Discriminating Codes in Geometric Setups

Sanjana Dey
ACM Unit, Indian Statistical Institute, Kolkata, India

Florent Foucaud
Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, 33400 Talence, France

Subhas C. Nandy
ACM Unit, Indian Statistical Institute, Kolkata, India

Arunabha Sen
Arizona State University, Tempe, AZ, USA

Abstract

We study two geometric variations of the discriminating code problem. In the *discrete version*, a finite set of points P and a finite set of objects S are given in $\mathbb{R}^d$. The objective is to choose a subset $S^* \subseteq S$ of minimum cardinality such that the subsets $S_i^* \subseteq S^*$ covering p_i, satisfy $S_i^* \neq \emptyset$ for each $i = 1, 2, \ldots, n$, and $S_i^* \neq S_j^*$ for each pair (i, j), $i \neq j$. In the *continuous version*, the solution set S^* can be chosen freely among a (potentially infinite) class of allowed geometric objects.

In the 1-dimensional case ($d = 1$), the points are placed on some fixed-line L, and the objects in S are finite segments of L (called intervals). We show that the discrete version of this problem is NP-complete. This is somewhat surprising as the continuous version is known to be polynomial-time solvable. This is also in contrast with most geometric covering problems, which are usually polynomial-time solvable in 1D.

We then design a polynomial-time 2-approximation algorithm for the 1-dimensional discrete case. We also design a PTAS for both discrete and continuous cases when the intervals are all required to have the same length.

We then study the 2-dimensional case ($d = 2$) for axis-parallel unit square objects. We show that both continuous and discrete versions are NP-hard, and design polynomial-time approximation algorithms with factors $4 + \epsilon$ and $32 + \epsilon$, respectively (for every fixed $\epsilon > 0$).

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Discriminating code, Approximation algorithm, Segment stabbing, Geometric Hitting set

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.24

Related Version A full version of the paper is available at `https://arxiv.org/abs/2009.10353`.

Funding *Florent Foucaud*: This author was partially funded by the ANR project HOSIGRA (ANR-17-CE40-0022) and the IFCAM project "Applications of graph homomorphisms" (MA/IFCAM/18/39).

1 Introduction

We consider geometric versions of the Discriminating Code problem, which are variations of classic geometric covering problems. A set of point sites P in $\mathbb{R}^d$ is given. For a set S of objects of $\mathbb{R}^d$, denote by S_i the set of objects of S that contain $p_i \in P$. The objective is to choose a minimum-size set S^* of objects such that $S_i^* \neq \emptyset$ for all $p_i \in P$ (covering), and $S_i^* \neq S_j^*$ for each pair of distinct sites $p_i, p_j \in P$ (discrimination). In the *discrete* version, the objects of S^* must be chosen among a specified set S of objects given in the input, while in the *continuous* version, only the points are given, and the objects can be chosen freely (among some infinite class of allowed objects).

31st International Symposium on Algorithms and Computation (ISAAC 2020).
Editors: Yixin Cao, Siu-Wing Cheng, and Minming Li; Article No. 24; pp. 24:1–24:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The problem is motivated as follows. Consider a terrain that is difficult to navigate. A set of sensors, each assigned a unique identification number (*id*), are deployed in that terrain, all of which can communicate with a single base station. If a region of the terrain suffers from some specific problem, a subset of sensors will detect that and inform the base station. From the *id*'s of the alerted sensors, one can uniquely identify the affected region, and a rescue team can be sent. The covering zone of each sensor can be represented by an object in S. The arrangement of the objects divides the entire plane into regions. A representative point of each region may be considered as a site. The set P consists of some of those sites. We need to determine the minimum number of sensors such that no two sites in P are covered by the same set of *id*s. Apart from coverage problems in sensor networks, this problem has applications in fault detection, heat prone zone in VLSI circuits, disaster management, environmental monitoring, localization and contamination detection [18, 24], to name a few.

The general version of the problem has been formulated as a graph problem, as follows.

> MINIMUM DISCRIMINATING CODE (MIN-DISC-CODE) [5, 6]
> **Input:** A connected bipartite graph $G = (U \cup V, E)$, where $E \subseteq \{(u, v)|u \in U, v \in V\}$.
> **Output:** A minimum-size subset $U^* \subseteq U$ such that $U^* \cap N(v) \neq \emptyset$ for all $v \in V$, and $U^* \cap N(v) \neq U^* \cap N(v')$ for every pair $v, v' \in V$, $v \neq v'$.

