\nolimits_{S'\subseteq S} OPT^{\mathrm{i}(S')}_{=j(S')}.$$

We give some properties of *up* and *down* that will be auxiliary to later lemmas.

Proposition 10.2.2 *For any j-square S and any child square S_i of S,*

$$\left(OPT^{+(S)}_{>j-1}\cup(up(S))^{\mathrm{b}(S)}\right)^{\mathrm{b}(S_i)}=(up(S_i))^{\mathrm{b}(S_i)}.$$

Proof: Observe that

$$\begin{aligned}&\left(OPT^{+(S)}_{>j-1}\cup(up(S))^{\mathrm{b}(S)}\right)^{\mathrm{b}(S_i)}\\ &\qquad=\left(OPT^{+(S)}_{>j-1}\cup\left(\bigcup\nolimits_{S'\supset S}OPT^{\mathrm{i}(S')}_{=j(S')}\right)^{\mathrm{b}(S)}\right)^{\mathrm{b}(S_i)}\\ &\qquad=\left(OPT^{+(S)}_{=j}\cup\bigcup\nolimits_{S'\supset S}OPT^{\mathrm{i}(S')}_{=j(S')}\right)^{\mathrm{b}(S_i)}\\ &\qquad=\left(\bigcup\nolimits_{S_i'\supset S_i}OPT^{\mathrm{i}(S_i')}_{=j(S_i')}\right)^{\mathrm{b}(S_i)}\\ &\qquad=(up(S_i))^{\mathrm{b}(S_i)}.\end{aligned}$$

The proposition follows. $\square$

Corollary 10.2.3 *It holds that $\bigcup_{i=1}^{4}(up(S_i))^{\mathrm{b}(S_i)}=(up(S))^{\mathrm{b}(S)}\cup OPT^{+(S)}_{>j-1}$ for any j-square S.*

Proposition 10.2.4 *For any j-square S,*

$$OPT^{+(S)}_{>j-1} \cup \bigcup_{i=1}^{4} \left(OPT^{\mathrm{i}(S_i)}_{>j-1} \cup down(S_i) \right) = OPT^{\mathrm{i}(S)}_{>j} \cup down(S).$$

Proof: Observe that

$$\begin{aligned} & OPT^{+(S)}_{>j-1} \cup \bigcup_{i=1}^{4} \left(OPT^{\mathrm{i}(S_i)}_{>j-1} \cup down(S_i) \right) \\ &\quad = \left(OPT^{+(S)}_{>j} \cup \bigcup_{i=1}^{4} OPT^{\mathrm{i}(S_i)}_{>j} \right) \cup \bigcup_{i=1}^{4} down(S_i) \\ &\qquad \cup \left(OPT^{+(S)}_{=j} \cup \bigcup_{i=1}^{4} OPT^{\mathrm{i}(S_i)}_{=j} \right) \\ &\quad = OPT^{\mathrm{i}(S)}_{>j} \cup \bigcup_{i=1}^{4} down(S_i) \cup OPT^{\mathrm{i}(S)}_{=j} \\ &\quad = OPT^{\mathrm{i}(S)}_{>j} \cup down(S). \end{aligned}$$

The proposition follows. $\square$

We are now ready to prove that cost is close to optimal.

Lemma 10.2.5 *It holds that*

$$\max \left\{ \sum_{S;\, j(S)=l} r_S \;\middle|\; \sum_{S;\, j(S)=l} \mathsf{cost}(S, \emptyset, r_S) \le B \right\} \ge p\left(uc\left(\bigcup_S OPT^{\mathrm{i}(S)}_{=j(S)} \right) \right).$$

Proof: To prove the lemma, we claim that for any j-square S

$$\begin{aligned} & \mathsf{cost}\left(S, (up(S))^{\mathrm{b}(S)}, p\left(uc_S\left((up(S))^{\mathrm{b}(S)} \cup OPT^{\mathrm{i}(S)}_{>j} \cup down(S) \right) \right) \right) \\ &\qquad \le \quad c\left(OPT^{\mathrm{i}(S)}_{>j} \right) + \sum_{S' \subseteq S} c\left(OPT^{\mathrm{i}(S')}_{=j(S')} \right). \end{aligned}$$

The intuition behind this formula is that if we consider the set W the optimum uses and the profit r attained by the optimum for S, the cost attained by $\mathsf{cost}(S, W, r)$ is at most the cost needed by the optimum. We prove it inductively on j. For $j = 0$, it is easily verified. Consider some $j > 0$ and assume inductively that the above statement holds for any $j' < j$. Observe that

$$\bigcup_{i=1}^{4} uc_{S_i}\left((up(S_i))^{\mathrm{b}(S_i)} \right) = uc_S\left(\bigcup_{i=1}^{4} (up(S_i))^{\mathrm{b}(S_i)} \right).$$

As $\mathcal{P}_{S_1}, \ldots, \mathcal{P}_{S_4}$ are pairwise disjoint sets, it follows from Corollary 10.2.3 and Proposition 10.2.4 that

$$\begin{aligned} & \sum_{i=1}^{4} p\left(uc_{S_i}\left((up(S_i))^{\mathrm{b}(S_i)} \cup OPT^{\mathrm{i}(S_i)}_{>j-1} \cup down(S_i) \right) \right) \\ &\quad = p\left(uc_S\left(\bigcup_{i=1}^{4} \left((up(S_i))^{\mathrm{b}(S_i)} \cup OPT^{\mathrm{i}(S_i)}_{>j-1} \cup down(S_i) \right) \right) \right) \\ &\quad = p\left(uc_S\left((up(S))^{\mathrm{b}(S)} \cup OPT^{+(S)}_{>j-1} \cup \bigcup_{i=1}^{4} \left(OPT^{\mathrm{i}(S_i)}_{>j-1} \cup down(S_i) \right) \right) \right) \\ &\quad = p\left(uc_S\left((up(S))^{\mathrm{b}(S)} \cup OPT^{\mathrm{i}(S)}_{>j} \cup down(S) \right) \right). \end{aligned}$$

