Title	A polynomial-time approximation scheme for the geometric unique coverage problem on unit squares
Author(s)	Ito, Takehiro; Nakano, Shin-ichi; Okamoto, Yoshio; Otachi, Yota; Uehara, Ryuhei; Uno, Takeaki; Uno, Yushi
Citation	Computational Geometry: Theory and Applications, 51: 25-39
Issue Date	2016-01
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/13835
Rights	Copyright (C)2016, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International license (CC BY-NC-ND 4.0). [http://creativecommons.org/licenses/by-nc-nd/4.0/] NOTICE: This is the author's version of a work accepted for publication by Elsevier. Changes resulting from the publishing process, including peer review, editing, corrections, structural formatting and other quality control mechanisms, may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Takehiro Ito, Shin-ichi Nakano, Yoshio Okamoto, Yota Otachi, Ryuhei Uehara, Takeaki Uno, and Yushi Uno, Computational Geometry: Theory and Applications, 51, 2016, 25-39, http://dx.doi.org/10.1016/j.comgeo.2015.10.004
Description	

A Polynomial-Time Approximation Scheme for the Geometric Unique Coverage Problem on Unit Squares

Takehiro Ito[a], Shin-ichi Nakano[b], Yoshio Okamoto[c], Yota Otachi[d], Ryuhei Uehara[d], Takeaki Uno[e], Yushi Uno[f]

[a] *Graduate School of Information Sciences, Tohoku University, Aoba-yama 6–6–05, Sendai, 980–8579, Japan*
[b] *Department of Computer Science, Gunma University, Kiryu 376–8515, Japan*
[c] *Department of Communication Engineering and Informatics, University of Electro-Communications, Chofugaoka 1–5–1, Chofu, Tokyo 182–8585, Japan*
[d] *School of Information Science, Japan Advanced Institute of Science and Technology, Asahidai 1–1, Nomi, Ishikawa 923–1292, Japan*
[e] *Principles of Informatics Research Division, National Institute of Informatics, 2–1–2 Hitotsubashi, Chiyoda-ku, Tokyo, 101–8430, Japan*
[f] *Graduate School of Science, Osaka Prefecture University, 1–1 Gakuen-cho, Naka-ku, Sakai 599-8531, Japan*

Abstract

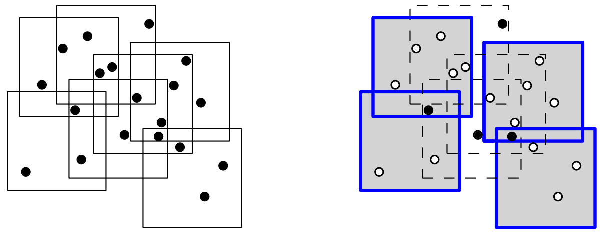
We give a polynomial-time approximation scheme for the unique unit-square coverage problem: given a set of points and a set of axis-parallel unit squares, both in the plane, we wish to find a subset of squares that maximizes the number of points contained in exactly one square in the subset. Erlebach and van Leeuwen (2008) introduced this problem as the geometric version of the unique coverage problem, and the best approximation ratio by van Leeuwen (2009) before our work was 2. Our scheme can be generalized to the budgeted unique unit-square coverage problem, in which each point has a profit, each square has a cost, and we wish to maximize the total profit of the uniquely covered points under the condition that the total cost is at most a given bound.

1. Introduction

Let $\mathcal{P}$ be a set of points and $\mathcal{D}$ a set of axis-parallel unit squares,[1] both in the plane $\mathbb{R}^2$. For a subset $\mathcal{C} \subseteq \mathcal{D}$ of unit squares, we say that a point $p \in \mathcal{P}$ is *uniquely covered* by $\mathcal{C}$ if there is exactly one square in $\mathcal{C}$ containing p. In the *unique unit-square coverage problem*, we are given a pair $\langle \mathcal{P}, \mathcal{D} \rangle$ of a set $\mathcal{P}$ of

Email addresses: `takehiro@ecei.tohoku.ac.jp` (Takehiro Ito), `nakano@cs.gunma-u.ac.jp` (Shin-ichi Nakano), `okamotoy@uec.ac.jp` (Yoshio Okamoto), `otachi@jaist.ac.jp` (Yota Otachi), `uehara@jaist.ac.jp` (Ryuhei Uehara), `uno@nii.ac.jp` (Takeaki Uno), `uno@mi.s.osakafu-u.ac.jp` (Yushi Uno)

[1]Throughout this paper, a unit square is of side length one and is closed, thus contains the boundary.



(a) (b)

Figure 1: (a) An instance $\langle \mathcal{P}, \mathcal{D} \rangle$ of the unique unit-square coverage problem and (b) an optimal solution to $\langle \mathcal{P}, \mathcal{D} \rangle$, where each square in the optimal solution is hatched and each uniquely covered point is drawn as a white circle.

points and a set $\mathcal{D}$ of axis-parallel unit squares as input, and we are asked to find a subset $\mathcal{C} \subseteq \mathcal{D}$ that maximizes the number of points uniquely covered by $\mathcal{C}$. An instance is shown in Figure 1(a), and an optimal solution to this instance is illustrated in Figure 1(b).

In a more general setting, in addition to an instance $\langle \mathcal{P}, \mathcal{D} \rangle$ of the unique unit-square coverage problem, we are given a non-negative real number B, called the *budget*, a non-negative real number $\mathsf{profit}(p)$ for each point $p \in \mathcal{P}$, called the *profit* of p, and a non-negative real number $\mathsf{cost}(S)$ for each square $S \in \mathcal{D}$, called the *cost* of S. In the *budgeted unique unit-square coverage problem*, we are asked to find a subset $\mathcal{C} \subseteq \mathcal{D}$ of total cost at most B such that the total profit of points in $\mathcal{P}$ uniquely covered by $\mathcal{C}$ is maximized. The unique unit-square coverage problem is a specialization of the budgeted unique unit-square coverage problem. To see this, set $\mathsf{profit}(p) = 1$ for all $p \in \mathcal{P}$, $\mathsf{cost}(S) = 0$ for all $S \in \mathcal{D}$, and $B = 0$.

1.1. Past work and motivation

Demaine et al. [7] formulated the non-geometric unique coverage problem in more general setting. They gave a polynomial-time $O(\log n)$-approximation algorithm[2] for the non-geometric unique coverage problem, where n is the number of elements (in the geometric version, n corresponds to the number of points). Guruswami and Trevisan [12] studied the same problem and its generalization, which they called the 1-in-k SAT. The unique coverage problem appears in several situations. The previous papers [7, 12] provide a connection with unlimited-supply single-minded envy-free pricing and the maximum cut problem. For details, see their papers.

The parameterized complexity of the unique coverage problem has also been studied by Misra et al. [19].

[2]For notational convenience, throughout the paper, we say that an algorithm for a maximization problem is α-*approximation* if it returns a solution with the objective value APX such that OPT $\leq \alpha$APX, where OPT is the optimal objective value, and hence $\alpha \geq 1$.

Motivated by applications from wireless networks, Erlebach and van Leeuwen [9] studied the geometric versions of the unique coverage problem especially on unit disks. In the context of wireless networks, each point corresponds to a customer location, and the center of each disk corresponds to a place where the provider can build a base station. If several base stations cover a certain customer location, then the resulting interference might cause this customer to receive no service at all. Ideally, each customer should be serviced by exactly one base station. This situation corresponds to the unique unit-disk coverage problem. They showed that the problem on unit disks is strongly NP-hard, and gave a polynomial-time 18-approximation algorithm; for the budgeted unique unit-disk coverage problem, they provided a polynomial-time $(18+\varepsilon)$-approximation algorithm for any fixed constant $\varepsilon > 0$ [9].

The unique unit-square coverage problem is an ℓ_∞ variant (or an ℓ_1 variant) of the unique unit-disk coverage problem. Erlebach and van Leeuwen [9] introduced the budgeted unique unit-square coverage problem, and gave a polynomial-time $(4+\varepsilon)$-approximation algorithm for any fixed constant $\varepsilon > 0$. Later, van Leeuwen [21] gave a proof that the problem on unit squares is also strongly NP-hard, and improved the approximation ratio to $2+\varepsilon$.

Optimization problems on axis-parallel unit squares and unit disks have been thoroughly studied since Huson and Sen [15]. A seminal paper by Hochbaum and Maass [13] established the shifting strategy, which has been used to give a polynomial-time approximation scheme (PTAS) for a lot of problems on unit squares and unit disks (see [14] for example). However, some problems such as coloring [6] and dispersion [11] (see also [8]) are APX-hard already for unit disks. The unique coverage problem is one among the problems for which we know the NP-hardness, but neither APX-hardness nor a PTAS was known. The existence of a PTAS for unit squares has been asked by van Leeuwen [21].

In a sister paper, we exhibit a polynomial-time approximation algorithm for the unique unit-disk coverage problem with approximation ratio $2+4/\sqrt{3}+\varepsilon$ ($< 4.3095+\varepsilon$), where $\varepsilon > 0$ is any fixed constant [16].

After the conference version [17] of this paper was published, Chan and Hu [4] gave another PTAS for the unique unit-square coverage problem, which is, as they claim, "simpler to describe" than ours.

1.2. Contribution of the paper

In this paper, we give the first PTAS for the unique unit-square coverage problem, and hence we improve the approximation ratio to $1+\varepsilon$ for any fixed constant $\varepsilon > 0$. The algorithm is generalized to give a PTAS for the budgeted unique unit-square coverage problem, too.

We employ the well-known shifting strategy, developed by Baker [1] and applied to the geometric problems by Hochbaum and Maass [13]. Namely, we partition the whole plane into "ribbons" of height one, and delete the points in every $1+\lceil 1/\varepsilon \rceil$ ribbons. Then, the instance is divided into several subinstances in which all points lie in a rectangle of height $\lceil 1/\varepsilon \rceil$. We compute optimal solutions to such subinstances, and take their union. The best among all choices of possible deletions will be a $(1+\varepsilon)$-approximate solution. On the other hand, van Leeuwen [21] was only able to solve a subinstance in a rectangle of height one, and thus only gave a 2-approximation since he removed the points in every two ribbons. A similar approach was used to give a PTAS for the weighted unit-square cover problem [10], but the adaptation to the unique coverage problem is by far more involved, as seen in this paper.

By the strong NP-hardness, we can conclude that there is no fully polynomial-time approximation scheme unless P = NP [20]; in this sense, a PTAS is the best approximation algorithm for the problem.

An extended abstract of this paper has been presented at the 13th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2012) [17].

2. Main result

The following is the main result of the paper.

Theorem 2.1. *For any fixed constant $\varepsilon > 0$, there is a polynomial-time $(1+\varepsilon)$-approximation algorithm for the unique unit-square coverage problem.*

We are given an instance $\langle \mathcal{P}, \mathcal{D} \rangle$. Our algorithm consists of two parts. In the first part, we partition the plane into horizontal ribbons of height one, and show in Section 2.2 that if there is a polynomial-time exact algorithm for the problem restricted to a constant number of ribbons, then the problem on $\langle \mathcal{P}, \mathcal{D} \rangle$ admits a PTAS. As the second part, Section 3 will be devoted to such a polynomial-time exact algorithm.

2.1. Preliminaries

A rectangle is *axis-parallel* if its boundary consists of horizontal and vertical line segments. Let R_W be an (unbounded) axis-parallel rectangle of width W and height ∞ which properly contains all points in $\mathcal{P}$ and all unit squares in $\mathcal{D}$. We fix the origin of the coordinate system on the left vertical boundary of R_W. For a square $S \in \mathcal{D}$, we define the (x, y)-coordinates of S as the coordinates of the top right corner of S; we denote by $x(S)$ the x-coordinate of S, and by $y(S)$ the y-coordinate of S. We can assume without loss of generality that a given set of squares is in general position, which means that, for the purposes of this paper, no horizontal (or vertical) side of a square is on the same line as the horizontal (resp., vertical) side of another square; otherwise, we can scale and translate the squares in polynomial time so that this condition is satisfied [21].