In the geometric version of MIN-DISC-CODE, which will be further referred to as the G-MIN-DISC-CODE, the two sets of nodes in the bipartite graph are U = a set of geometric objects S, and V = a set of points P in $\mathbb{R}^d$, and an object is adjacent to all the points it contains. The *code* of a point $p \in P$ with respect to a subset $S' \subseteq S$ is the subset of S' that contains p. Given an instance (P, S), two points $p_i, p_j \in P$ are called *twins* if each member in S that contains p_i also contains p_j, and vice-versa. An instance (P, S) of G-MIN-DISC-CODE is *twin-free* if no two points in P are twins. Geometrically, if we consider the arrangement [8] $\mathcal{A}$ of the geometric objects S, then the instance (P, S) is twin-free if each cell of $\mathcal{A}$ contains at most one point of P. As mentioned earlier, for a twin-free instance, a subset of S that can uniquely assign codes to all the points in P is said to *discriminate* the points of P and is called a *discriminating code* or *disc-code* in short. In the discrete version of the problem, our objective is to find a subset $S^* \subseteq S$ of minimum cardinality that is a disc-code for the points in P. In the continuous version, we can freely choose the objects of S^*. The two problems are formally stated as follows.

> DISCRETE-G-MIN-DISC-CODE
> **Input:** A point set P to be discriminated, and a set of objects S to be used for the discrimination.
> **Output:** A minimum-size subset $S^* \subseteq S$ which discriminates all points in P.

> CONTINUOUS-G-MIN-DISC-CODE
> **Input:** A point set P to be discriminated.
> **Output:** A minimum-size set S^* of objects that discriminate the points in P, and that can be placed *anywhere* in the region under consideration.

Related work. The general MIN-DISC-CODE problem is NP-hard and hard to approximate [5, 6, 17]. In the context of the above-mentioned practical applications, DISCRETE-G-MIN-DISC-CODE in 2D was defined in [2], where an integer programming formulation (ILP) of the problem was given along with an experimental study. CONTINUOUS-G-MIN-DISC-CODE was introduced in [13], and shown to be NP-complete for disks in 2D, but polynomial-time in 1D (even when the intervals are restricted to have bounded length). These two problems are related to the class of *geometric covering problems*, for which also both the discrete and continuous version are studied extensively [16]. A related problem is the TEST COVER problem [9], which is similar to MIN-DISC-CODE (but defined on hypergraphs). It is equivalent to the variant of MIN-DISC-CODE where the covering condition "$U^* \cap N(v) \neq \emptyset$" is not required. Thus, a discriminating code is a test cover, but the converse may not be true.

Geometric versions of TEST COVER have been studied under various names. For example, the *separation* problems in [3, 7, 14] can be seen as continuous geometric versions of test cover in 2D, where the objects are half-planes. TEST COVER behaves very similarly to MIN-DISC-CODE, and our techniques could be applied to TEST COVER to obtain similar results. Such results do not exist in the literature. Similar problems are also called *shattering* problems, see [22]. A well-studied special case of MIN-DISC-CODE for graphs is the problem MINIMUM IDENTIFYING CODE (MIN-ID-CODE). This problem was studied in particular for the related setting of geometric intersection graphs, for example on unit disk graphs [20] and interval graphs [4, 10, 11].

Our results. We show that DISCRETE-G-MIN-DISC-CODE in 1D, that is, the problem of discriminating points on a real line by interval objects of arbitrary length, is NP-complete. For this we reduce from 3-SAT. Here, the challenge is to overcome the linear nature of the problem and to transmit the information across the entire construction without affecting intermediate regions. This result is in contrast with CONTINUOUS-G-MIN-DISC-CODE in 1D, which is polynomial-time solvable [13]. This is also in contrast with most geometric covering problems, which are usually polynomial-time solvable in 1D [16]. We then design a polynomial-time 2-factor approximation algorithm for DISCRETE-G-MIN-DISC-CODE in 1D. To this end we use the concept of minimum edge-covers in graphs, whose optimal solution can be found by computing a maximum matching of the graph. We also design a polynomial-time approximation scheme (PTAS) for both DISCRETE-G-MIN-DISC-CODE and CONTINUOUS-G-MIN-DISC-CODE in 1D, when the objects are required to all have the same (unit) length. We also study both problems in 2D for axis-parallel unit square objects, which form a natural extension of 1D intervals to the 2D setting. The continuous version is known to be NP-complete for unit disks [13], and we show that the reduction can be adapted to our setting, for both the continuous and discrete case. We then design polynomial-time constant-factor approximation algorithms for both problems in the same setting, of factors $4+\epsilon$ for CONTINUOUS-G-MIN-DISC-CODE, and $32+\epsilon$ for DISCRETE-G-MIN-DISC-CODE (for any fixed $\epsilon > 0$). To this end, we re-formulate the problem as an instance of stabbing a set L of given line segments by placing unit squares in $\mathbb{R}^2$. (Here a line segment $\ell \in L$ is *stabbed* by a unit square if exactly one end-point of ℓ is contained in the square.)