Then by induction, the definition of cost, and Proposition 10.2.2,

$$\begin{aligned}
&\mathsf{cost}\left(S, (up(S))^{\mathrm{b}(S)}, p\left(uc_S\left((up(S))^{\mathrm{b}(S)} \cup OPT^{\mathrm{i}(S)}_{>j} \cup down(S)\right)\right)\right) \\
&\quad\le\ c\Big(OPT^{+(S)}_{>j-1}\Big) \\
&\qquad + \sum_{i=1}^{4} \mathsf{cost}\Bigg(S_i, \left(OPT^{+(S)}_{>j-1} \cup (up(S))^{\mathrm{b}(S)}\right)^{\mathrm{b}(S_i)}, \\
&\qquad\qquad p\Big(uc_{S_i}\left((up(S_i))^{\mathrm{b}(S_i)} \cup OPT^{\mathrm{i}(S_i)}_{>j-1} \cup down(S_i)\right)\Big)\Bigg) \\
&\quad=\ c\Big(OPT^{+(S)}_{>j-1}\Big) \\
&\qquad + \sum_{i=1}^{4} \mathsf{cost}\Bigg(S_i, (up(S_i))^{\mathrm{b}(S_i)}, \\
&\qquad\qquad p\Big(uc_{S_i}\left((up(S_i))^{\mathrm{b}(S_i)} \cup OPT^{\mathrm{i}(S_i)}_{>j-1} \cup down(S_i)\right)\Big)\Bigg) \\
&\quad\le\ c\left(OPT^{+(S)}_{>j-1}\right) + \sum_{i=1}^{4}\left(c\left(OPT^{\mathrm{i}(S_i)}_{>j-1}\right) + \textstyle\sum_{S_i' \subseteq S_i} c\left(OPT^{\mathrm{i}(S_i')}_{=j(S_i')}\right)\right) \\
&\quad=\ c\left(OPT^{\mathrm{i}(S)}_{>j}\right) + \sum_{S' \subseteq S} c\left(OPT^{\mathrm{i}(S')}_{=j(S')}\right),
\end{aligned}$$

proving the claim. Because l is the level of the largest disk, $up(S) = \emptyset$ and $OPT^{\mathrm{i}(S)}_{>l} = \emptyset$ for any l-square S. Hence

$$\begin{aligned}
\sum_{S;\, j(S)=l} \mathsf{cost}\left(S, \emptyset, p\left(uc_S\left(down(S)\right)\right)\right) &\le \sum_{S;\, j(S)=l}\ \sum_{S' \subseteq S} c\left(OPT^{\mathrm{i}(S')}_{=j(S')}\right) \\
&\le c(OPT) \\
&\le B
\end{aligned}$$

and thus

$$\begin{aligned}
\max\left\{ \sum_{S;\, j(S)=l} r_S \,\middle|\, \sum_{S;\, j(S)=l} \mathsf{cost}(S, \emptyset, r_S) \le B \right\} \\
\ge \sum_{S;\, j(S)=l} p\left(uc_S\left(down(S)\right)\right) \\
= p\left(uc\left(\textstyle\bigcup_S OPT^{\mathrm{i}(S)}_{=j(S)}\right)\right).
\end{aligned}$$

This proves the lemma. □

It follows immediately that for any $\{r_S\}_{S;\, j(S)=l}$ attaining the maximum of Lemma 10.2.5, $\bigcup_{S;\, j(S)=l} \mathsf{sol}(S, \emptyset, r_S)$ is a set of disks of cost at most B for which the total profit of the points uniquely covered by this set is at least $p\left(uc\left(\bigcup_S OPT^{\mathrm{i}(S)}_{=j(S)}\right)\right)$, as requested.

10.2.2 Computing the cost- and sol-Functions

We say that a j-square S is *nonempty* if it contains a level j disk and *empty* otherwise. A j-square S is said to be *relevant* if one of its three siblings is nonempty, there is a nonempty $S' \supseteq S$ of level at most $j + \lceil \log k \rceil$, or $j = 0$ and S contains at least one point of $\mathcal{P}$. This implies that any nonempty square is relevant and that there are at most $\mathrm{O}(k^2\,|\mathcal{D}| + |\mathcal{P}|)$ relevant squares. Note that the definition of relevant used here is slightly different from the definition used in Chapter 7.

A relevant square S is a *relevant child* of another relevant square S' if $S \subset S'$ and there is no relevant square S'' with $S \subset S'' \subset S'$. If S is a relevant child of S', S' is a *relevant parent* of S. We show that to compute cost it is sufficient to consider only relevant squares.

Lemma 10.2.6 *For each relevant level* 0 *square* S*, all* cost- *and* sol-*values can be computed in time* $\mathrm{O}(k|\mathcal{D}^S|\,|\mathcal{P}|^2\,p_{\max}\,2^{32k\gamma/\pi})$.

Proof: Since $|\mathcal{D}^{\mathrm{b}(S)}_{>0}| \leq 16k\gamma/\pi$ by Lemma 7.1.1, enumerating all $W \subseteq \mathcal{D}^{\mathrm{b}(S)}_{>0}$ takes $\mathrm{O}(2^{16k\gamma/\pi})$ time.

We show how to compute $\min\{c(T) \,|\, p(uc_S(T \cup W)) \geq r;\ T \subseteq \mathcal{D}^{\mathrm{i}(S)}_{\geq 0}\}$ for a particular set $W \subseteq \mathcal{D}^{\mathrm{b}(S)}_{>0}$ and each $r = 0, \ldots, |\mathcal{P}| \cdot p_{\max}$. Partition S into k vertical slabs of width exactly one and assign a point of $\mathcal{P}$ to a slab if the point is contained in the slab (w.l.o.g. no point lies on a slab boundary). Observe that any disk of $\mathcal{D}^{\mathrm{i}(S)}_{\geq 0}$ covering a point in a certain slab must intersect the left or the right boundary of this slab. Now order the slabs from left to right. Define for each slab $i = 1, \ldots, k$, each subset X of the set of disks intersecting the right boundary of slab i, and each $r = 0, \ldots, |\mathcal{P}| \cdot p_{\max}$ a function $a(i, X, r)$ as

$$\begin{aligned}\min\big\{\, & c(X) + a(i-1, Y, r - p(uc_i(X \cup Y \cup W))) \,\big|\\ & Y \subseteq \text{disks intersecting left boundary of } i, r \geq p(uc_i(X \cup Y \cup W))\big\}.\end{aligned}$$

Here $uc_i(X \cup Y \cup W)$ is the set of points in slab i uniquely covered by $X \cup Y \cup W$. We set $a(0, \emptyset, 0) = 0$ and $a(0, \emptyset, r) = \infty$ for each $r \neq 0$ (as we only consider disks in $\mathcal{D}^{\mathrm{i}(S)}$, no disk intersects the left boundary of the first slab). Observe that the minima we are looking for are in $a(k, \emptyset, r)$. Furthermore, we can compute them in $\mathrm{O}(k|\mathcal{D}^S|\,|\mathcal{P}|^2\,p_{\max}\,2^{16k\gamma/\pi})$ time, as computing $p(uc_i(X \cup Y \cup W))$ takes $\mathrm{O}((|X| + |Y| + |W|) \cdot |\mathcal{P}|)$ time and a vertical line in S intersects at most $8k\gamma/\pi$ disks of $\mathcal{D}^{\mathrm{i}(S)}$ (see Lemma 7.1.1). $\square$

To compute cost and sol for j-squares with $j > 0$, we require the following auxiliary proposition. The problem described in the proposition statement is an instance of Multiple-Choice Knapsack and may be solved using a similar method as the one in Theorem 10.1.4.