We partition the rectangle R_W into *ribbons* $R_i = [0, W] \times [i, i+1)$, $i \in \mathbb{Z}$, that is, each ribbon is a rectangle of width W and height one. We may assume without loss of generality that no point in $\mathcal{P}$ and no horizontal side of a square in $\mathcal{D}$ is on the same line as the horizontal boundary of any ribbon [21]. Therefore, every unit square of side length one intersects exactly two (consecutive) ribbons. We may assume that each ribbon in R_W contains at least one point in $\mathcal{P}$ and intersects at least one square in $\mathcal{D}$; otherwise, we can simply ignore such ribbons.

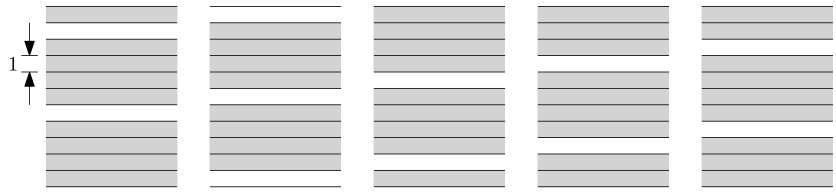


Figure 2: The set R_W^j of ribbons for each $j \in \{0, \ldots, k\}$ when $k = 4$.

We thus deal with only a polynomial number of ribbons. Let $R_1, R_2, \ldots, R_t$ be the ribbons in R_W ordered from bottom to top.

For a set G of ribbons, we denote by $\mathcal{P} \cap G$ the set of all points in $\mathcal{P}$ contained in the ribbons in G. For a point set $\mathcal{P}$ and a square set $\mathcal{C} \subseteq \mathcal{D}$, we denote by $\mathsf{profit}(\mathcal{P}, \mathcal{C})$ the number of points in $\mathcal{P}$ that are uniquely covered by $\mathcal{C}$.

2.2. Restricting the problem to a constant number of ribbons

As the first part of our algorithm, we give the following lemma, by applying the well-known shifting strategy [9, 13].

Lemma 2.2. *Let $k = \lceil 1/\varepsilon \rceil$ be a fixed constant, and suppose that we can obtain an optimal solution to $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$ in polynomial time for every set G consisting of at most k ribbons. Then, we can obtain a $(1 + \varepsilon)$-approximate solution to $\langle \mathcal{P}, \mathcal{D} \rangle$ in polynomial time.*

Proof. For an index $j \in \{0, \ldots, k\}$, let R_W^j be the set of ribbons obtained from R_W by deleting the ribbons R_i, $1 \leq i \leq t$, if and only if $i \equiv j \bmod k + 1$, as illustrated in Figure 2. We regard the remaining (at most) k consecutive ribbons in R_W^j as forming one *group*. Then, those groups have pairwise distance more than one, and hence no square (with side length one) can cover points in two distinct groups. Therefore, we can independently solve the problem on $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$, where G is a group in R_W^j. (Indeed, it suffices to consider the squares in $\mathcal{D}$ which intersect the group G.) Combining the optimal solutions for all groups in R_W^j, we obtain an optimal solution $\mathcal{C}^j \subseteq \mathcal{D}$ to $\langle \mathcal{P} \cap R_W^j, \mathcal{D} \rangle$. As our approximate solution $\mathcal{C}_A \subseteq \mathcal{D}$ to $\langle \mathcal{P}, \mathcal{D} \rangle$, we choose the best one from $\mathcal{C}^j$, $0 \leq j \leq k$, and hence we have

$$\mathsf{profit}(\mathcal{P}, \mathcal{C}_A) \geq \max_{0 \leq j \leq k} \mathsf{profit}(\mathcal{P} \cap R_W^j, \mathcal{C}^j). \quad (1)$$

Clearly, we can obtain the approximate solution $\mathcal{C}_A$ in polynomial time if the problem on $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$ for each group G can be optimally solved in polynomial time.

We now show that the above algorithm is a $(1 + \varepsilon)$-approximation to the original instance $\langle \mathcal{P}, \mathcal{D} \rangle$. Consider an arbitrary optimal solution $\mathcal{C}^* \subseteq \mathcal{D}$ to $\langle \mathcal{P}, \mathcal{D} \rangle$. The shifting strategy [13] with respect to the index j implies that there exists an index $j^* \in \{0, 1, \ldots, k\}$ such that

$$\frac{k}{k+1} \mathsf{profit}(\mathcal{P}, \mathcal{C}^*) \leq \mathsf{profit}(\mathcal{P} \cap R_W^{j^*}, \mathcal{C}^*).$$

Remember that $\mathcal{C}^{j^*}$ is an optimal solution to $\langle \mathcal{P} \cap R_W^{j^*}, \mathcal{D} \rangle$. Therefore, we have $\mathsf{profit}(\mathcal{P} \cap R_W^{j^*}, \mathcal{C}^*) \leq \mathsf{profit}(\mathcal{P} \cap R_W^{j^*}, \mathcal{C}^{j^*})$. Since $k = \lceil 1/\varepsilon \rceil$, we thus have

$$\mathsf{profit}(\mathcal{P}, \mathcal{C}^*) \leq \left(1 + \frac{1}{k}\right) \cdot \mathsf{profit}(\mathcal{P} \cap R_W^{j^*}, \mathcal{C}^*) \leq (1 + \varepsilon) \cdot \mathsf{profit}(\mathcal{P} \cap R_W^{j^*}, \mathcal{C}^{j^*}).$$

By Eq. (1), we thus have $\mathsf{profit}(\mathcal{P}, \mathcal{C}^*) \leq (1 + \varepsilon)\mathsf{profit}(\mathcal{P}, \mathcal{C}_A)$, as required. □

3. Algorithm for a constant number of ribbons

Together with Lemma 2.2, the following lemma completes the proof of Theorem 2.1.

Lemma 3.1. *The unique unit-square coverage problem on $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$ can be optimally solved in polynomial time for a set G consisting of at most k ribbons, where k is a constant.*

The proof of Lemma 3.1 is constructive, namely, we give such an algorithm. In this section, we introduce Lemmas 3.3 and 3.5, which are key lemmas of this paper, and give the whole algorithm based on them; Section 4 gives the proofs of the two key lemmas. The proof of Lemma 3.1 will be based on the key lemmas.

3.1. Basic idea of our algorithm

Our algorithm employs a dynamic programming approach based on the line-sweep paradigm. Namely, we look at points and squares from left to right, and extend the uniquely covered region sequentially. However, adding one square S at the rightmost position can influence a lot of squares that were already chosen, and can change the situation drastically (we say that S *influences* a square S' if the region uniquely covered by S' changes after the addition of S). We therefore need to keep track of the squares that are possibly influenced by a newly added square. Unless the number of those squares is bounded by some constant (or the logarithm of the input size), this approach cannot lead to a polynomial-time algorithm. Unfortunately, new squares may influence arbitrarily many (i.e., a super-constant or super-logarithmic number of) squares.

Instead of adding a square at the rightmost position, we add a square S such that the number of squares that were already chosen and influenced by S can be bounded by a constant. Lemmas 3.3 and 3.5 state that we can do this for any set of squares, as long as a trivial condition for the square set to be an optimal solution is satisfied. Furthermore, such a square can be found in polynomial time.

3.2. Basic definitions

We may assume without loss of generality that the set G consists of consecutive ribbons, forming a group; otherwise we can simply solve the problem for each group, because those groups have pairwise distance more than one. Suppose that G consists of k consecutive ribbons $R_{j+1}, R_{j+2}, \ldots, R_{j+k}$ in R_W,

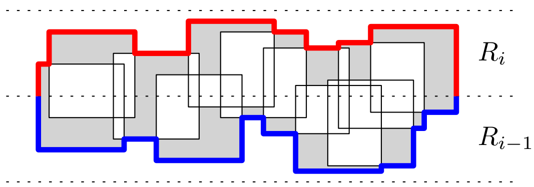


Figure 3: A set $\mathcal{C}$ of squares in $\mathcal{D}_i$, together with $A_1(\mathcal{C})$ (gray), the upper envelope (red) and the lower envelope (blue). The dotted lines show the lower boundaries of R_{i-1}, R_i and R_{i+1}.

ordered from bottom to top, for some integer j. If a square can cover points in $\mathcal{P} \cap G$, then it is totally included in ribbons $R_j, R_{j+1}, \ldots, R_{j+k+1}$. For notational convenience, in the remainder of this section, we assume $j = 0$ without loss of generality. Note that the two ribbons R_0 and R_{k+1} are not in G.

Since no horizontal side of a square is on the same line as the horizontal boundary of any ribbon, if a square in $\mathcal{D}$ intersects G, then it intersects the lower boundary of exactly one ribbon R_i, $i \in \{1, \ldots, k+1\}$. For each $i \in \{1, \ldots, k+1\}$, we denote by $\mathcal{D}_i \subseteq \mathcal{D}$ the subset of all squares in $\mathcal{D}$ intersecting the lower boundary of R_i. Note that the square sets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{k+1}$ form a partition of the squares intersecting G. No square in $\mathcal{D}_i$ intersects any square in $\mathcal{D}_j$ with $j \leq i-2$ or $j \geq i+2$. Furthermore, if a square S_i in $\mathcal{D}_i$ intersects a square S_{i+1} in $\mathcal{D}_{i+1}$ (or a square S_{i-1} in $\mathcal{D}_{i-1}$), then the intersection $S_i \cap S_{i+1}$ must be in R_i (resp., $S_{i-1} \cap S_i$ must be in R_{i-1}).

For a square set $\mathcal{C} \subseteq \mathcal{D}$, let $A_0(\mathcal{C})$, $A_1(\mathcal{C})$, $A_2(\mathcal{C})$ and $A_{\geq 3}(\mathcal{C})$ be the areas covered by no square, exactly one square, exactly two squares, and three or more squares in $\mathcal{C}$, respectively. Then, each point contained in the area $A_1(\mathcal{C})$ is uniquely covered by $\mathcal{C}$.

3.3. Properties on square subsets of $\mathcal{D}_i$

In this subsection, we deal with squares only in a set $\mathcal{C} \subseteq \mathcal{D}_i$ and the region uniquely covered by them. Of course, squares in $\mathcal{D}_{i-1} \cup \mathcal{D}_{i+1}$ may influence squares in $\mathcal{C}$; this difficulty will be discussed in Section 3.5.

3.3.1. Upper and lower envelopes

Let $\mathcal{C} \subseteq \mathcal{D}_i$ be a square set. Since $\mathcal{C}$ is in general position, we can partition the boundary of the closure of $A_1(\mathcal{C})$ into two types: The boundary between $A_0(\mathcal{C})$ and $A_1(\mathcal{C})$; and that between $A_1(\mathcal{C})$ and $A_2(\mathcal{C})$. We call the former the *union boundary* of $\mathcal{C}$. In Figure 3, the union boundary of $\mathcal{C}$ is illustrated as (red or blue) thick lines. We call the union boundary in R_i (or R_{i-1}) the *upper* (resp., *lower*) *envelope* of $\mathcal{C}$. We say that a square S *forms* the boundary of an area A if a portion of a side of S is appears on the boundary of the closure of A. Let $UE(\mathcal{C})$ and $LE(\mathcal{C})$ be the sequences of squares that form the upper and lower envelopes of $\mathcal{C}$, from right to left, respectively. Note that a square $S \in \mathcal{C}$ may appear in both $UE(\mathcal{C})$ and $LE(\mathcal{C})$. An example is shown in Figure 3.

Consider an arbitrary optimal solution $\mathcal{C}^* \subseteq \mathcal{D}_i$ to $\langle \mathcal{P} \cap (R_{i-1} \cup R_i), \mathcal{D}_i \rangle$. If there is a square $S \in \mathcal{C}^*$ contained in the union of $\mathcal{C}^* \setminus \{S\}$, i.e., $S \cap A_1(\mathcal{C}^*) = \emptyset$, then we can simply remove it from $\mathcal{C}^*$ without losing the optimality. Thus, hereafter we deal with a square set $\mathcal{C} \subseteq \mathcal{D}_i$ such that every square S in $\mathcal{C}$ forms the union boundary of $\mathcal{C}$, that is, $S \in UE(\mathcal{C})$ or $S \in LE(\mathcal{C})$ holds. (Note that some square $S \in \mathcal{C}$ may satisfy both $S \in UE(\mathcal{C})$ and $S \in LE(\mathcal{C})$.) This property enables us to extend the upper and lower envelopes sequentially.