We propose a 4-factor approximation algorithm for this stabbing problem, which, to the best of our knowledge, is the first polynomial-time constant-factor algorithm for it.[1]

Our results are summarized in Table 1. Due to space restrictions, the proofs of the statements marked with $\star$ can be found in the full version.

[1] Such algorithms exist for a related, but different, segment-stabbing problem by unit disks, where a disk stabs a segment if it intersects it once or twice [21, 15].

Table 1 Summary of our results.

Object Type	Continuous-G-Min-Disc-Code		Discrete-G-Min-Disc-Code	
	Hardness	Algorithm	Hardness	Algorithm
1D intervals	-	Polynomial [13]	NP-hard (Thm. 5)	2-approximable (Thm. 9)
1D unit intervals	Open	PTAS (Thm. 13)	Open	PTAS (Thm. 13)
2D axis parallel unit squares	NP-hard (Thm. 14)	$(4+\epsilon)$-approximable (Thm. 17)	NP-hard (Thm. 14)	$(32+\epsilon)$-approximable (Thm. 18)

2 The one-dimensional case

An instance (P, S) of Discrete-G-Min-Disc-Code is a set $P = \{p_1, \ldots, p_n\}$ of points and a set S of m intervals of arbitrary lengths placed on the real line $\mathbb{R}$. Assuming that the points are sorted with respect to their coordinate values, we define $n+1$ *gaps* $\mathcal{G} = \{g_1, \ldots, g_{n+1}\}$, where $g_1 = (-\infty, p_1)$, $g_i = (p_{i-1}, p_i)$ for $2 \leq i \leq n$, and $g_{n+1} = (p_n, \infty)$. One can check whether (P, S) is twin-free in $O(n \log n + m \log m)$ i.e. $O(m \log m)$ because $m \geq \frac{n}{2}$.

Observe that (i) if both endpoints of an interval $s \in S$ lie in the same gap of $\mathcal{G}$, then it can not discriminate any pair of points; thus s is *useless*, and (ii) if more than one interval in S have both their endpoints in the same two gaps, say $g_a = (p_a, p_{a+1}), g_b = (p_b, p_{b+1}) \in \mathcal{G}$, then both of them discriminate the exact same point-pairs. Thus, they are *redundant* and we need to keep only one such interval. In a linear scan, we can first eliminate the useless and redundant intervals. From now onwards, m will denote the number of intervals, none of which are useless or redundant. Hence, $m = O(n^2)$.

2.1 NP-completeness for the general 1D case

Discrete-G-Min-Disc-Code is in NP, since given a subset $S' \subseteq S$, in polynomial time one can test whether the problem instance (P, S') is twin-free (i.e. whether the code of every point in P induced by S' is unique). Our reduction for proving NP-hardness is from the NP-complete 3-SAT-2l problem [26] (defined below), to Discrete-G-Min-Disc-Code.

> 3-SAT-2l
> **Input:** A collection of m clauses $C = \{c_1, c_2, \ldots, c_m\}$ where each clause contains at most three literals, over a set of n Boolean variables $X = \{x_1, x_2, \ldots, x_n\}$, and each literal appears at most twice.
> **Output:** A truth assignment of X such that each clause is satisfied.

Given an instance (X, C) of 3-SAT-2l, we construct in polynomial time an instance $\Gamma(X, C)$ of Discrete-G-Min-Disc-Code on the real line $\mathbb{R}$. The main challenge of this reduction is to be able to connect variable and clause gadgets, despite the linear nature of our 1D setting. The basic idea is that we will construct an instance where some specific set of *critical* point-pairs will need to be discriminated (all other pairs being discriminated by some partial solution forced by our gadgets). Let us start by describing our basic gadgets.