Proposition 10.2.7 *For any j-square S, given $W \subseteq \mathcal{D}^{\mathrm{b}(S)}_{>j}$ and $U \subseteq \mathcal{D}^{+(S)}_{>j-1}$, we can compute*

$$\min\left\{\sum_{i=1}^{4}\mathsf{cost}(S_i,(U\cup W)^{\mathrm{b}(S_i)},r_i) \mid \sum_{i=1}^{4} r_i = r\right\}$$

for all $r = 0,\ldots,|\mathcal{P}|\cdot p_{\max}$ in $\mathrm{O}((|\mathcal{P}|\cdot p_{\max})^2)$ time.

Proof: We compute for all $r' = 0,..,|P|\cdot p_{\max}$ the values of

$$g(1,r') = \mathsf{cost}(S_1,(U\cup W)^{\mathrm{b}(S_1)},r')$$

and, for all $i = 2,3,4$, the values of

$$g(i,r') = \min\{g(i-1,r'-r'') + \mathsf{cost}(S_i,(U\cup W)^{\mathrm{b}(S_i)},r'') \mid r'' = 0,\ldots,r'\}.$$

This takes time $\mathrm{O}((|P|\cdot p_{\max})^2)$. The values we need are the values $g(4,r)$ for $r = 0,\ldots,|\mathcal{P}|\cdot p_{\max}$. □

Lemma 10.2.8 *For each relevant j-square S ($j > 0$) which has relevant $(j-1)$-square children, all* cost- *and* sol-*values for S can be computed in time $\mathrm{O}(2^{32k\gamma/\pi}(|\mathcal{P}|\cdot p_{\max})^2)$.*

Proof: This follows from $|\mathcal{D}^{\mathrm{b}(S)}_{>j}|, |\mathcal{D}^{+(S)}_{>j-1}| \leq 16k\gamma/\pi$ (see Lemma 7.1.1) and Proposition 10.2.7. □

Lemma 10.2.9 *For each relevant j-square S ($j > 0$) with no relevant level $j-1$ children, in $\mathrm{O}((|\mathcal{D}|+|\mathcal{P}|)\cdot 2^{32\gamma/\pi}(|\mathcal{P}|\cdot p_{\max})^2)$ time all* cost- *and* sol-*values can be computed.*

Proof: It is easy to show from Lemma 7.2.6 that

$$\mathsf{cost}(S,W,r) =$$

$$\min\left\{\sum_{\text{relevant child } S' \text{ of } S}\mathsf{cost}(S',W^{b(S')},r_{S'}) \;\middle|\; \sum_{\text{relevant child } S' \text{ of } S} r_{S'} = r\right\}$$

for any $W \subseteq \mathcal{D}^{\mathrm{b}(S)}_{>j}$ and any $r = 0,\ldots,|\mathcal{P}|\cdot p_{\max}$. Furthermore, $\mathcal{D}^{\mathrm{b}(S)}_{>j} = \mathcal{D}^{\mathrm{b}(S)}_{\geq j+\lceil\log k\rceil}$ and thus $\left|\mathcal{D}^{\mathrm{b}(S)}_{>j}\right| = \left|\mathcal{D}^{\mathrm{b}(S)}_{\geq j+\lceil\log k\rceil}\right| \leq 32\gamma/\pi$. For a fixed $W \subseteq \mathcal{D}^{\mathrm{b}(S)}_{>j}$, we follow the approach of Proposition 10.2.7 and extend it to deal with all relevant children of S. The increase in time complexity is linear in the number of relevant children of S. As a relevant square has $\mathrm{O}(|\mathcal{D}|+|\mathcal{P}|)$ relevant children, the time bound follows. □

Proof of Theorem 10.2.1: Combining Lemmas 10.2.6, 10.2.8, and 10.2.9, we can compute the cost- and sol-values for all relevant squares in $\mathrm{O}((k^2|\mathcal{D}|+|\mathcal{P}|)\cdot k\,|\mathcal{D}|\cdot 2^{32k\gamma/\pi}(|\mathcal{P}|\cdot p_{\max})^2)$ time. By using an extension to Proposition 10.2.7 as in Lemma 10.2.9, $\max\left\{\sum_{S;\,j(S)=l} r_S \;\middle|\; \sum_{S;\,j(S)=l}\mathsf{cost}(S,\emptyset,r_S) \leq B\right\}$ can be computed in $\mathrm{O}((|\mathcal{D}|+|\mathcal{P}|)\cdot(|\mathcal{P}|\cdot p_{\max})^2)$ time. The theorem then follows from Lemma 10.2.5. □

10.2.3 The Approximation Algorithm

Given the above algorithm, the shifting technique is applied as follows. Given an integer a ($0 \leq a \leq k-1$), a line of level j is *active* if it is of the form $y = (hk + a2^{l-j})2^j$ or $x = (vk + a2^{l-j})2^j$ for $h, k \in \mathbb{Z}$. Active lines partition the plane into j-squares as before, shifted with respect to a. Hence we can still apply Theorem 10.2.1. Let C_a denote the set of disks output by the algorithm for the j-squares induced by a and let $C_{\max}$ be the set among such sets with the maximum profit of uniquely covered points of $\mathcal{P}$.

Lemma 10.2.10 $p(uc(C_{\max})) \geq (1 - 4/k) \cdot p(uc(OPT))$, *where* OPT *is a solution for which* $p(uc(OPT))$ *is maximum under* $c(OPT) \leq B$.