3.3.2. Top squares and the key lemma

When we add a "new" square S to the current square set $\mathcal{C} \setminus \{S\}$, we need to know the symmetric difference of $A_1(\mathcal{C})$ and $A_1(\mathcal{C} \setminus \{S\})$: The area $A_1(\mathcal{C}) \setminus A_1(\mathcal{C} \setminus \{S\}) \subseteq A_1(\mathcal{C})$ is the uniquely covered area obtained by adding S, and the area $A_1(\mathcal{C} \setminus \{S\}) \setminus A_1(\mathcal{C}) \subseteq A_2(\mathcal{C})$ is the non-uniquely covered area due to the addition of S. However, it suffices to know $A_1(\mathcal{C} \setminus \{S\}) \setminus A_1(\mathcal{C})$ and its boundary since the boundary of $A_1(\mathcal{C}) \setminus A_1(\mathcal{C} \setminus \{S\})$ is formed only by S and the squares forming the boundary of $A_1(\mathcal{C} \setminus \{S\}) \setminus A_1(\mathcal{C})$.

For a square S in a set $\mathcal{C} \subseteq \mathcal{D}$, let $\Delta(\mathcal{C}, S)$ be the set of all squares in $\mathcal{C}$ that form the boundary of $A_1(\mathcal{C} \setminus \{S\}) \setminus A_1(\mathcal{C})$. An example is shown in Figure 4. Clearly, every square in $\Delta(\mathcal{C}, S)$ has non-empty intersection with S. As we mentioned in Section 3.1, $\Delta(\mathcal{C}, S)$ may contain arbitrarily many (i.e., a super-constant or super-logarithmic number of) squares if we simply choose the rightmost square S in $\mathcal{C}$.

The plan is as follows. (The formal definitions will be given later.)

- For each square set $\mathcal{C} \subseteq \mathcal{D}_i$, we canonically identify a subset of "rightmost" squares, which we call *top squares*. Such a set of top squares of $\mathcal{C}$ has the following property: there always exists a top square S of $\mathcal{C}$ such that $\Delta(\mathcal{C}, S)$ only contains top squares of $\mathcal{C}$. Such a square S will be called *stable* in the set of top squares.

- Some square sets may have the same set of top squares. This defines a preimage: given a set $\mathcal{F}$ of squares, we identify the family of all square sets contained in $\mathcal{D}_i$ such that their sets of top squares are equal to $\mathcal{F}$. This family will be denoted by $\mathfrak{C}_i(\mathcal{F})$. A candidate $\mathcal{F}$ of the set of top squares will be called a *feasible* set. Note that $\mathfrak{C}_i(\mathcal{F}) \neq \emptyset$ when $\mathcal{F}$ is feasible.

- To perform the dynamic programming, we only maintain a value for every feasible set $\mathcal{F}$. The dynamic programming computes the value for $\mathcal{F}$ "from left to right." To obtain the value for $\mathcal{F}$, we look at a stable square S in $\mathcal{F}$ and delete it from each square set $\mathcal{C}$ in $\mathfrak{C}_i(\mathcal{F})$. Then, $\mathcal{C} \setminus \{S\}$ has another set of top squares for which the value was already computed in the course of dynamic programming. Since $\Delta(\mathcal{C}, S)$ only contains top squares of $\mathcal{C}$, we can also compute the area uniquely covered by $\Delta(\mathcal{C}, S)$ in polynomial time. This enables us to calculate the value for $\mathcal{F}$, and we can complete our dynamic-programming computation.

Following this plan, we first give definitions.

For a square set $\mathcal{C} \subseteq \mathcal{D}_i$, a square $S \in \mathcal{C}$ is called a *top square* of $\mathcal{C}$ if one of the following conditions (i)–(iv) holds:

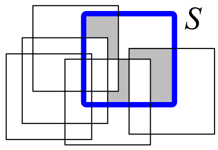


Figure 4: The gray region shows $A_1(\mathcal{C} \setminus \{S\}) \setminus A_1(\mathcal{C})$ for the thick square S.

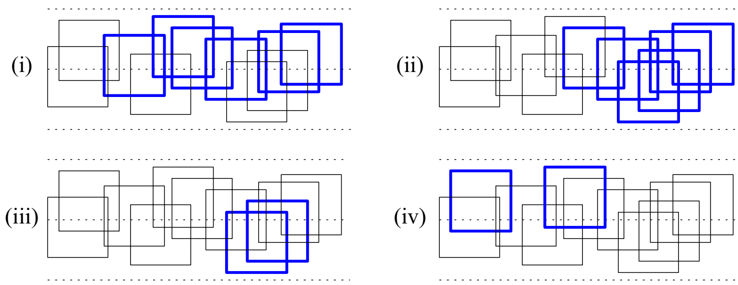


Figure 5: An example of top squares. The (blue) thick squares are top squares, and the numbers correspond to the conditions in the definition.

(i) S is one of the six rightmost squares of $UE(\mathcal{C})$;
(ii) S is one of the six rightmost squares of $LE(\mathcal{C})$;
(iii) S is one of the two rightmost squares of $UE(LE(\mathcal{C}) \setminus UE(\mathcal{C}))$;
(iv) S is one of the two rightmost squares of $LE(UE(\mathcal{C}) \setminus LE(\mathcal{C}))$.

An example is given in Figure 5. We denote by $\mathsf{Top}(\mathcal{C})$ the set of top squares of $\mathcal{C}$. Note that a square may satisfy more than one of the conditions above; indeed, there is no square set $\mathcal{C} \subseteq \mathcal{D}_i$ such that $|\mathsf{Top}(\mathcal{C})| = 16$ since the rightmost square in $\mathcal{C}$ always satisfies both (i) and (ii).

A square set $\mathcal{F} \subseteq \mathcal{D}_i$ is *feasible on* $\mathcal{D}_i$ if $\mathsf{Top}(\mathcal{F}) = \mathcal{F}$. For a feasible square set $\mathcal{F} \subseteq \mathcal{D}_i$, we denote by $\mathfrak{C}_i(\mathcal{F})$ the set of all square subsets of $\mathcal{D}_i$ whose top squares are equal to $\mathcal{F}$, that is,

$$\mathfrak{C}_i(\mathcal{F}) = \{\mathcal{C} \subseteq \mathcal{D}_i \mid \mathsf{Top}(\mathcal{C}) = \mathcal{F}\}.$$

A top square S in a feasible set $\mathcal{F}$ is said to be *stable in* $\mathcal{F}$ if $\Delta(\mathcal{C}, S)$ consists only of top squares in $\mathcal{F}$ for any square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$. The following lemma implies that, for a feasible square set $\mathcal{F} \subseteq \mathcal{D}_i$, we can check in polynomial time whether a top square $S \in \mathcal{F}$ is stable in $\mathcal{F}$.

Lemma 3.2. *Let S be any (top) square in a feasible set $\mathcal{F} \subseteq \mathcal{D}_i$. Then, S is stable in $\mathcal{F}$ if and only if $S' \notin \Delta(\mathcal{F} \cup \{S'\}, S)$ holds for every square $S' \in \mathcal{D}_i \setminus \mathcal{F}$ such that $\mathsf{Top}(\mathcal{F} \cup \{S'\}) = \mathcal{F}$.*

PROOF. By the definition of stable squares, the necessity clearly holds. We thus show the sufficiency: If S is not stable in $\mathcal{F}$, then there exists a non-top square $S' \in \mathcal{D}_i \setminus \mathcal{F}$ such that $\mathsf{Top}(\mathcal{F} \cup \{S'\}) = \mathcal{F}$ and $S' \in \Delta(\mathcal{F} \cup \{S'\}, S)$.

Since S is not stable in $\mathcal{F}$, there exists a square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$ such that $\Delta(\mathcal{C}, S)$ contains non-top squares of $\mathcal{C}$. Let S' be an arbitrary non-top square in $\mathcal{C} \setminus \mathcal{F}$. Then, we have $\mathsf{Top}(\mathcal{F} \cup \{S'\}) = \mathcal{F}$ and $S' \in \Delta(\mathcal{F} \cup \{S'\}, S)$. □

Indeed, stable top squares will be crucial to our algorithm: If a top square S is stable in a feasible set $\mathcal{F} \subseteq \mathcal{D}_i$, then $\Delta(\mathcal{C}, S)$ contains at most 16 top squares in $\mathcal{F}$ for any square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$; and hence we can compute $\Delta(\mathcal{C}, S)$ in polynomial time. Therefore, below is the key lemma for our dynamic programming algorithm, whose proof will be given in Section 4.1.

Lemma 3.3. *For any feasible square set $\mathcal{F} \subseteq \mathcal{D}_i$, there always exists a top square $K(\mathcal{F})$ which is stable in $\mathcal{F}$. Moreover, $K(\mathcal{F})$ can be found in polynomial time.*

The proof of Lemma 3.3 is postponed to Section 4.1. In most cases, we choose the rightmost square of $\mathcal{F}$ as $K(\mathcal{F})$. However, when the rightmost square intersects too many other squares, such a choice does not work. Indeed, $K(\mathcal{F})$ will be one of the following five squares:

1. the rightmost square of $\mathcal{F}$;
2. the rightmost square of $LE(\mathcal{F}) \setminus UE(\mathcal{F})$;
3. the second rightmost square of $LE(\mathcal{F}) \setminus UE(\mathcal{F})$;
4. the rightmost square of $UE(\mathcal{F}) \setminus LE(\mathcal{F})$; and
5. the second rightmost square of $UE(\mathcal{F}) \setminus LE(\mathcal{F})$.

In Figure 5, $K(\mathcal{F})$ is the rightmost square. In Figure 6, the left figure shows a case where $K(\mathcal{F})$ is the rightmost square of $LE(\mathcal{F}) \setminus UE(\mathcal{F})$ and the right figure shows a case where $K(\mathcal{F})$ is the second rightmost square of $LE(\mathcal{F}) \setminus UE(\mathcal{F})$. The other two cases can be obtained symmetrically.

3.4. Algorithm for the problem on $\langle \mathcal{P} \cap (R_{i-1} \cup R_i), \mathcal{D}_i \rangle$

To gain intuition, we first present the dynamic programming algorithm when the set of points is restricted to $\mathcal{P} \cap (R_{i-1} \cup R_i)$ and the set of squares is restricted to $\mathcal{D}_i$. We later generalize this approach to the general case. Let $G = R_{i-1} \cup R_i$. We want to solve the problem on $\langle \mathcal{P} \cap G, \mathcal{D}_i \rangle$ optimally in polynomial time.

For a feasible square set $\mathcal{F}$ on $\mathcal{D}_i$, let $f(\mathcal{F})$ be the maximum number of points in $\mathcal{P} \cap G$ uniquely covered by a square set in $\mathfrak{C}_i(\mathcal{F})$, that is,

$$f(\mathcal{F}) = \max\{\mathsf{profit}(\mathcal{P} \cap G, \mathcal{C}) \mid \mathcal{C} \in \mathfrak{C}_i(\mathcal{F})\},$$

where $\mathsf{profit}(\mathcal{P} \cap G, \mathcal{C})$ is the number of points in $\mathcal{P} \cap G$ that are uniquely covered by $\mathcal{C}$. Then, since every subset of $\mathcal{D}_i$ belongs to $\mathfrak{C}_i(\mathcal{F})$ for some feasible set $\mathcal{F}$, the optimal value $\mathrm{OPT}(\mathcal{P} \cap G, \mathcal{D}_i)$ for $\langle \mathcal{P} \cap G, \mathcal{D}_i \rangle$ can be computed as

$$\mathrm{OPT}(\mathcal{P} \cap G, \mathcal{D}_i) = \max\{f(\mathcal{F}) \mid \mathcal{F} \text{ is feasible on } \mathcal{D}_i\}.$$

Since $|\mathcal{F}| < 16$, this computation can be done in polynomial time if we have the values $f(\mathcal{F})$ for all feasible square sets $\mathcal{F}$ on $\mathcal{D}_i$.