▶ **Definition 1.** *A* covering gadget Π *consists of three intervals I, J, K and four points p_1, p_2, p_3 and p_4 satisfying $p_1 \in I$, $p_2 \in I \cap J$, $p_3 \in I \cap J \cap K$ and $p_4 \in J \cap K$ as in Fig. 1. Every other interval of the construction will either contain all four points, or none. There may exist a set of points in $K \setminus \{I \cup J\}$, depending on the need of the reduction.*

▶ **Observation 2.** *Points p_1,p_2,p_3,p_4 can only be discriminated by choosing all three intervals I, J, K in the solution.*

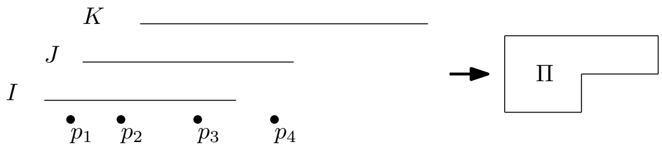

Figure 1 A covering gadget Π, and its schematic representation.

Proof. Follows from the fact that none of the intervals in $\Gamma(X, C)$ that is not a member of the covering gadget Π can discriminate the four points in Π. Moeover, if we do not choose I, then p_3, p_4 are not discriminated. If we do not choose J, p_1, p_2 are not discrimnated. If we do not choose K, p_2, p_3 are not discriminated. ◀

Let us now define the gadgets modeling the clauses and variables of the 3-SAT-2l instance.

▶ **Definition 3.** *Let c_i be a clause of C. The* clause gadget *for c_i, denoted $G_c(c_i)$, is defined by a covering gadget $\Pi(c_i)$ along with two points p_{c_i}, p'_{c_i} placed in $K \setminus \{I \cup J\}$ (see Fig. 2).*

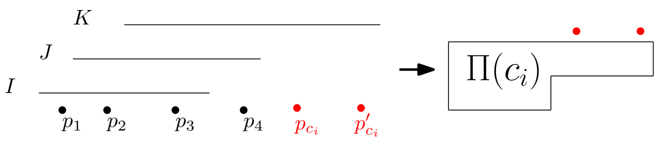

Figure 2 A covering gadget $G_c(c_i)$, and its schematic representation.

The idea behind the clause gadget is that some interval that ends between points p_{c_i}, p'_{c_i} will have to be taken in the solution, so that this pair gets discriminated.

▶ **Definition 4.** *Let x_j be a variable of X. The* variable gadget *for x_j, denoted $G_v(x_j)$, is defined by a covering gadget $\Pi(x_j)$, and five points $p^1_{x_j}, \ldots, p^5_{x_j}$ placed consecutively in $K \setminus \{I \cup J\}$. It also contains six intervals $I^0_{x_j}$, $I^1_{x_j}$, $I^2_{x_j}$, $I^0_{\bar{x}_j}$, $I^1_{\bar{x}_j}$, $I^2_{\bar{x}_j}$, as in Fig. 3. The right end points will depend on the formula.*

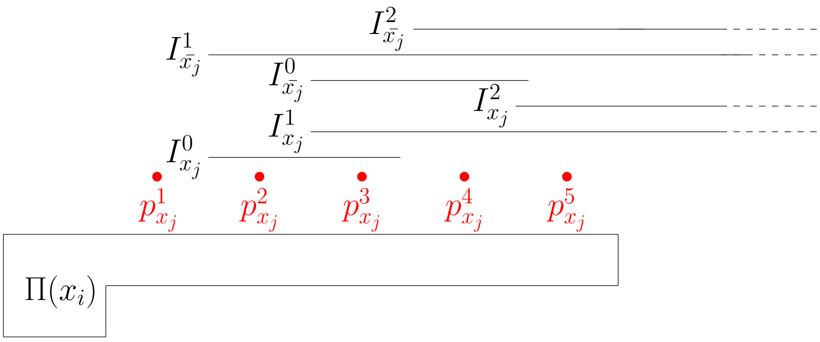

Figure 3 A variable gadget $G_v(x_j)$.