Proof: Let $\mathcal{D}_a^b = \bigcup_S \mathcal{D}_{=j(S)}^{\mathrm{b}(S)}$ be the set of disks intersecting the boundary of a j-square S induced by a at their level. Observe that for a fixed value of a the points uniquely covered by OPT by disks in $OPT \cap \mathcal{D}_a^b$ are precisely the points uniquely covered by $OPT \cap \mathcal{D}_a^b$ not covered by disks in $OPT - \mathcal{D}_a^b$. Then it follows from Theorem 10.2.1 that

$$\begin{aligned} p(uc(C_a)) &\geq p\left(uc\left(\bigcup_S OPT_{=j(S)}^{\mathrm{i}(S)}\right)\right) \\ &= p\left(uc\left(OPT - (OPT \cap \mathcal{D}_a^b)\right)\right) \\ &\geq p\left(uc(OPT)\right) - p\left((uc(OPT \cap \mathcal{D}_a^b) - cov(OPT - \mathcal{D}_a^b)\right). \end{aligned}$$

Following the proof of Lemma 7.2.8, a disk is in $\mathcal{D}_a^b$ for at most 4 values of a. Hence

$$\sum_{a=0}^{k-1} p\left(uc(OPT \cap \mathcal{D}_a^b) - cov(OPT - \mathcal{D}_a^b)\right) \leq 4 \cdot p(uc(OPT)).$$

Then

$$\begin{aligned} & k \cdot p(uc(C_{\max})) \\ &\qquad \geq \sum_{a=0}^{k-1} p(uc(C_a)) \\ &\qquad \geq \sum_{a=0}^{k-1} \left(p(uc(OPT)) - p\left(uc(OPT \cap \mathcal{D}_a^b) - cov(OPT - \mathcal{D}_a^b)\right)\right) \\ &\qquad \geq k \cdot p(uc(OPT)) - 4 \cdot p(uc(OPT)). \end{aligned}$$

Therefore $p(uc(C_{\max})) \geq (1 - 4/k) \cdot p(uc(OPT))$. $\square$

In the following, we denote by x the given instance of Budgeted Unique Coverage and by $|x|$ the length of some natural encoding of x. We can clearly assume that $|x| \geq |\mathcal{D}| + |\mathcal{P}|$, where $\mathcal{D}$ and $\mathcal{P}$ denote the given sets of disks and points, respectively.

Theorem 10.2.11 *There is an eptas for Budgeted Unique Coverage on disks of bounded ply, i.e. of ply* $\gamma = \gamma(|x|) = \mathrm{o}(\log |x|)$.

Proof: Consider any $\epsilon > 0$. Choose k as the largest odd integer such that $32k\gamma/\pi \leq \log|x|$. If $k < 5$, output $\emptyset$ (or any other arbitrary solution of cost at most B). Otherwise, scale the profits by $\frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|}$, similar to Theorem 9.1.21. Following the analysis of this theorem and applying Theorem 10.2.1 and Lemma 10.2.10, we obtain a $(1-4/k)\cdot(1-\epsilon)$-approximation of the optimum in

$$\mathrm{O}(k^2(k^2\,|\mathcal{D}| + |\mathcal{P}|)\cdot|\mathcal{D}|\cdot 2^{32k\gamma/\pi}(|\mathcal{P}|^2/\epsilon)^2)$$

time. By the choice of k, this is bounded by

$$\mathrm{O}(\log^2|x|\cdot(|\mathcal{D}|\log^2|x| + |\mathcal{P}|)\cdot|\mathcal{D}|\cdot|x|\cdot(|\mathcal{P}|^2/\epsilon)^2).$$

Hence in time polynomial in the size of the input and $1/\epsilon$, a feasible solution is computed. Furthermore, there is a c_ϵ such that $k \geq 4/\epsilon$ and $k \geq 5$ for all $|x| \geq c_\epsilon$. Therefore if $|x| \geq c_\epsilon$, we obtain a $(1-\epsilon)^2$-approximation of the optimum. The theorem then follows from Theorem 2.2.4. □

This theorem can be easily extended to Budgeted Low-Coverage on fat objects of bounded ply. We simply adjust the algorithm of Lemma 10.2.6 to modulate the profits by the satisfactions and apply scaling as in Theorem 10.1.5.

Theorem 10.2.12 *There is an eptas for Budgeted Low-Coverage on fat objects of bounded ply, i.e. of ply $\gamma = \gamma(|x|) = \mathrm{o}(\log|x|)$.*

If the objects are arbitrary unit disks, but the density of the set of points (i.e. the maximum number of points of $\mathcal{P}$ in any 1×1 box) is bounded, then we can reduce the set of unit disks to a set of unit disks of bounded ply.

Lemma 10.2.13 *For any instance of Budgeted Low-Coverage on a set $\mathcal{D}$ of unit disks and a set $\mathcal{P}$ of points for which the density is bounded by some constant $d > 0$, there is an equivalent instance on a set $\mathcal{D}'$ of unit disks of ply at most $324d^2$ and the same set $\mathcal{P}$ of points.*

Proof: Remove all disks from $\mathcal{D}$ that do not cover any point of $\mathcal{P}$. Then iteratively remove disks d from $\mathcal{D}$ for which there is a disk $d' \in \mathcal{D}$ such that $d \cap \mathcal{P} = d' \cap \mathcal{P}$ and $c(d) \geq c(d')$. Let $\mathcal{D}'$ be the resulting set of disks. Clearly, the instance of Budgeted Low-Coverage on $\mathcal{D}'$ and $\mathcal{P}$ is equivalent to the instance on the original sets $\mathcal{D}$ and $\mathcal{P}$. Moreover, we claim that the ply of this set $\mathcal{D}'$ is at most $324d^2$.

To see this, we apply an argument reminiscent of one by Hochbaum and Maass [150]. Let $\delta > 0$ denote the smallest distance over all disks $d \in \mathcal{D}'$ of d and a point of $\mathcal{P} - d$. For any disk $d \in \mathcal{D}'$, let $\widehat{d}$ denote the disk of radius $1/2 + \min\{\delta, 1/4\}$ centered at the same point as d. We may assume that for any disk $d \in \mathcal{D}'$ there are two points $u, u' \in \mathcal{P}$ such that u and u' lie on the boundary of d or $\widehat{d}$. Otherwise we can move d to such a position while keeping $d \cap \mathcal{P}$ the same. By the construction of $\mathcal{D}'$, this implies that any pair of points $u, u' \in \mathcal{P}$ can 'induce' at most 8 disks of $\mathcal{D}'$.