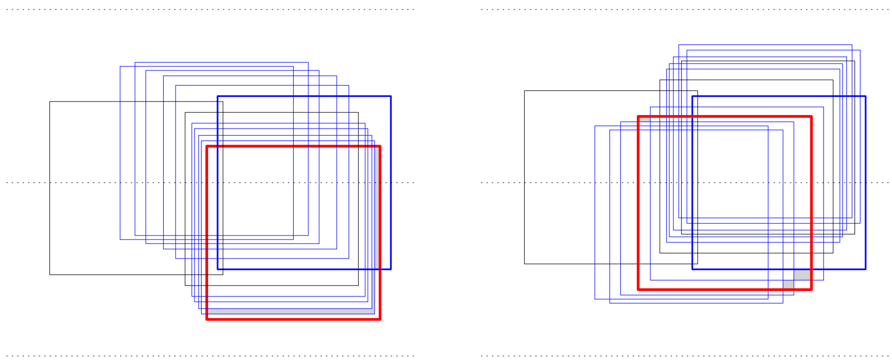


Figure 6: The choice of stable squares. The blue squares are top squares, and the red one is stable. (Left) The rightmost square of $LE(\mathcal{F}) \setminus UE(\mathcal{F})$ is stable in $\mathcal{F}$. (Right) The second rightmost square of $LE(\mathcal{F}) \setminus UE(\mathcal{F})$ is stable in $\mathcal{F}$. In each of the figures, the gray region depicts $A_1(\mathcal{C} \setminus \{S\}) \setminus A_1(\mathcal{C})$ when S is the red square. We may observe that $\Delta(\mathcal{C}, S)$ consists only of top squares. On the other hand, $\Delta(\mathcal{C}, S)$ contains a non-top square (black square) when S is the rightmost square (a thick blue square). Note that, in the right figure, the rightmost square of $LE(\mathcal{F}) \setminus UE(\mathcal{F})$ is not stable in $\mathcal{F}$, and neither of the rightmost square nor the second rightmost square of $UE(\mathcal{F}) \setminus LE(\mathcal{F})$ is stable in $\mathcal{F}$.

We thus compute $f(\mathcal{F})$ in polynomial time for all feasible square sets $\mathcal{F}$ on $\mathcal{D}_i$, according to the "parent-child relation." For two feasible square sets $\mathcal{F}$ and $\mathcal{F}'$ on $\mathcal{D}_i$, we say that $\mathcal{F}'$ is a *child* of $\mathcal{F}$ if there exists a square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$ such that $\mathsf{Top}(\mathcal{C} \setminus \{K(\mathcal{F})\}) = \mathcal{F}'$. The *parent-child relation* for the feasible square sets on $\mathcal{D}_i$ is a binary relation specified by "$\mathcal{F}$ is a child of $\mathcal{F}'$," which may also be viewed as a directed graph such that the vertex set is the family of feasible square sets on $\mathcal{D}_i$ and an arc exists from $\mathcal{F}'$ to $\mathcal{F}$ if and only if $\mathcal{F}$ is a child of $\mathcal{F}'$.

Lemma 3.4. *The parent-child relation for the feasible square sets on $\mathcal{D}_i$ can be constructed in polynomial time. Furthermore, the parent-child relation is acyclic.*

Proof. We can enumerate all feasible square sets on $\mathcal{D}_i$ as follows: We first generate all sets $\mathcal{C} \subseteq \mathcal{D}_i$ consisting of 16 squares, and then check whether $\mathsf{Top}(\mathcal{C}) = \mathcal{C}$. The number of candidates for $\mathcal{C}$ is bounded by $|\mathcal{D}_i|^{16}$ and the check can be done in polynomial time. Therefore, the enumeration can be performed in polynomial time.

For a feasible square set $\mathcal{F}$ on $\mathcal{D}_i$, let $\mathcal{C}$ be any square set in $\mathfrak{C}_i(\mathcal{F})$. Then, we have $|\mathsf{Top}(\mathcal{C} \setminus \{K(\mathcal{F})\}) \setminus \mathsf{Top}(\mathcal{C})| \leq 2$ since the top square $K(\mathcal{F})$ can appear in at most two sets among $UE(\mathcal{C})$, $LE(\mathcal{C})$, $UE(LE(\mathcal{C}) \setminus UE(\mathcal{C}))$ and $LE(UE(\mathcal{C}) \setminus LE(\mathcal{C}))$. Therefore, the number of candidates of children of $\mathcal{F}$ can be bounded by $O(|\mathcal{D}_i|^2)$. We can thus construct the parent-child relation in polynomial time.

For acyclicity, consider the sequence of the x-coordinates of top squares from right to left. Any child $\mathcal{F}'$ has a sequence lexicographically smaller than

its parent $\mathcal{F}$, or $\mathcal{F}' \subset \mathcal{F}$. This implies that the parent-child relation is acyclic.
□

With the parent-child relation, we give the algorithm that solves the problem on $\langle \mathcal{P} \cap G, \mathcal{D}_i \rangle$.

Let $\mathcal{D}^0$ be the square set consisting of the leftmost 16 squares in $\mathcal{D}_i$. As the initialization, we first compute $f(\mathcal{F})$ for all feasible sets $\mathcal{F}$ on $\mathcal{D}^0$. Since $|\mathcal{D}^0|$ is constant, the total number of feasible sets $\mathcal{F}$ on $\mathcal{D}^0$ is also constant. Therefore, this initialization can be done in polynomial time.

We then compute $f(\mathcal{F})$ for a feasible square set $\mathcal{F}$ on $\mathcal{D}_i$ from the values $f(\mathcal{F}')$ for all children $\mathcal{F}'$ of $\mathcal{F}$. Since the parent-child relation is acyclic, we can find a feasible square set $\mathcal{F}$ such that the values $f(\mathcal{F}')$ are already computed for all children $\mathcal{F}'$ of $\mathcal{F}$. For a square set $\mathcal{C} \subseteq \mathcal{D}_i$ and a square $S \in \mathcal{C}$, we denote by $z(\mathcal{C}, S)$ the difference of the number of uniquely covered points in $\mathcal{P} \cap G$ caused by adding S to $\mathcal{C} \setminus \{S\}$, that is, the number of points in $\mathcal{P} \cap G$ that are included in $S \cap A_1(\mathcal{C})$ minus the number of points in $\mathcal{P} \cap G$ that are included in $S \cap A_1(\mathcal{C} \setminus \{S\})$. By the definition of a stable square, we have $z(\mathcal{F}, K(\mathcal{F})) = z(\mathcal{C}, K(\mathcal{F}))$ for all square sets $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$. Therefore, we can correctly compute $f(\mathcal{F})$ by

$$f(\mathcal{F}) = \max\{f(\mathcal{F}') \mid \mathcal{F}' \text{ is a child of } \mathcal{F}\} + z(\mathcal{F}, K(\mathcal{F})).$$

This way, the algorithm correctly solves the problem on $\langle \mathcal{P} \cap G, \mathcal{D}_i \rangle$ in polynomial time.

3.5. Properties on square subsets of $\mathcal{D}$

We then get back to the general case where we want to solve the problem on $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$. Remember that the ribbons $R_0, R_1, \ldots, R_{k+1}$ are ordered from bottom to top, and that $\mathcal{D}_i$ is the set of all squares in $\mathcal{D}$ intersecting the lower boundary of R_i for each $i \in \{1, \ldots, k+1\}$. For a square set $\mathcal{C} \subseteq \mathcal{D}$, let $\mathcal{C}_i = \mathcal{C} \cap \mathcal{D}_i$ for each $i \in \{1, \ldots, k+1\}$. Then, these square sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{k+1}$ form a partition of $\mathcal{C}$.

The plan is as follows.

- We look at the parts $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{k+1}$ in the partition of $\mathcal{C}$, and consider a set of top squares in each part. This way, we may obtain the union of $k+1$ sets of top squares.
- We prove in Lemma 3.5 that there exists at least one top square S in this union such that $\Delta(\mathcal{C}, S)$ consists only of top squares in this union. This square S can be treated as a "stable" square in this general case.
- The property above enables us to develop a dynamic-programming algorithm as hinted in Section 3.4.

We will follow this plan, and introduce the relevant concepts.

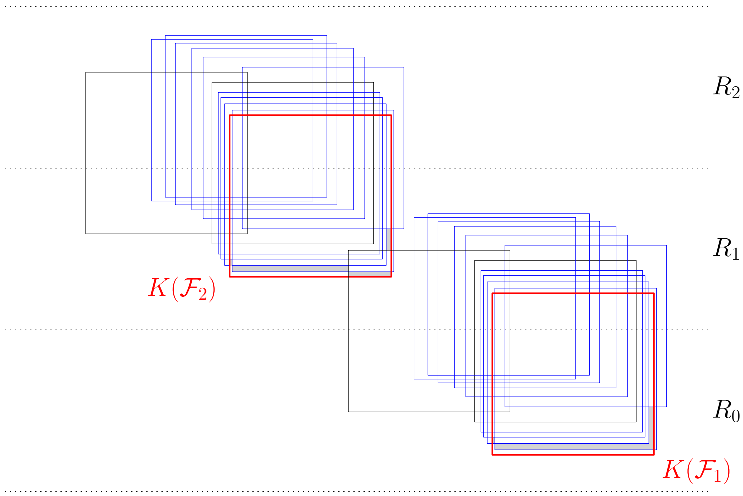


Figure 7: An illustration of a safe family. The blue squares form $\mathcal{F}$, and $K(\mathcal{F}_1)$ and $K(\mathcal{F}_2)$ are shown as red squares. Since $\Delta(\mathcal{C}, K(\mathcal{F}_2))$ has one square that does not belong to $\mathcal{F}$, $\mathcal{F}_2$ is not safe for $\mathcal{F}$. On the other hand, $\mathcal{F}_1$ is safe for $\mathcal{F}$ as $\Delta(\mathcal{C}, K(\mathcal{F}_1)) \subset \mathcal{F}$. The gray region shows $A_1(\mathcal{C} \setminus \{S\}) \setminus A_1(\mathcal{C})$ when S is one of the red squares.

A square set $\mathcal{F} \subseteq \mathcal{D}$ is *feasible on* $\mathcal{D}$ if $\mathsf{Top}(\mathcal{F} \cap \mathcal{D}_i) = \mathcal{F} \cap \mathcal{D}_i$ for each $i \in \{1, \dots, k+1\}$. For a feasible square set $\mathcal{F}$ on $\mathcal{D}$ and $i \in \{1, \dots, k+1\}$, we denote by $\mathcal{F}_i = \mathcal{F} \cap \mathcal{D}_i$, and let

$$\mathfrak{C}(\mathcal{F}) = \{\mathcal{C} \subseteq \mathcal{D} \mid \mathsf{Top}(\mathcal{C}_i) = \mathcal{F}_i \text{ for each } i \in \{1, \dots, k+1\}\}.$$

We say that $\mathcal{F}_i$ is *safe for* $\mathcal{F}$ if $\Delta(\mathcal{C}, K(\mathcal{F}_i)) \subset \mathcal{F}$ for any square set $\mathcal{C} \in \mathfrak{C}(\mathcal{F})$, where $K(\mathcal{F}_i)$ is the stable top square in $\mathcal{F}_i$ which is selected as in the proof of Lemma 3.3. In Figure 7, a case where $k = 2$ is illustrated. There, $\mathcal{F}_1$ is safe for $\mathcal{F}$, but $\mathcal{F}_2$ is not safe for $\mathcal{F}$.

The below is another key lemma for our dynamic programming algorithm, which shows at least one $\mathcal{F}_q$ is safe for $\mathcal{F}$.

Lemma 3.5. *For any feasible square set $\mathcal{F}$ on $\mathcal{D}$, there exists an index $q \in \{1, \dots, k+1\}$ such that $\mathcal{F}_q$ is safe for $\mathcal{F}$.*

The proof will be given in Section 4.2.

3.6. Algorithm for the problem on $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$

We are now ready to describe our algorithm for the problem on $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$. The algorithm follows the guidance of Section 3.4, but the treatment is much more general here.