In a variable gadget $G_v(x_j)$, the intervals $I^1_{x_j}$ and $I^2_{x_j}$ represent the occurrences of literal x_j, while $I^1_{\bar{x}_j}$ and $I^2_{\bar{x}_j}$ represent the occurrences of $\bar{x}_j$. The right end points of each of these four intervals will be in the clause gadget of the clause that the occurrence of the literal belongs to. Thus, $\Gamma(X, C)$ is constructed as follows. Note that we can assume that every literal appears in at least one clause (otherwise, we can fix the truth value of the variable and obtain a smaller equivalent instance).

- For each variable $x_i \in X$, $\Gamma(X,C)$ contains a variable gadget $G_v(x_i)$.
- The gadgets $G_v(x_1), G_v(x_2), \ldots, G_v(x_n)$ are positioned consecutively, in this order, without overlap.
- For each clause $c_j \in C$, $\Gamma(X,C)$ contains a clause gadget $G_c(c_j)$.
- The gadgets $G_c(c_1), G_c(c_2), \ldots, G_c(c_m)$ are positioned consecutively, in this order, after the variable gadgets, without overlap.
- For every variable x_i, assume x_i appears in clauses c_{i_1} and c_{i_2}, and $\bar{x_i}$ appears in c_{i_3} and c_{i_4} (possibly $i_1 = i_2$ or $i_3 = i_4$). Then, we extend interval $I^1_{x_i}$ so that it ends between $p_{c_{i_1}}$ and $p'_{c_{i_1}}$; $I^2_{x_i}$ ends between $p_{c_{i_2}}$ and $p'_{c_{i_2}}$; $I^1_{\bar{x_i}}$ ends between $p_{c_{i_3}}$ and $p'_{c_{i_3}}$; $I^2_{\bar{x_i}}$ ends between $p_{c_{i_4}}$ and $p'_{c_{i_4}}$.

Let $\mathcal{C}^\Pi$ be the union of the disc-codes (i.e. all intervals of type I, J, K, by Observation 2) of all covering gadgets. Observe that $\mathcal{C}^\Pi$ discriminates the points p_1, p_2, p_3, p_4 in each covering gadget Π, and any point covered by K from any other point not covered by K. It follows that all point-pairs are discriminated by $\mathcal{C}^\Pi$, except the following critical ones:

- the pairs among the five points $p^1_{x_i}, \ldots, p^5_{x_i}$ of each variable gadget $G_v(x_i)$, and
- the point pair $\{p_{c_j}, p'_{c_j}\}$ of each clause gadget $G_c(c_j)$.

▶ **Theorem 5** ($\star$). *Discrete-G-Min-Disc-Code in 1D is NP-complete.*

Proof (sketch). We prove that (X,C) is satisfiable if and only if $\Gamma(X,C)$ has a disc-code of size $6n+3m$. In both parts of the proof, we will consider the set $\mathcal{C}^\Pi$ defined above. Each variable gadget and clause gadget contains one covering gadget. Thus, $|\mathcal{C}^\Pi| = 3(n+m)$.

Consider first some satisfying truth assignment of X. We build a solution set $\mathcal{C}$ as follows. First, we put all intervals of $\mathcal{C}^\Pi$ in $\mathcal{C}$. Then, for each variable x_i, if x_i is true, we add intervals $I^0_{x_i}$, $I^1_{x_i}$ and $I^2_{x_i}$ to $\mathcal{C}$. Otherwise, we add intervals $I^0_{\bar{x_i}}$, $I^1_{\bar{x_i}}$ and $I^2_{\bar{x_i}}$ to $\mathcal{C}$. Notice that $|\mathcal{C}| = 6n+3m$. As observed before, it suffices to show that $\mathcal{C}$ discriminates the point-pair $\{p_{c_j}, p'_{c_j}\}$ of each clause gadget $G_c(c_j)$, and the points $p^1_{x_i}, \ldots, p^5_{x_i}$ of each variable gadget $G_v(x_i)$. (All other pairs are discriminated by $\mathcal{C}^\Pi$.)