Now consider some point $p \in \mathbb{R}^2$. By the preceding arguments, any disk of $\mathcal{D}'$ overlapping p is induced by points that are within distance $3/2$ of p. Because the density is d, there are at most $9d$ such points. Including disks that cover a single point of $\mathcal{P}$, the number of disks of $\mathcal{D}'$ overlapping p is at most $8\binom{9d}{2} + 9d \leq 324d^2$. $\square$

The lemma clearly extends to unit-size fat objects.

Now use Lemma 10.2.13 in conjunction with Theorem 10.2.12 to improve on the results of Section 10.1 if the point set has bounded density.

Theorem 10.2.14 *There is an eptas for Budgeted Low-Coverage on unit fat objects and a set of points of bounded density, i.e. of density* $d = d(|x|) = \mathrm{o}(\sqrt{\log |x|})$.

Finally, observe that the algorithms of Theorem 10.2.12 and Theorem 10.2.14 extend to Geometric Budgeted Maximum Coverage, which was already discussed in Chapter 9. To approximate this problem, adjust the algorithm of Lemma 10.2.6 to include the profit of all covered points, not only of uniquely covered points.

Theorem 10.2.15 *There exists an eptas for Geometric Budgeted Maximum Coverage on fat objects of bounded ply, i.e. of ply* $\gamma = \gamma(|x|) = \mathrm{o}(\log |x|)$, *and on unit fat objects and a point set of bounded density, i.e. of density* $d = d(|x|) = \mathrm{o}(\sqrt{\log |x|})$. *Here* $|x|$ *is the size of the input.*

10.3 Geometric Membership Set Cover

We study the geometric version of Minimum Membership Set Cover. We give an approximation algorithm for Geometric Membership Set Cover on unit squares (squares with side length 1), achieving a constant approximation ratio. Its running time is polynomial only if the optimum objective value is bounded by a constant.

Consider an instance of Minimum Membership Set Cover on a set $\mathcal{S}$ of unit squares and a set $\mathcal{P}$ of points. We partition the plane into horizontal slabs of height 1 and compute a separate solution for each such slab. The solution for a slab must cover all points in the slab while ensuring that the maximum membership of points inside and outside the slab is bounded. In the end, the output is the union of the solutions for the different slabs.

To solve the problem for a slab M, we observe that the unit squares of any minimal solution consist of squares intersecting the top boundary of M, all of which are on the lower envelope of their union, and of squares intersecting the bottom boundary of M, all of which are on the upper envelope of their union. This enables a sweep-line approach in which we maintain $2\ell+1$ squares around the current position on each of the two envelopes, where ℓ is a given bound on the maximum membership. The details are presented in the following lemma.

Lemma 10.3.1 *Let $\mathcal{P}$ be a set of points and $\mathcal{S}$ be a set of unit squares. Let M denote the slab contained between the lines $y = 0$ (inclusive) and $y = 1$ (exclusive). Let $\mathcal{P}_M = \mathcal{P} \cap M$ and $\overline{\mathcal{P}_M} = \mathcal{P} - \mathcal{P}_M$. For any constant ℓ, there is a polynomial-time algorithm that either asserts that there is no $C \subseteq \mathcal{S}$ that covers $\mathcal{P}$ with $\mathrm{mem}_C(\mathcal{P}) \leq \ell$, or computes a set $C \subseteq \mathcal{S}$ that covers all points in $\mathcal{P}_M$ and satisfies $\mathrm{mem}_C(\mathcal{P}_M) \leq \ell$ (points inside M are covered at most ℓ times) and $\mathrm{mem}_C(\overline{\mathcal{P}_M}) \leq 2\ell$ (points outside M are covered at most 2ℓ times).*

Proof: Let $\mathcal{S}_t$ ($\mathcal{S}_b$) be the set of squares in $\mathcal{S}$ that intersect the top (bottom) boundary of M. We may assume that no square intersects both boundaries. Then all squares that intersect M are in either $\mathcal{S}_t$ or $\mathcal{S}_b$. We use a vertical sweep-line that moves from left to right and stops at all points in $\mathcal{P}_M$. Let $\mathcal{P}_M = \{u_1, \ldots, u_k\}$, with points indexed in order of nondecreasing x-coordinate. For a given position of the sweep-line, a $(2\ell + 1)$-tuple $\mathsf{s}_t = (s_1, \ldots, s_{2\ell+1})$ of distinct squares from $\mathcal{S}_t$ is a *proper tuple* if all squares are on the lower envelope of their union (in the order $s_1, s_2, \ldots, s_{2\ell+1}$) and the intersection of the sweep-line and this envelope is with $s_{\ell+1}$. We allow any prefix and/or suffix of a proper tuple to consist of dummy objects that do not contain any points in order to represent the case that fewer than ℓ squares appear before or after the current position on the lower envelope. We ignore this technicality below. Proper tuples s_b from $\mathcal{S}_b$ are defined analogously.

If the sweep-line is at the point u_i, then a pair $(\mathsf{s}_t, \mathsf{s}_b)$ of proper tuples is called *admissible* if

- the point u_i is covered by the union of the squares in s_t and s_b, and
- the maximum membership of the union of the squares in s_t and s_b (taking into account all points of $\mathcal{P}$, also the ones outside M) is at most ℓ.

For each point $u_i \in \mathcal{P}_M$, we consider the set A_i of all admissible pairs $(\mathsf{s}_t, \mathsf{s}_b)$. We say that $(\mathsf{s}_t, \mathsf{s}_b) \in A_i$ and $(\mathsf{s}'_t, \mathsf{s}'_b) \in A_{i+1}$ are *compatible* if

- $\mathsf{s}_t = \mathsf{s}'_t$ or $\mathsf{s}'_t = (s_2, \ldots, s_{2\ell+1}, s_{2\ell+2})$ for some new square $s_{2\ell+2}$, and
- the analogous condition holds for s_b and s'_b.

Then we check if a sequence of compatible tuples exists from some $(\mathsf{s}_t, \mathsf{s}_b) \in A_1$ to some $(\mathsf{s}'_t, \mathsf{s}'_b) \in A_k$. If there is such a sequence π, the union of the squares from all tuples of the sequence is the solution C. Otherwise, the algorithm outputs that there is no solution with maximum membership at most ℓ.