For a feasible square set $\mathcal{F}$ on $\mathcal{D}$, let $f(\mathcal{F})$ be the maximum number of points in $\mathcal{P} \cap G$ uniquely covered by a square set in $\mathfrak{C}(\mathcal{F})$, that is,

$$f(\mathcal{F}) = \max\{\mathsf{profit}(\mathcal{P} \cap G, \mathcal{C}) \mid \mathcal{C} \in \mathfrak{C}(\mathcal{F})\},$$

where $\mathsf{profit}(\mathcal{P} \cap G, \mathcal{C})$ is the number of points in $\mathcal{P} \cap G$ that are uniquely covered by $\mathcal{C}$. Then, since every subset of $\mathcal{D}$ belongs to $\mathfrak{C}(\mathcal{F})$ for some feasible set $\mathcal{F}$, the optimal value $\mathrm{OPT}(\mathcal{P} \cap G, \mathcal{D})$ for $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$ can be computed as

$$\mathrm{OPT}(\mathcal{P} \cap G, \mathcal{D}) = \max\{f(\mathcal{F}) \mid \mathcal{F} \text{ is feasible on } \mathcal{D}\}.$$

Since $|\mathcal{F}| < 16(k+1)$, this computation can be done in polynomial time if we have the values $f(\mathcal{F})$ for all feasible square sets $\mathcal{F}$ on $\mathcal{D}$.

We thus compute $f(\mathcal{F})$ in polynomial time for all feasible square sets $\mathcal{F}$ on $\mathcal{D}$, according to the "parent-child relation." For a square set $\mathcal{C} \subseteq \mathcal{D}$, we denote simply by $\mathsf{Top}(\mathcal{C}) = \bigcup_{1 \le i \le k+1} \mathsf{Top}(\mathcal{C}_i)$. For a feasible square set $\mathcal{F}$ on $\mathcal{D}$, let $K(\mathcal{F}) = K(\mathcal{F}_q)$ where $\mathcal{F}_q = \mathcal{F} \cap \mathcal{D}_q$ is safe for $\mathcal{F}$. For two feasible square sets $\mathcal{F}$ and $\mathcal{F}'$ on $\mathcal{D}$, we say that $\mathcal{F}'$ is a *child* of $\mathcal{F}$ if there exists a square set $\mathcal{C} \in \mathfrak{C}(\mathcal{F})$ such that $\mathsf{Top}(\mathcal{C} \setminus \{K(\mathcal{F})\}) = \mathcal{F}'$. The *parent-child relation* for the feasible square sets on $\mathcal{D}$ is a binary relation specified by "$\mathcal{F}$ is a child of $\mathcal{F}'$," which may also be viewed as a directed graph as before.

Lemma 3.6. *The parent-child relation for the feasible square sets on $\mathcal{D}$ can be constructed in polynomial time. Furthermore, the parent-child relation is acyclic.*

Proof. We can enumerate all feasible square sets on $\mathcal{D}$, as follows: We first generate all sets $\mathcal{C} \subseteq \mathcal{D}$ consisting of $16(k+1)$ squares, and then check whether $\mathsf{Top}(\mathcal{C} \cap \mathcal{D}_i) = \mathcal{C} \cap \mathcal{D}_i$ for each $i \in \{1, \ldots, k+1\}$. Since k is a constant, this enumeration can be done in polynomial time.

For a feasible square set $\mathcal{F}$ on $\mathcal{D}$, let $\mathcal{C}$ be any square set in $\mathfrak{C}(\mathcal{F})$. Then, we have $|\mathsf{Top}(\mathcal{C} \setminus \{K(\mathcal{F})\}) \setminus \mathsf{Top}(\mathcal{C})| \le 2$ since the top square $K(\mathcal{F}) = K(\mathcal{F}_q)$ can appear in at most two sets among $UE(\mathcal{C}_q)$, $LE(\mathcal{C}_q)$, $UE(LE(\mathcal{C}_q) \setminus UE(\mathcal{C}_q))$ and $LE(UE(\mathcal{C}_q) \setminus LE(\mathcal{C}_q))$. Therefore, the number of candidates of children of $\mathcal{F}$ can be bounded by $O(|\mathcal{D}|^2)$. We can thus construct the parent-child relation in polynomial time.

Consider the sequence of the x-coordinates of top squares from right to left. Any child $\mathcal{F}'$ has a sequence lexicographically smaller than its parent $\mathcal{F}$, or $\mathcal{F}' \subset \mathcal{F}$. This implies that the parent-child relation is acyclic. □

We finally give the algorithm that solves the problem on $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$.

For each $i \in \{1, \ldots, k+1\}$, let $\mathcal{D}_i^0$ be the square set consisting of the first 16 squares in $\mathcal{D}_i$ having the smallest x-coordinates. Let $\mathcal{D}^0 = \bigcup_{1 \le i \le k+1} \mathcal{D}_i^0$, then $|\mathcal{D}^0| \le 16(k+1)$. As the initialization, we first compute $f(\mathcal{F})$ for all feasible sets $\mathcal{F}$ on $\mathcal{D}^0$. Since $|\mathcal{D}^0|$ is a constant, the total number of feasible sets $\mathcal{F}$ on $\mathcal{D}^0$ is also a constant. Therefore, this initialization can be done in polynomial time.

We then compute $f(\mathcal{F})$ for a feasible square set $\mathcal{F}$ on $\mathcal{D}$ from the values $f(\mathcal{F}')$ for all children $\mathcal{F}'$ of $\mathcal{F}$. Since the parent-child relation is acyclic, we can find a feasible square set $\mathcal{F}$ such that the values $f(\mathcal{F}')$ are already computed for all children $\mathcal{F}'$ of $\mathcal{F}$. By Lemma 3.5 there always exists a feasible square set $\mathcal{F}_q = \mathcal{F} \cap \mathcal{D}_q$ on $\mathcal{D}_q$ which is safe for $\mathcal{F}$, and hence by Lemma 3.3 we have a stable top square $K(\mathcal{F}) = K(\mathcal{F}_q)$ in polynomial time. For a square set $\mathcal{C} \subseteq \mathcal{D}$ and a square $S \in \mathcal{C}$, we denote by $z(\mathcal{C}, S)$ the difference of the number of uniquely covered points in $\mathcal{P} \cap G$ caused by adding S to $\mathcal{C} \setminus \{S\}$, that is, the number of points in $\mathcal{P} \cap G$ that are included in $S \cap A_1(\mathcal{C})$ minus the number of points in $\mathcal{P} \cap G$ that are included in $S \cap A_1(\mathcal{C} \setminus \{S\})$. Since $\mathcal{F}_q$ is safe for $\mathcal{F}$ and $K(\mathcal{F}) = K(\mathcal{F}_q)$, we have $z(\mathcal{F}, K(\mathcal{F})) = z(\mathcal{C}, K(\mathcal{F}))$ for all square sets $\mathcal{C} \in \mathfrak{C}(\mathcal{F})$. Therefore, we can correctly update $f(\mathcal{F})$ by

$$f(\mathcal{F}) := \max\{f(\mathcal{F}') \mid \mathcal{F}' \text{ is a child of } \mathcal{F}\} + z(\mathcal{F}, K(\mathcal{F})). \tag{2}$$

This way, the algorithm correctly solves the problem on $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$ in polynomial time.

This completes the proof of Lemma 3.1. □

4. Proofs of key lemmas

To finalize the whole proof, we give proofs of Lemmas 3.3 and 3.5 in Sections 4.1 and 4.2, respectively.

4.1. Proof of Lemma 3.3

To prove Lemma 3.3, we need a thorough preparation. We first give several properties on squares composing uniquely covered regions. Using them, we then give the proof of Lemma 3.3. Remember that we deal with squares only in a set $\mathcal{C} \subseteq \mathcal{D}_i$ and the region uniquely covered by them.

4.1.1. Upper and lower envelopes

We first give the following lemma for the upper envelope; its counterpart holds for the lower envelope by a symmetric argument.

Lemma 4.1. *Let $\mathcal{C} \subseteq \mathcal{D}_i$ be a square set, and suppose that a square $S \in \mathcal{C}$ is not in $UE(\mathcal{C})$. Then, any point in $S \cap R_i$ is covered by at least one square in $UE(\mathcal{C})$.*

We remind that $x(S)$ and $y(S)$ refer to the coordinates of the top right corner of S.

Proof. Let $p = (x, y) \in S \cap R_i$ be an arbitrary point. Consider the vertical line ℓ through p. Then, ℓ meets the upper envelope in R_i. Let $S' \in UE(\mathcal{C})$ be a square that meets ℓ. Since $S \notin UE(\mathcal{C})$, it holds that $x(S') - 1 < x < x(S')$ and $y \le y(S) < y(S')$. Since $p \in R_i$ and $S' \in \mathcal{C} \subseteq \mathcal{D}_i$, we have $y(S') - 1 < y$. Therefore, $p \in S'$. □

We give the following lemma for the upper envelope.

Lemma 4.2. *Let S and S' be any two squares in a square set $\mathcal{C} \subseteq \mathcal{D}_i$ with $x(S) < x(S')$. Suppose that there are q squares $S_1, S_2, \ldots, S_q$, $q \ge 1$, such that $S_j \in UE(\mathcal{C})$ and $x(S) < x(S_j) < x(S')$ for each index $j \in \{1, \ldots, q\}$. Then, any point in $S \cap S' \cap R_i$ is covered by at least $2 + q$ squares unless the intersection is empty.*

Proof. We show that every point $p' = (x', y')$ in $S \cap S' \cap R_i$ is covered by every square S_j, $1 \le j \le q$, that is, both $x(S_j) - 1 \le x' \le x(S_j)$ and $y(S_j) - 1 \le y' \le y(S_j)$ hold.

We first show that $x(S_j) - 1 \le x' \le x(S_j)$ holds. Since $p' \in S \cap S' \cap R_i$ and $x(S) < x(S')$, we have $x(S') - 1 \le x' \le x(S)$. Then, since $x(S) < x(S_j) < x(S')$, we have $x(S_j) - 1 < x(S') - 1 \le x' \le x(S) < x(S_j)$.

We then show that $y(S_j) - 1 \le y' \le y(S_j)$ holds. Since all the squares in $\mathcal{D}_i$ intersect the lower boundary of R_i, we have $y(S_j) - 1 \le y'$. On the other hand, suppose for a contradiction that $y' > y(S_j)$ holds. Since $p' \in S \cap S' \cap R_i$, we have $y' \le \min\{y(S), y(S')\}$. Then, we have $y(S_j) < y' \le \min\{y(S), y(S')\}$. Since $x(S) < x(S_j) < x(S')$, every point $(x'', y(S_j))$, composing the top side of S_j, is contained in $S \cup S'$, where $x(S) - 1 < x(S_j) - 1 \le x'' \le x(S_j) < x(S')$. Thus, the top side of S_j does not appear in the upper envelope of $\mathcal{C}$ at all. This contradicts $S_j \in UE(\mathcal{C})$. □

Similar arguments establish the counterpart for the lower envelope, as follows.

Lemma 4.3. *Let S and S' be any two squares in a square set $\mathcal{C} \subseteq \mathcal{D}_i$ with $x(S) < x(S')$. Suppose that there are q squares $S_1, S_2, \ldots, S_q$, $q \ge 1$, such that $S_j \in LE(\mathcal{C})$ and $x(S) < x(S_j) < x(S')$ for each index $j \in \{1, \ldots, q\}$. Then, any point in $S \cap S' \cap R_{i-1}$ is covered by at least $2 + q$ squares unless the intersection is empty.*

4.1.2. Top squares

We denote by $U\Delta(\mathcal{C}, S)$ the set of all squares that form the boundary of $(A_1(\mathcal{C} \setminus \{S\}) \setminus A_1(\mathcal{C})) \cap R_i$, and by $L\Delta(\mathcal{C}, S)$ the set of all squares that form the boundary of $(A_1(\mathcal{C} \setminus \{S\}) \setminus A_1(\mathcal{C})) \cap R_{i-1}$. By the definition, we clearly have the following lemma.