We see that the algorithm is correct as follows. If the algorithm outputs a solution C, it is clear that C covers $\mathcal{P}_M$. To bound the maximum membership, note that the solution consists of a set C_t of squares from $\mathcal{S}_t$ that all meet the lower envelope of their union, and a set C_b of squares from $\mathcal{S}_b$ that all meet the upper envelope of their union. Observe that $C = C_t \cup C_b$. We imagine the squares in C_t to be ordered from left to right as they appear on their lower envelope, and similarly for C_b. Terms such as 'before,' 'after,' and 'between' refer to this order. We now argue separately about the maximum membership for points in $\mathcal{P}_M$ and in $\overline{\mathcal{P}_M}$ and show that the membership for points in $\mathcal{P}_M$ is at most ℓ and that for points in $\overline{\mathcal{P}_M}$ it is at most 2ℓ.

For a point u_i in $\mathcal{P}_M$, note that the squares in C_t that contain u_i are consecutive on the lower envelope of C_t. If there were two squares containing u_i and another square in between that does not contain u_i, then the latter square would not be on the lower envelope. Furthermore, if C_t contains any square covering u_i, then also the square that is on the envelope at the x-coordinate of u_i must contain u_i. Hence, if g squares from C_t contain u_i, then the proper tuple s_t of the admissible pair $(\mathsf{s}_t, \mathsf{s}_b)$ that was chosen for the sweep-line at u_i must contain at least $\min\{g, \ell+1\}$ squares containing u_i. The analogous statement holds for C_b and s_b. Therefore, if more than ℓ squares from $C_t \cup C_b$ were to contain u_i, then at least $\ell + 1$ of the squares in the pair $(\mathsf{s}_t, \mathsf{s}_b)$ would contain u_i, and thus the pair would not be admissible. Hence points in $\mathcal{P}_M$ are covered at most ℓ times, i.e. $\mathrm{mem}_C(\mathcal{P}_M) \leq \ell$.

Now consider a point u in $\overline{\mathcal{P}_M}$ that lies below M (the reasoning for points above M is analogous). The point u cannot be contained in any square in C_t. Consider the set C_u of squares in C_b that contain u. We claim that C_u consists of at most two subsets of consecutive (i.e. consecutive on the upper envelope of C_b) squares in C_b. This can be seen as follows. Consider the first (i.e. leftmost) square x in C_b that contains u. Let y be the first square after x in C_b that does not contain u. If y is entirely to the right of u, then no further square in C_b can contain u, and thus C_u is one consecutive subset of C_b (starting with x and ending with the square just before y).

So assume that there is a square after y in C_b that contains u. Then y must be entirely above u. Let z be the first square after y that contains u. Clearly, z must be to the lower right of y (i.e. z can be obtained from y by shifting y right and down). All further squares (after z) whose x-range contains the x-coordinate of u must be to the lower right of z and hence contain u. For if one of them, say w, was to the upper right of z, then z could not be on the upper envelope of C_b, as z would be below the upper envelope of $\{y, w\}$. By repeating this argument for squares to the right of z, we can show that C_u consists of at most two consecutive subsets of C_b.

By construction, the number of consecutive squares from C_b containing u is bounded by ℓ. Otherwise, the sequence π would include a pair of tuples having $\ell+1$ consecutive squares containing u, but such a pair would not be admissible. As there are at most two consecutive subsets of C_b containing u, we have that u is contained in at most 2ℓ squares from C_b. Hence $\mathrm{mem}_C(\overline{\mathcal{P}_M}) \leq 2\ell$.

We have shown that if the algorithm outputs a solution C, then C covers all points in $\mathcal{P}_M$ and satisfies $\mathrm{mem}_C(\mathcal{P}_M) \leq \ell$ and $\mathrm{mem}_C(\overline{\mathcal{P}_M}) \leq 2\ell$. On the other hand, if there is a solution that covers $\mathcal{P}_M$ and has maximum membership at most ℓ, then the $(2\ell + 1)$-tuples of consecutive squares on the two envelopes allow to construct a valid candidate sequence for the algorithm, and thus the algorithm will indeed output a solution. This implies that if the algorithm does not output a solution, then there is no solution that covers $\mathcal{P}_M$ and has membership at most ℓ (and thus also no solution that covers $\mathcal{P}$ and has membership at most ℓ).

For the running time, note that each A_i contains $O(|\mathcal{S}|^{4\ell+2})$ pairs of tuples. Moreover, for each admissible pair in A_i, there are $O(|\mathcal{S}|^2)$ compatible pairs in A_{i+1}. It follows that we can check for the existence of a sequence π of compatible pairs in $O(|\mathcal{P}| \cdot |\mathcal{S}|^{4\ell+4})$ time. □

Theorem 10.3.2 *There is a polynomial-time* 5*-approximation algorithm for instances of Geometric Membership Set Cover on unit squares if the optimal objective value is bounded by an arbitrary constant L.*

Proof: For a given constant ℓ, the following procedure either computes a solution with maximum membership at most 5ℓ or asserts that no solution with maximum membership at most ℓ exists. Partition the plane into horizontal slabs of unit height. For each slab M that contains at least one point from $\mathcal{P}$, run the algorithm of Lemma 10.3.1 to compute a cover $C_M \subseteq \mathcal{S}$ for the points inside M with maximum membership at most ℓ for points in M and at most 2ℓ for points outside M. If for one of the slabs the algorithm of Lemma 10.3.1 outputs that there is no cover with maximum membership at most ℓ, return that the whole instance has no solution with objective value at most ℓ. Otherwise, return the union of the solutions C_M computed for all slabs M. Note that the squares in the solution computed for a slab M can only cover points in M and in the slabs directly above and below M. A point in M is covered at most ℓ times by squares in the solution computed for M, at most 2ℓ times by squares in the solution computed for the slab directly above M, and at most 2ℓ times by squares in the solution computed for the slab directly below M. This shows that every point in $\mathcal{P}$ is covered at most 5ℓ times.

Now run the above procedure for $\ell = 1, 2, \ldots, L$. The first time the procedure returns a cover, we output that cover and terminate. If the procedure does not return a cover for any of the calls, we output that the instance does not have a solution with maximum membership at most L. □

The approximation algorithm does not seem to extend to unit disks directly. One problem is that a point outside a slab could be contained in several unit disks that are not consecutive on the envelope of the selected unit disks. Hence even if the maximum membership of consecutive unit disks on an envelope is bounded by ℓ, the maximum membership of the overall solution could be large.

10.4 Hardness of Approximation

We give several hardness results for both Geometric Unique Coverage and Geometric Membership Set Cover.

We first consider Geometric Unique Coverage. Recall that the approximability of Unique Coverage has not been fully settled yet [83]. We can show however that in some cases the approximability of Geometric Unique Coverage is equal to approximability of general Unique Coverage.

Theorem 10.4.1 *There is a gap-preserving reduction from Unique Coverage to Geometric Unique Coverage on the following objects:*

- *convex polygons,*
- *translated copies of a single polygon,*
- *rotated copies of a single convex polygon,*
- *α-fat objects for any $\alpha > 1$.*

Proof: We use similar reductions as in Section 8.5 in the way outlined in the proof of Theorem 9.2.1. □

This theorem implies for instance that for any $\epsilon > 0$, it is hard to approximate Geometric Unique Coverage on the above object types within ratio $\Omega(1/\log^{\sigma(\epsilon)} n)$, assuming that NP $\not\subseteq$ BPTIME(2^{n^ϵ}), where $\sigma(\epsilon)$ is some constant dependent on ϵ.

For Minimum Membership Set Cover, we can give more results. Recall that Minimum Membership Set Cover is not approximable within $(1-\epsilon)\ln n$ for any $\epsilon > 0$, unless NP $\subset$ DTIME$(n^{\mathrm{O}(\log\log n)})$ [186]. In a more restricted setting, the problem remains APX-hard. Let *Minimum Membership k-Set Cover* be the variant of Minimum Membership Set Cover where each set has cardinality at most k.

Theorem 10.4.2 *Minimum Membership k-Set Cover is APX-hard for $k \geq 4$.*

Proof: Recall that Minimum k-Set Cover is APX-hard for any $k \geq 3$ (by reduction from Minimum Vertex Cover on graphs of degree at most 3 [9]). Kuhn et al. [186] give a gap-preserving reduction from Minimum Set Cover to Minimum Membership Set Cover. Here one element is added to the universe and to each set. Hence a minimum set cover corresponds to a set cover of minimum membership in the new set system and vice versa. The theorem follows immediately. □

Using these results, we can prove the following theorem in the same spirit as Theorem 9.2.1.

Theorem 10.4.3 *Geometric Membership Set Cover is not approximable to $(1-\epsilon)\ln n$ for any $\epsilon > 0$, unless NP $\subset$ DTIME$(n^{\mathrm{O}(\log\log n)})$, on the following objects:*

- *convex polygons,*
- *translated copies of a single polygon,*
- *rotated copies of a single convex polygon,*
- *α-fat objects for any $\alpha > 1$.*

Geometric Membership Set Cover is APX-hard on the following objects:

- *convex polygons with r corners, where $r \geq 4$,*
- *α-fat objects of constant description complexity for any $\alpha > 1$.*

We strengthen these APX-hardness results by showing a lower bound on the approximation ratio that any polynomial-time approximation algorithm for Geometric Membership Set Cover can attain on unit disks and on unit squares.

Theorem 10.4.4 *There is no polynomial-time approximation algorithm attaining an approximation ratio smaller than 2 for Minimum Membership Set Cover on unit disks or unit squares, unless P=NP.*

Proof: We claim that for Geometric Membership Set Cover on unit disks or on unit squares, it is NP-hard to decide if a solution with maximum membership 1 exists or not. It is clear that the theorem statement follows from this claim.

We give a reduction from the NP-complete problem of checking whether a planar graph G of maximum degree 4 is 3-colorable [116]. We create an instance of Minimum Membership Set Cover on unit disks as follows (the construction on unit squares is very similar). First, we compute a rectilinear embedding of G in the plane [255], which determines the layout of the components of the construction. For each vertex v, we construct a vertex gadget as shown in Figure 10.2. In order to cover the point p_v once, a solution must choose exactly one of the three disks containing p_v, and this corresponds to assigning a color to v. Depending on the choice, either 0, 1, or 2 points among the triple of points on the right are already covered.

The next gadget is a transport gadget, which allows transporting a chosen color along a chain of disks. The gadget for transporting information from left to right is shown in Figure 10.3.

Depending on which of the three disks on the left is in the solution, the triple of points uniquely determines which one of the next three disks needs to be chosen to achieve maximum membership equal to 1, and so on.

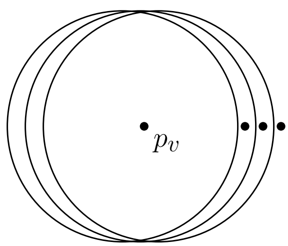

Figure 10.2: The vertex gadget.

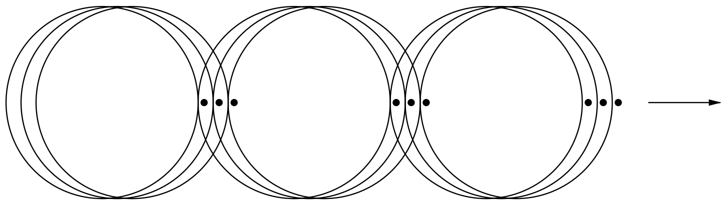

Figure 10.3: The transport gadget.

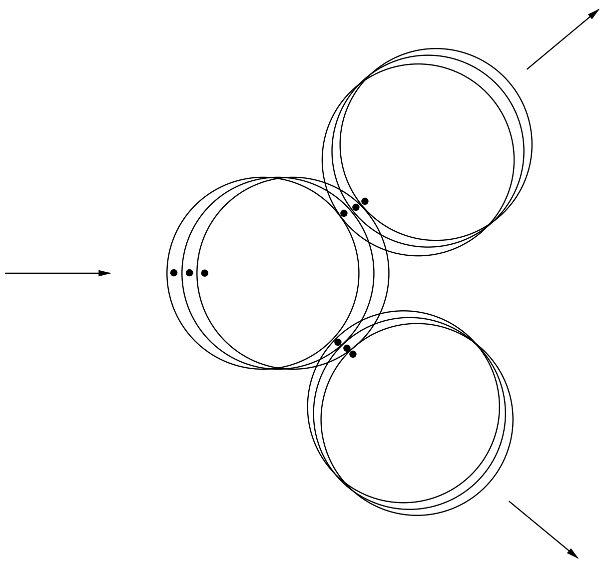

Figure 10.4: The copy gadget.

It is also easy to duplicate information. The copy gadget shown in Figure 10.4 demonstrates how this is accomplished.