Lemma 4.4. *Let S and S' be two squares in a set $\mathcal{C} \subseteq \mathcal{D}_i$. Then, S' is not in $U\Delta(\mathcal{C}, S)$ if any point in $S' \cap S \cap R_i$ is contained in $A_{\ge 3}(\mathcal{C} \setminus \{S\})$. Similarly, S' is not in $L\Delta(\mathcal{C}, S)$ if any point in $S' \cap S \cap R_{i-1}$ is contained in $A_{\ge 3}(\mathcal{C} \setminus \{S\})$.*
□

For a feasible square set $\mathcal{F} \subseteq \mathcal{D}_i$, let $UE(\mathcal{F}) = (K_1^\top, K_2^\top, \ldots, K_\alpha^\top)$ with

$$x(K_\alpha^\top) < x(K_{\alpha-1}^\top) < \ldots < x(K_1^\top), \tag{3}$$

and let $LE(\mathcal{F}) = (K_1^\perp, K_2^\perp, \ldots, K_\beta^\perp)$ with

$$x(K_\beta^\perp) < x(K_{\beta-1}^\perp) < \ldots < x(K_1^\perp). \tag{4}$$

Note that some squares may appear in both $UE(\mathcal{F})$ and $LE(\mathcal{F})$. In particular, we always have $K_1^\top = K_1^\perp$. Then, we have the following lemma.

Lemma 4.5. *For a square $K_j^\top \in UE(\mathcal{F})$, $1 \le j \le 4$, suppose that there exists a square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$ such that $U\Delta(\mathcal{C}, K_j^\top)$ contains a non-top square Q of $\mathcal{C}$ with $x(Q) < x(K_j^\top)$. Then, $|LE(\mathcal{F})| \ge 6$ and either $|UE(\mathcal{F})| \le j + 1$ or $x(K_{j+2}^\top) < x(K_6^\perp)$ holds.*

Proof. We first claim that there exists at most one square $K^\top \in UE(\mathcal{C})$ such that $x(Q) < x(K^\top) < x(K_j^\top)$. Suppose for a contradiction that there exist two squares $K, K' \in UE(\mathcal{C})$ such that $x(Q) < x(K) < x(K') < x(K_j^\top)$. Then, by Lemma 4.2 every point in $Q \cap K_j^\top \cap R_i$ is covered by at least four squares and hence is contained in $A_{\geq 3}(\mathcal{C} \setminus \{K_j^\top\})$. By Lemma 4.4 we then have $Q \notin U\Delta(\mathcal{C}, K_j^\top)$, a contradiction.

This claim implies that $Q \notin UE(\mathcal{C})$; otherwise, since $1 \leq j \leq 4$, we have $Q \in \{K_2^\top, K_3^\top, K_4^\top, K_5^\top, K_6^\top\}$ and hence Q is a top square in $\mathcal{F}$. Remember that each square in $\mathcal{C}$ appears in $UE(\mathcal{C})$ or $LE(\mathcal{C})$, and hence we have $Q \in LE(\mathcal{C})$. Then, since $\mathsf{Top}(\mathcal{C}) = \mathcal{F}$ and $Q \notin \mathcal{F}$, we have $|LE(\mathcal{F})| \geq 6$ and

$$x(Q) < x(K_6^\perp). \tag{5}$$

The claim also implies that either $|UE(\mathcal{F})| \leq j+1$ or

$$x(K_{j+2}^\top) < x(Q) \tag{6}$$

holds. By Eqs. (5) and (6) we have $x(K_{j+2}^\top) < x(K_6^\perp)$, as required. □

Similar arguments establish the counterpart of Lemma 4.5, as follows.

Lemma 4.6. *For a square $K_j^\perp \in LE(\mathcal{F})$, $1 \leq j \leq 4$, suppose that there exists a square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$ such that $L\Delta(\mathcal{C}, K_j^\perp)$ contains a non-top square Q of $\mathcal{C}$ with $x(Q) < x(K_j^\perp)$. Then, $|UE(\mathcal{F})| \geq 6$ and either $|LE(\mathcal{F})| \leq j+1$ or $x(K_{j+2}^\perp) < x(K_6^\top)$ holds.* □

Using Lemmas 4.5 and 4.6, we have the following lemma.

Lemma 4.7. *For a feasible square set $\mathcal{F} \subseteq \mathcal{D}_i$, let S_1 be the square in $\mathcal{F}$ with the largest x-coordinate. Then, the following (a) and (b) hold:*

- (*a*) *If there exists a square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$ such that $U\Delta(\mathcal{C}, S_1)$ contains a non-top square of $\mathcal{C}$, then $L\Delta(\mathcal{C}', S_1) \subset \mathcal{F}$ holds for all square sets $\mathcal{C}' \in \mathfrak{C}_i(\mathcal{F})$;*
- (*b*) *If there exists a square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$ such that $L\Delta(\mathcal{C}, S_1)$ contains a non-top square of $\mathcal{C}$, then $U\Delta(\mathcal{C}', S_1) \subset \mathcal{F}$ holds for all square sets $\mathcal{C}' \in \mathfrak{C}_i(\mathcal{F})$.*

The situation (a) is illustrated in Figure 6 (left).

Proof. Note that $S_1 = K_1^\top = K_1^\perp$. We show that (a) holds. (The proof for (b) is similar.)

Suppose that there exists a square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$ such that $U\Delta(\mathcal{C}, S_1)$ contains a non-top square Q of $\mathcal{C}$. Since S_1 is the square with the largest x-coordinate, we have $x(Q) < x(S_1)$. Then, since $S_1 = K_1^\top$, by Lemma 4.5 we have

$$|LE(\mathcal{F})| \geq 6 \tag{7}$$

and either $|UE(\mathcal{F})| \leq 2$ or

$$x(K_3^\top) < x(K_6^\perp) \tag{8}$$

holds.

We now show that $L\Delta(\mathcal{C}', S_1) \subset \mathcal{F}$ holds for all square sets $\mathcal{C}' \in \mathfrak{C}_i(\mathcal{F})$. Suppose for a contradiction that there exists a square set $\mathcal{C}'' \in \mathfrak{C}_i(\mathcal{F})$ such that $L\Delta(\mathcal{C}'', S_1)$ contains a non-top square Q' of $\mathcal{C}''$. Then, since $S_1 = K_1^{\perp}$ and $x(Q') < x(S_1)$, by Lemma 4.6 we have $|UE(\mathcal{F})| \geq 6$ and either $|LE(\mathcal{F})| \leq 2$ or $x(K_3^{\perp}) < x(K_6^{\top})$ holds. By Eq. (7) we thus have

$$x(K_3^{\perp}) < x(K_6^{\top}). \tag{9}$$

Moreover, the inequality $|UE(\mathcal{F})| \geq 6$ implies that Eq. (8) holds. Therefore, by Eqs. (4), (8) and (9) we have $x(K_3^{\top}) < x(K_6^{\top})$. This contradicts Eq. (3). □

Note that Lemma 4.7 implies that, for any square set $\mathcal{C} \in \mathfrak{C}_i(\mathcal{F})$, at most one of $U\Delta(\mathcal{C}, S_1)$ and $L\Delta(\mathcal{C}, S_1)$ can contain non-top squares of $\mathcal{C}$.

4.1.3. Proof of Lemma 3.3

We now prove Lemma 3.3. We consider the following cases, and prove that there is a stable top square $K(\mathcal{F})$ in each case. Let S_1 be the square in $\mathcal{F}$ whose x-coordinate is largest. Note that $S_1 = K_1^{\top} = K_1^{\perp}$.

Case 1: S_1 is stable in $\mathcal{F}$.

In this case, we set $K(\mathcal{F}) = S_1$. Note that by Lemma 3.2 we can check whether S_1 is stable in $\mathcal{F}$ in polynomial time.

Case 2: S_1 is not stable in $\mathcal{F}$.

Since S_1 is not stable in $\mathcal{F}$, by Lemma 3.2 there exists a non-top square $Q \in \mathcal{D}_i \setminus \mathcal{F}$ such that $Q \in \Delta(\mathcal{F} \cup \{Q\}, S_1)$ and $\mathsf{Top}(\mathcal{F} \cup \{Q\}) = \mathcal{F}$. Lemma 4.7 allows us to assume $Q \in U\Delta(\mathcal{F} \cup \{Q\}, S_1)$ without loss of generality. (The case for $Q \in L\Delta(\mathcal{F} \cup \{Q\}, S_1)$ is symmetric.) Then, by Lemma 4.5 we have

$$|LE(\mathcal{F})| \geq 6 \tag{10}$$

and either $|UE(\mathcal{F})| \leq 2$ or

$$x(K_3^{\top}) < x(K_6^{\perp}) \tag{11}$$

holds.

Consider an arbitrary non-top square $Q' \in \mathcal{D}_i \setminus \mathcal{F}$ such that $\mathsf{Top}(\mathcal{F} \cup \{Q'\}) = \mathcal{F}$. We claim that

$$x(Q') < x(K_6^{\perp}). \tag{12}$$

Note that Eq. (10) ensures that the square $K_6^{\perp}$ exists. Since Q' is a non-top square, we clearly have $x(Q') < x(K_6^{\perp})$ if $Q' \in LE(\mathcal{F} \cup \{Q'\})$. We thus consider the case where $Q' \in UE(\mathcal{F} \cup \{Q'\})$. Then, since Q' is a non-top square, $|UE(\mathcal{F})| \geq 6$ and $x(Q') < x(K_6^{\top})$ hold. Furthermore, $|UE(\mathcal{F})| \geq 6$ implies that Eq. (11) holds, and hence by Eq. (3) we have $x(Q') < x(K_6^{\perp})$. Therefore, in either case, Eq. (12) holds.

Let S_2 and S_3 be the rightmost and the second rightmost squares in $LE(\mathcal{F}) \setminus UE(\mathcal{F})$, respectively. Since either $|UE(\mathcal{F})| \leq 2$ or $x(K_3^{\top}) < x(K_6^{\perp})$ holds, at most two squares in $UE(\mathcal{F})$ can appear also in $K_1^\perp, K_2^\perp, \ldots, K_6^\perp$. Furthermore, $S_1 = K_1^\perp = K_1^\top$. Therefore, we have $S_2 \in \{K_2^\perp, K_3^\perp\}$ and $S_3 \in \{K_3^\perp, K_4^\perp\}$. We consider the following two sub-cases.

Case 2-1: S_3 is in $UE(LE(\mathcal{F}) \setminus UE(\mathcal{F}))$.

In this case, we show that S_2 is stable in $\mathcal{F}$, and hence we set $K(\mathcal{F}) = S_2$. By Lemma 3.2 it suffices to show that $Q' \notin \Delta(\mathcal{F} \cup \{Q'\}, S_2)$ for every square $Q' \in \mathcal{D}_i \setminus \mathcal{F}$ such that $\mathsf{Top}(\mathcal{F} \cup \{Q'\}) = \mathcal{F}$.

We first show that $Q' \notin L\Delta(\mathcal{F} \cup \{Q'\}, S_2)$. Since $S_2 \in \{K_2^\perp, K_3^\perp\}$, by Eq. (12) we have

$$x(Q') < x(K_6^\perp) < x(K_5^\perp) < x(K_4^\perp) < x(S_2).$$

By Lemma 4.3 every point in $Q' \cap S_2 \cap R_{i-1}$ is covered by at least five squares, and hence is contained in $A_{\geq 3}((\mathcal{F} \cup \{Q'\}) \setminus \{S_2\})$. By Lemma 4.4 we thus have $Q' \notin L\Delta(\mathcal{F} \cup \{Q'\}, S_2)$, as required.

We then show that $Q' \notin U\Delta(\mathcal{F} \cup \{Q'\}, S_2)$. Since $S_3 \in \{K_3^\perp, K_4^\perp\}$ and $x(S_3) < x(S_2)$, by Eq. (12) we have $x(Q') < x(S_3) < x(S_2)$. Since $S_3 \in UE(LE(\mathcal{F}) \setminus UE(\mathcal{F}))$, by Lemma 4.2 every point in $Q' \cap S_2 \cap R_i$ is covered by at least three squares. Moreover, since $S_2 \notin UE(\mathcal{F})$, by Lemma 4.1 every point in $Q' \cap S_2 \cap R_i$ is covered by at least one square in $UE(\mathcal{F})$. Thus, in total, every point in $Q' \cap S_2 \cap R_i$ is covered by at least four squares in $\mathcal{F}$, and hence is contained in $A_{\geq 3}((\mathcal{F} \cup \{Q'\}) \setminus \{S_2\})$. By Lemma 4.4 we thus have $Q' \notin U\Delta(\mathcal{F} \cup \{Q'\}, S_2)$, as required.