Finally, we need a checking gadget to check whether two vertices u and v that are adjacent in G have indeed been assigned different colors. The gadget is shown in Figure 10.5, assuming that a chain transporting the color of u arrives from the left and a chain transporting the color of v arrives from the right. The triple of points on the left (of which 0, 1, or 2 are already covered by the chain transporting u's color) forces a unique choice of exactly one of the three solid disks on the left to be included in the solution, and similarly for the triple of points on the right. The solid disk d_ℓ chosen on the left contains exactly one of a, b, and c, and the same holds for the solid disk d_r chosen on the right. If u and v have received the same color, then d_ℓ and d_r cover the same point among a, b and c, a contradiction to the maximum membership being 1. If u and v have different colors, then d_ℓ and d_r cover two different points among a, b, and c, and the third point can be covered by one of the three dashed disks.

In summary, we have that the constructed instance of Minimum Membership Set Cover on unit disks has a solution with maximum membership 1 if and only if the given planar graph is 3-colorable.

A similar construction is possible for unit squares, only the checking gadget is slightly more complicated (see Figure 10.6). Each of the three points in the middle needs to be covered by a dashed square, but each of the dashed squares contains also the point to its left or to its right (or both). If both vertices have received the same color, the row for that color will have the point to the left and to the right of the middle point already covered, and it is impossible to cover only the point in the middle with a dashed square. □

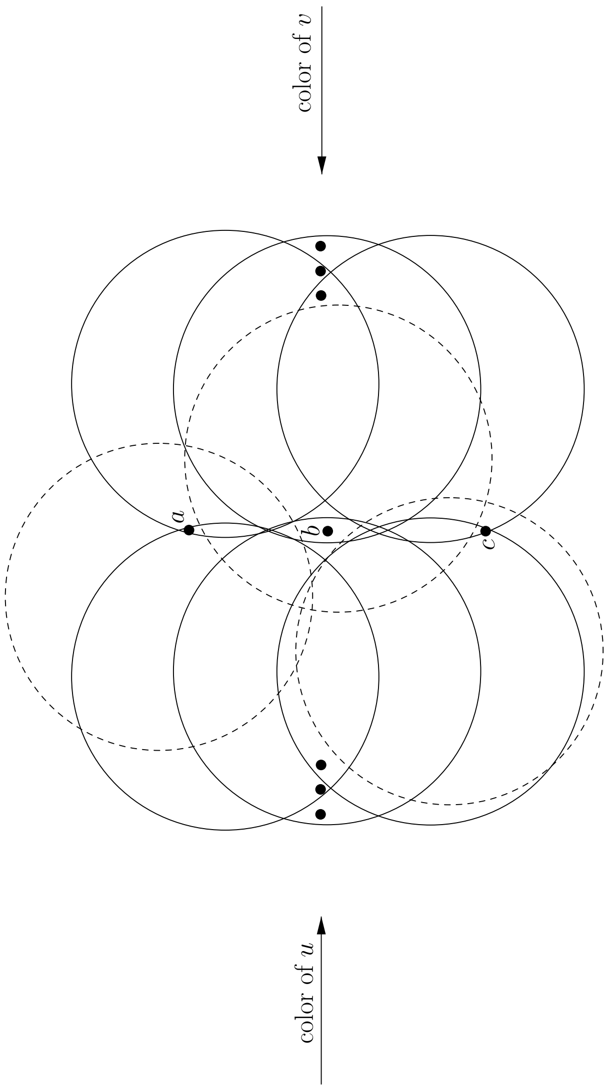

Figure 10.5: The checking gadget.

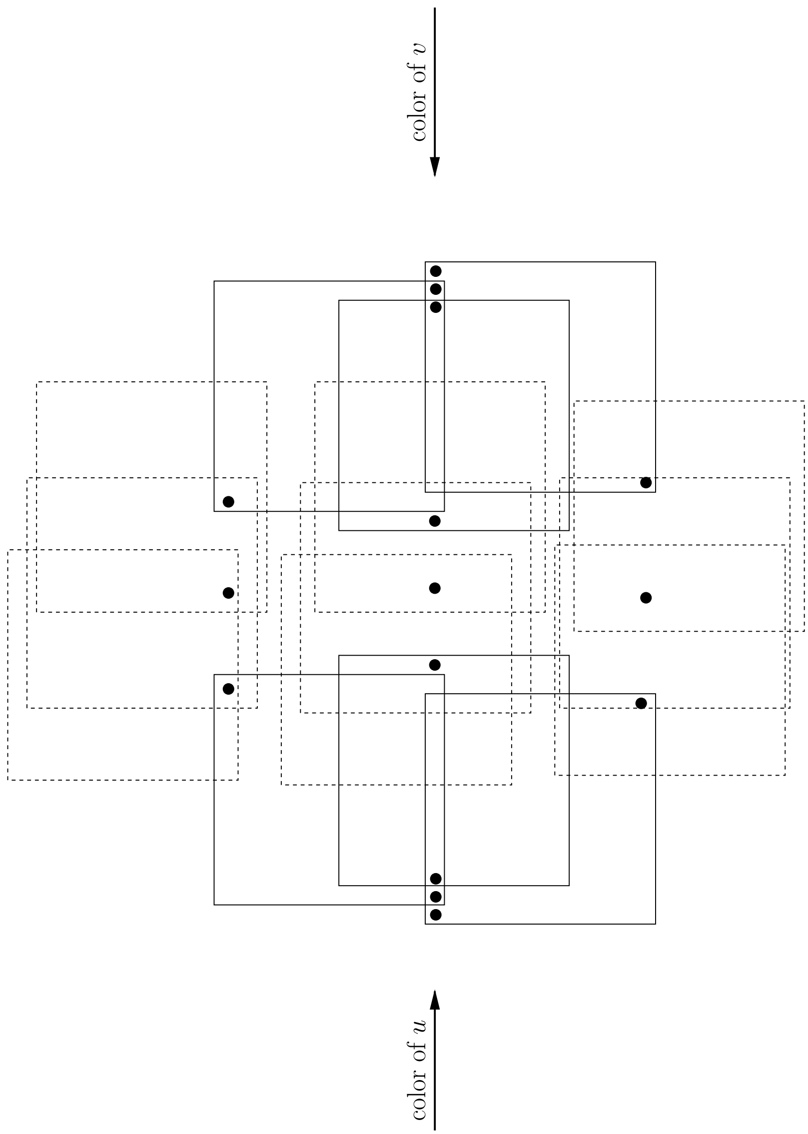

Figure 10.6: The checking gadget for unit squares.