Case 2-2: S_3 is not in $UE(LE(\mathcal{F}) \setminus UE(\mathcal{F}))$.

In this case, we show that S_3 is stable in $\mathcal{F}$, and hence we set $K(\mathcal{F}) = S_3$. By Lemma 3.2 it suffices to show that $Q' \notin \Delta(\mathcal{F} \cup \{Q'\}, S_3)$ for every square $Q' \in \mathcal{D}_i \setminus \mathcal{F}$ such that $\mathsf{Top}(\mathcal{F} \cup \{Q'\}) = \mathcal{F}$.

We first show that $Q' \notin L\Delta(\mathcal{F} \cup \{Q'\}, S_3)$. Since $S_3 \in \{K_3^\perp, K_4^\perp\}$, by Eq. (12) we have

$$x(Q') < x(K_6^\perp) < x(K_5^\perp) < x(S_3).$$

By Lemma 4.3 every point in $Q' \cap S_3 \cap R_{i-1}$ is covered by at least four squares, and hence is contained in $A_{\geq 3}((\mathcal{F} \cup \{Q'\}) \setminus \{S_3\})$. By Lemma 4.4 we thus have $Q' \notin L\Delta(\mathcal{F} \cup \{Q'\}, S_3)$, as required.

We then show that $Q' \notin U\Delta(\mathcal{F} \cup \{Q'\}, S_3)$. Since $S_3 \notin UE(LE(\mathcal{F}) \setminus UE(\mathcal{F}))$, by applying Lemma 4.1 to $LE(\mathcal{F}) \setminus UE(\mathcal{F})$, every point in $Q' \cap S_3 \cap R_i$ is covered by at least one square X in $UE(LE(\mathcal{F}) \setminus UE(\mathcal{F}))$. Moreover, since $S_3 \notin UE(\mathcal{F})$, by applying Lemma 4.1 to $\mathcal{F}$, every point in $Q' \cap S_3 \cap R_i$ is covered by at least one square Y in $UE(\mathcal{F})$. Note that $X \neq Y$ since $X \in UE(LE(\mathcal{F}) \setminus UE(\mathcal{F}))$ and $Y \in UE(\mathcal{F})$. Thus, in total, every point in $Q' \cap S_3 \cap R_i$ is covered by at least four squares (Q', S_3, X, Y), and hence is contained in $A_{\geq 3}((\mathcal{F} \cup \{Q'\}) \setminus \{S_3\})$. By Lemma 4.4 we thus have $Q' \notin U\Delta(\mathcal{F} \cup \{Q'\}, S_3)$, as required. □

4.2. Proof of Lemma 3.5

We first give an auxiliary lemma which states that at least one of $\mathcal{F}_i$ and $\mathcal{F}_{i+1}$ is safe for the other for each $i \in \{1, \ldots, k\}$, and then give the proof of Lemma 3.5.

4.2.1. Auxiliary lemma

Let $\mathcal{F}$ be a feasible square set on $\mathcal{D}$, and let $\mathcal{C}$ be a square set in $\mathfrak{C}(\mathcal{F})$. For each $i \in \{1, \dots, k+1\}$, let $ux(\mathcal{C}_i)$ be the x-coordinate of the leftmost point of the area $R_i \cap K(\mathcal{F}_i) \cap \big(A_1(\mathcal{C}_i) \cup A_2(\mathcal{C}_i)\big)$, while let $lx(\mathcal{C}_i)$ be the x-coordinate of the leftmost point of the area $R_{i-1} \cap K(\mathcal{F}_i) \cap \big(A_1(\mathcal{C}_i) \cup A_2(\mathcal{C}_i)\big)$. Remember that no horizontal side of a square is on the same line as the horizontal boundary of any ribbon, and that $A_1(\mathcal{C}_i) \cap S \neq \emptyset$ for every square $S \in \mathcal{C}_i$. Therefore, $ux(\mathcal{C}_i)$ and $lx(\mathcal{C}_i)$ are well-defined. Since $K(\mathcal{F}_i)$ is stable in $\mathcal{F}_i$, we see that $ux(\mathcal{C}_i)$ is invariant under the choice of $\mathcal{C} \in \mathfrak{C}(\mathcal{F})$. Thus, we also write $ux(\mathcal{F}_i)$ to mean $ux(\mathcal{C}_i)$ for any $\mathcal{C} \in \mathfrak{C}(\mathcal{F})$. The same applies to $lx(\mathcal{F}_i)$.

We first give the following lemma.

Lemma 4.8. *Let $\mathcal{F}_i$ be a feasible square set on $\mathcal{D}_i$. Let $\mathcal{C} \subseteq \mathcal{D}_i$ be any square set in $\mathfrak{C}_i(\mathcal{F}_i)$, and Q be a non-top square of $\mathcal{C}$. Then,*

(*a*) *every point $(x, y) \in Q \cap R_i \cap \big(A_1(\mathcal{C}) \cup A_2(\mathcal{C})\big)$ satisfies $x < ux(\mathcal{F}_i)$; and*

(*b*) *every point $(x, y) \in Q \cap R_{i-1} \cap \big(A_1(\mathcal{C}) \cup A_2(\mathcal{C})\big)$ satisfies $x < lx(\mathcal{F}_i)$.*

Proof. We show that (a) holds; the proof for (b) is symmetric.

Suppose for a contradiction that there exists a point $(x', y') \in Q \cap R_i \cap \big(A_1(\mathcal{C}) \cup A_2(\mathcal{C})\big)$ which satisfies $x' \geq ux(\mathcal{F}_i)$. Since the square $K(\mathcal{F}_i)$ is stable in $\mathcal{F}_i$, no point in $K(\mathcal{F}_i) \cap Q$ is contained in $A_1(\mathcal{C}) \cup A_2(\mathcal{C})$. Therefore, we have

$$K(\mathcal{F}_i) \cap Q \cap R_i \cap \big(A_1(\mathcal{C}) \cup A_2(\mathcal{C})\big) = \emptyset, \tag{13}$$

and hence (x', y') is not contained in $K(\mathcal{F}_i)$. Since Q is a non-top square of $\mathcal{C}$ and $K(\mathcal{F}_i)$ is a top square of $\mathcal{C}$ which is selected as in the proof of Lemma 3.3, we have $x(Q) < x(K(\mathcal{F}_i))$. Since $x' \leq x(Q)$ and $x(K(\mathcal{F}_i)) - 1 \leq ux(\mathcal{F}_i)$, we then have

$$x(K(\mathcal{F}_i)) - 1 \leq ux(\mathcal{F}_i) \leq x' \leq x(Q) < x(K(\mathcal{F}_i)). \tag{14}$$

By Eq. (14) we have $x(K(\mathcal{F}_i)) - 1 \leq x' < x(K(\mathcal{F}_i))$. Since (x', y') is not contained in $K(\mathcal{F}_i)$ and is contained in R_i, we have $y' > y(K(\mathcal{F}_i))$. Notice that by Eq. (14) we have $x(Q) - 1 \leq ux(\mathcal{F}_i) \leq x(Q)$. Then, Q contains any point $(ux(\mathcal{F}_i), y'')$ in $K(\mathcal{F}_i)$ which is in $R_i \cap K(\mathcal{F}_i) \cap \big(A_1(\mathcal{C}_i) \cup A_2(\mathcal{C}_i)\big)$. This contradicts Eq. (13). □

Let $\mathcal{F}$ be a feasible square set on $\mathcal{D}$. Then, for each $i \in \{2, \dots, k\}$, $\mathcal{F}_{i-1}$, $\mathcal{F}_i$ and $\mathcal{F}_{i+1}$ are feasible square sets on $\mathcal{D}_{i-1}$, $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$, respectively. We say that $\mathcal{F}_i$ is *safe for* $\mathcal{F}_{i+1}$ if $\Delta(\mathcal{C}_i \cup \mathcal{C}_{i+1}, K(\mathcal{F}_i)) \subset \mathcal{F}_i \cup \mathcal{F}_{i+1}$ for any square set $\mathcal{C}$ in $\mathfrak{C}(\mathcal{F})$. Similarly, we say that $\mathcal{F}_i$ is *safe for* $\mathcal{F}_{i-1}$ if $\Delta(\mathcal{C}_{i-1} \cup \mathcal{C}_i, K(\mathcal{F}_i)) \subset \mathcal{F}_{i-1} \cup \mathcal{F}_i$ for any square set $\mathcal{C} \in \mathfrak{C}(\mathcal{F})$. For the sake of notational convenience, let $\mathcal{D}_0 = \emptyset$ and $\mathcal{D}_{k+2} = \emptyset$; $\mathcal{F}_1$ is always safe for $\mathcal{F}_0$; and $\mathcal{F}_{k+1}$ is always safe for $\mathcal{F}_{k+2}$. Since each ribbon is of height 1 and each unit square is of side length 1, the square $K(\mathcal{F}_i) \in \mathcal{D}_i$ intersects squares in $\mathcal{D}_{i-1} \cup \mathcal{D}_i \cup \mathcal{D}_{i+1}$ only. Therefore, for $i \in \{1, \dots, k+1\}$, $\mathcal{F}_i$ is safe for $\mathcal{F}$ if and only if $\mathcal{F}_i$ is safe for both $\mathcal{F}_{i-1}$ and $\mathcal{F}_{i+1}$.

Then, Lemma 4.8 gives the following lemma.

Lemma 4.9. *Let $\mathcal{F}$ be a feasible square set on $\mathcal{D}$. Then, for each $i \in \{1, \ldots, k\}$, the following (a) and (b) hold:*

(a) $\mathcal{F}_i$ is safe for $\mathcal{F}_{i+1}$ if $lx(\mathcal{F}_{i+1}) < ux(\mathcal{F}_i)$; and

(b) $\mathcal{F}_{i+1}$ is safe for $\mathcal{F}_i$ if $ux(\mathcal{F}_i) < lx(\mathcal{F}_{i+1})$.

PROOF. We show that (a) holds: If $lx(\mathcal{F}_{i+1}) < ux(\mathcal{F}_i)$, then $\Delta(\mathcal{C}_i \cup \mathcal{C}_{i+1}, K(\mathcal{F}_i)) \subset \mathcal{F}_i \cup \mathcal{F}_{i+1}$ for any square set $\mathcal{C}$ in $\mathfrak{C}(\mathcal{F})$. (The proof for (b) is symmetric.)

Consider an arbitrary square set $\mathcal{C} \in \mathfrak{C}(\mathcal{F})$, and let Q be a square in $\mathcal{C}_i \cup \mathcal{C}_{i+1}$ such that $Q \notin \mathcal{F}_i \cup \mathcal{F}_{i+1}$. We will show that $Q \notin \Delta(\mathcal{C}_i \cup \mathcal{C}_{i+1}, K(\mathcal{F}_i))$. Note that, however, we have $Q \notin \Delta(\mathcal{C}_i \cup \mathcal{C}_{i+1}, K(\mathcal{F}_i))$ if $Q \in \mathcal{C}_i$, because the square $K(\mathcal{F}_i)$ is stable in $\mathcal{F}_i$.

We thus consider the case where $Q \in \mathcal{C}_{i+1}$. Since $K(\mathcal{F}_i) \in \mathcal{C}_i$, the intersection $K(\mathcal{F}_i) \cap Q$ is contained in R_i. Therefore, similarly to Lemma 4.4, we have $Q \notin \Delta(\mathcal{C}_i \cup \mathcal{C}_{i+1}, K(\mathcal{F}_i))$ if any point in $Q \cap K(\mathcal{F}_i) \cap R_i$ is contained in $A_{\geq 3}(\mathcal{C}_i \cup \mathcal{C}_{i+1} \setminus \{K(\mathcal{F}_i)\})$.

Since $K(\mathcal{F}_i) \in \mathcal{C}_i$, if a point in $Q \cap K(\mathcal{F}_i) \cap R_i$ is contained in $A_{\geq 3}(\mathcal{C}_{i+1})$, then the point is contained in $A_{\geq 3}(\mathcal{C}_i \cup \mathcal{C}_{i+1} \setminus \{K(\mathcal{F}_i)\})$. Therefore, we consider a point (x', y') in $Q \cap K(\mathcal{F}_i) \cap R_i$ which is contained in $A_1(\mathcal{C}_{i+1}) \cup A_2(\mathcal{C}_{i+1})$; and hence (x', y') is contained in at least one square in $\mathcal{C}_{i+1}$. Then, by Lemma 4.8 we have $x' < lx(\mathcal{F}_{i+1})$ and hence $x' < ux(\mathcal{F}_i)$. This implies that the point (x', y') is contained in at least three squares in $\mathcal{C}_i$ (one of which is $K(\mathcal{F}_i)$). Thus, the point (x', y') is contained in $A_{\geq 3}(\mathcal{C}_i \cup \mathcal{C}_{i+1} \setminus \{K(\mathcal{F}_i)\})$. □

4.2.2. Proof of Lemma 3.5

Since no vertical side of a square is on the same line as the vertical side of another square, $ux(\mathcal{F}_i) \neq lx(\mathcal{F}_{i+1})$ for each $i \in \{1, \ldots, k\}$. Therefore, by Lemma 4.9 at least one of $\mathcal{F}_i$ and $\mathcal{F}_{i+1}$ is safe for the other. Remember that $\mathcal{F}_1$ is always safe for $\mathcal{F}_0$, and that $\mathcal{F}_{k+1}$ is always safe for $\mathcal{F}_{k+2}$. Therefore, there exists at least one index $q \in \{1, \ldots, k+1\}$, such that $\mathcal{F}_q$ is safe for both $\mathcal{F}_{q-1}$ and $\mathcal{F}_{q+1}$. Then, $\mathcal{F}_q$ is safe for $\mathcal{F}$. □

5. Budgeted version

In this section, we give the following theorem.

Theorem 5.1. *For any fixed constant $\varepsilon > 0$, there is a polynomial-time $(1+\varepsilon)$-approximation algorithm for the budgeted unique unit-square coverage problem.*

We give a sketch how to adapt the algorithm above to the budgeted unique unit-square coverage problem. To this end, we first describe the adaptation to give an optimal solution to $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$ in pseudo-polynomial time when budget, cost, and profit are all integers.

We keep the same strategy, but for the dynamic programming, we slightly change the definition of f. In the budgeted version, $\mathsf{profit}(\mathcal{P}, \mathcal{C})$ means the total profit of the points in $\mathcal{P}$ that are uniquely covered by $\mathcal{C}$, and $\mathsf{cost}(\mathcal{C})$ means the total cost of the squares in $\mathcal{C}$. Let $X = \sum_{p \in \mathcal{P}} \mathsf{profit}(p)$, then $\mathsf{profit}(\mathcal{P}, \mathcal{C}) \leq X$ for any square set $\mathcal{C} \subseteq \mathcal{D}$. For a feasible square set $\mathcal{F} \subseteq \mathcal{D}$ and an integer $x \in \{0, \ldots, X\}$, let $g(\mathcal{F}, x)$ be the minimum total cost of squares in a set $\mathcal{C} \in \mathfrak{C}(\mathcal{F})$ such that the total profit of uniquely covered points in $\mathcal{P} \cap G$ by $\mathcal{C}$ is at least x, that is,

$$g(\mathcal{F}, x) = \min\{\mathsf{cost}(\mathcal{C}) \mid \mathcal{C} \in \mathfrak{C}(\mathcal{F}) \text{ and } \mathsf{profit}(\mathcal{P} \cap G, \mathcal{C}) \geq x\}.$$

If there is no square set $\mathcal{C} \in \mathfrak{C}(\mathcal{F})$ such that $\mathsf{profit}(\mathcal{P} \cap G, \mathcal{C}) \geq x$, then let $g(\mathcal{F}, x) = +\infty$. Then, the optimal value $\mathrm{OPT}(\mathcal{P} \cap G, \mathcal{D})$ for the budgeted version on $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$ can be computed as

$$\mathrm{OPT}(\mathcal{P} \cap G, \mathcal{D}) = \max\{x \mid 0 \leq x \leq X, g(\mathcal{F}, x) \leq B\}.$$

We proceed along the same way as the algorithm in Section 3.6, except for the update formula (2) that should be replaced by

$$g(\mathcal{F}, x) := \min\{g(\mathcal{F}', y) \mid \mathcal{F}' \text{ is a child of } \mathcal{F},\ y + z(\mathcal{F}, K(\mathcal{F})) \geq x\} + \mathsf{cost}(K(\mathcal{F})),$$

where $z(\mathcal{F}, K(\mathcal{F}))$ means the difference of the total profit of uniquely covered points in $\mathcal{P} \cap G$ caused by adding the square $K(\mathcal{F})$ to $\mathcal{F} \setminus \{K(\mathcal{F})\}$. This way, we obtain an optimal solution to $\langle \mathcal{P} \cap G, \mathcal{D} \rangle$ for a group G consisting of at most k consecutive ribbons. Note that the blowup in the running time is only polynomial in X.

Let $R_1, R_2, \ldots, R_t$ be the ribbons in R_W ordered from bottom to top. For each $j \in \{0, \ldots, k\}$, let R_W^j be the set of groups $G_1, G_2, \ldots$, each of which consists of at most k ribbons, obtained from R_W by deleting the ribbons R_i if and only if $i = j \bmod k + 1$, as illustrated in Figure 2. We now explain how to obtain a solution to the problem on $\langle \mathcal{P} \cap R_W^j, \mathcal{D} \rangle$. The adapted algorithm above can solve the problem on each group G_l in R_W^j, and hence suppose that we have computed $g(\mathcal{F}, x)$ for each group G_l and all integers $x \in \{0, \ldots, X\}$. Then, obtaining a solution to $\langle \mathcal{P} \cap R_W^j, \mathcal{D} \rangle$ can be regarded as solving an instance of the multiple-choice knapsack problem [5, 18], as follows: The capacity of the knapsack is equal to the budget B; each $g(\mathcal{F}, x)$ in G_l and $x \in \{0, 1, \ldots, X\}$ have a corresponding item with profit x and cost $g(\mathcal{F}, x)$; and the items corresponding to G_l form a class, from which at most one item can be packed into the knapsack. The multiple-choice knapsack problem can be solved in pseudo-polynomial time which polynomially depends on X [5, 18], and hence we can obtain an optimal solution to $\langle \mathcal{P} \cap R_W^j, \mathcal{D} \rangle$, $0 \leq j \leq k$, in pseudo-polynomial time.

Then, by the standard scale-and-round technique (as used for the ordinary knapsack problem) [5, 18], for any fixed constant $\varepsilon' > 0$, we obtain a $(1 + \varepsilon')$-approximate solution to $\langle \mathcal{P} \cap R_W^j, \mathcal{D} \rangle$ for each $j \in \{0, \ldots, k\}$. Overall, we can obtain such an approximate solution to each of the $k + 1$ subinstances $\langle \mathcal{P} \cap R_W^j, \mathcal{D} \rangle$, $0 \leq j \leq k$, in polynomial time. By taking the best one, we can obtain a $(1 + \varepsilon)$-approximate solution to $\langle \mathcal{P}, \mathcal{D} \rangle$ for any fixed constant $\varepsilon > 0$, by choosing ε' appropriately.

6. Conclusion

The PTAS in this paper combines the well-known shifting strategy [1, 13] and a novel dynamic programming algorithm to solve the problem restricted to regions of constant height, and answers a question by van Leeuwen [21]. The generality of the approach enables us to solve the budgeted version, too.

In a sister paper [16], we give a polynomial-time $(2+4/\sqrt{3}+\varepsilon)$-approximation algorithm for the unique *unit-disk* coverage problem for any fixed constant $\varepsilon > 0$, thus improving the approximation ratio of 18 by Erlebach and van Leeuwen [9]. The basic idea is similar to our PTAS in this paper, but the situation is much more complicated for unit disks.

The reader may wonder why the technique developed in this paper cannot readily yield a PTAS for the unit disk case. The current technique involves two aspects; one is the partition of the whole plane to adapt the shifting strategy, and the other is a polynomial-time algorithm for each group. However, these two aspects may affect each other in the following sense. If we would stick to a partition of the whole plane to obtain a PTAS, then we were not able to develop a polynomial-time algorithm for each group. If we would want to have a polynomial-time algorithm for each group, then the partition could not be good enough to give a better approximation ratio. Indeed, for the unit disks, as treated in our sister paper [16], we have a different way of partitioning the whole plane, so that the polynomial-time algorithm can be developed.[3] Then, the approximation ratio got worse, and we only have a $(2+4/\sqrt{3}+\varepsilon)$-approximation algorithm, not a PTAS.

The running time of our PTAS is a polynomial of degree depending on $1/\varepsilon$. It is desirable to obtain a PTAS such that the degree of its polynomial running time does not depend on $1/\varepsilon$: Such a PTAS is called an efficient PTAS (EPTAS). The existence of an EPTAS would be excluded by showing W[1]-hardness (unless FPT = W[1]) [2, 3], but the unique coverage problem is fixed-parameter tractable [19], thus unlikely to be W[1]-hard. The existence of an EPTAS is left open.

Acknowledgments

The authors thank anonymous referees of the preliminary version and of this journal version for their helpful suggestions. This work is partially supported by Grant-in-Aid for Scientific Research (KAKENHI) including 24106004, 24106005, 24220003, 24700008, 25330003, 26330009, 15H00853, 15K00009, and by the Funding Program for World-Leading Innovative R&D on Science and Technology, Japan.

[3]This sentence contains some inaccuracy since we do not really develop a polynomial-time algorithm for each group, but rather we design a PTAS for each group.

References

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

[2] Bazgan, C.: Schémas d'approximation et complexité paramétrée. Rapport de DEA, Université Paris Sud (1995)

[3] Cesati, M., Trevisan, L.: On the efficiency of polynomial time approximation schemes. Information Processing Letters 64, 165–171 (1997)

[4] Chan, T.M., Hu, N. Geometric red-blue set cover for unit squares and related problems. Computational Geometry: Theory and Applications 48, 380–385 (2015)

[5] Chandra, A.K., Hirschberg, D.S., Wong, C.K.: Approximate algorithms for some generalized knapsack problems. Theoretical Computer Science 3, 293–304 (1976)

[6] Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs, Discrete Mathematics 86, 165–177 (1990)

[7] Demaine, E.D., Hajiaghayi, M.T., Feige, U., Salavatipour, M.R.: Combination can be hard: approximability of the unique coverage problem. SIAM J. on Computing 38, 1464–1483 (2008)

[8] Dumitrescu, A., Jiang, M.: Dispersion in disks. Theory Comput. Syst. 51, 125–142 (2012)

[9] Erlebach, T., van Leeuwen, E.J.: Approximating geometric coverage problems. Proc. SODA 2008, pp. 1267–1276 (2008)

[10] Erlebach, T., van Leeuwen, E.J.: PTAS for weighted set cover on unit squares. Proc. APPROX-RANDOM 2010, pp. 166–177 (2010)

[11] Fiala, J., Kratochvíl, J., Proskurowski, A.: Systems of distant representatives. Discrete Applied Mathematics 145, 306–316 (2005)

[12] Guruswami, V., Trevisan, L.: The complexity of making unique choices: Approximating 1-in-k SAT. Proc. APPROX-RANDOM 2005, pp. 99–110 (2005)

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

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

[15] Huson, M.L., Sen, A.: Broadcast scheduling algorithms for radio networks. Proc. IEEE MILCOM 1995, pp. 647–651 (1995)

[16] Ito, T., Nakano, S.-i., Okamoto, Y., Otachi, Y., Uehara, R., Uno, T., Uno, Y.: A 4.31-approximation for the geometric unique coverage problem on unit disks. Theoretical Computer Science 544, 14–31 (2014)

[17] Ito, T., Nakano, S.-i., Okamoto, Y., Otachi, Y., Uehara, R., Uno, T., Uno, Y.: A polynomial-time approximation scheme for the geometric unique coverage problem on unit squares. Proc. of SWAT 2012, pp. 24–35 (2012)

[18] Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer-Verlag, 2004.

[19] Misra, N., Moser, H., Raman, V., Saurabh, S., Sikdar, S.: The parameterized complexity of unique coverage and its variants. Algorithmica 65, 517–544 (2013)

[20] Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, 1994

[21] van Leeuwen, E. J.: Optimization and approximation on systems of geometric objects. Ph.D. Thesis, University of Amsterdam, 2009