For the converse, assume that $\mathcal{C}$ is a discriminating code of $\Gamma(X,C)$ of size $6n+3m$. By Observation 2, $\mathcal{C}^\Pi \subseteq \mathcal{C}$. Thus there are $3n$ intervals of $\mathcal{C}$ that are not in $\mathcal{C}^\Pi$, and we show that each variable gadget contains exactly three. Then, we show how to construct a truth assignment of (X,C). Notice that at least one of $I^0_{x_i}$ and $I^0_{\bar{x_i}}$ must belong to $\mathcal{C}$, otherwise some points of $G_v(x_i)$ cannot be discriminated. If $I^0_{x_i} \in \mathcal{C}$, but $I^0_{\bar{x_i}} \notin C$, then necessarily $I^1_{x_i} \in \mathcal{C}$ and $I^2_{x_i} \in \mathcal{C}$, and we can set x_i to true. Similarly, if $I^0_{\bar{x_i}} \in \mathcal{C}$ but $I^0_{x_i} \notin \mathcal{C}$, we set it to false. If both are in $\mathcal{C}$, we choose the truth value depending on which third interval of the gagdet belongs to $\mathcal{C}$. The properties of the gadget then ensure that this assignment is satisfying. ◀

2.2 A 2-approximation algorithm for the general 1D case

We next use the classic algorithm solving the edge-cover problem of an undirected graph to design a 2-factor approximation algorithm for Discrete-G-Min-Disc-Code in 1D.

Edge-Cover **Input:** An undirected graph $G = (V,E)$. **Output:** A subset $E' \subseteq E$ such that every vertex is incident to at least one edge of E'.

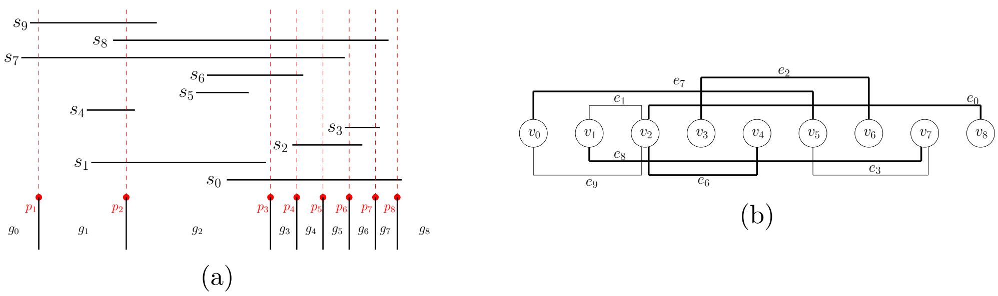

Figure 4 (a) An instance (P, S), (b) corresponding graph $G = (V, E)$ with MEC edges highlighted. Note that s_4 and s_5 are redundant intervals.

We create a graph $G = (V, E)$, where $V = \{v_0, v_1, \ldots, v_n\}$ corresponds to the set $\mathcal{G}$ of gaps. For each interval $s_i = (a_i, b_i) \in S$, we create an edge $e_i = (v_\alpha, v_\beta) \in E$ if $a_i \in g_\alpha$ and $b_i \in g_\beta$. See Figure 4 for an example. As we have removed useless and redundant intervals (as defined at the beginning of Section 2), there are no loops and multiple edges in G. Thus, $|V| = n + 1$ and $|E| \leq m$. The *minimum edge-cover* (MEC) E' consists of (i) the edges of a maximum matching in G, and (ii) for each unmatched vertex (if exists), any arbitrary edge incident to that vertex [12]. It can be computed in time $O(\min(n^2, m\sqrt{n}))$ [19].

Let S' be the set of intervals corresponding to the edges of E'. Clearly, S' discriminates all *consecutive* point-pairs of P, since for each gap g_i, there is an interval with an endpoint in g_i. Moreover, S' is an optimal set of intervals discriminating all consecutive point-pairs. Thus, any solution to DISCRETE-G-MIN-DISC-CODE for (P, S) has size at least $|S'|$, since any such solution should in particular discriminate consecutive point-pairs.

▶ **Lemma 6** ($\star$). *The points in P can be classified into sets $U, Q_0, \ldots, Q_k$ using the set S', with the following properties.*

- *A subset $U \subseteq P$ will receive unique codes by S',*
- *A subset $Q_0 \subset P$ may not be covered by the intervals of S', and hence they will not receive any code. If $|Q_0| > 0$ then the elements in Q_0 are non-consecutive.*
- *Some subsets $Q_1, \ldots, Q_k$ of points (of sizes > 1) of P may each receive the same nonempty code by S'. In that case, the members of each of those subsets are non-consecutive.*

Proof of Lemma 6. Clearly, since S' discriminates all consecutive point-pairs, for any integer i, any two points of Q_i cannot be consecutive. ◀

▶ **Lemma 7.** *Denote by $I(Q_i)$, the interval starting at the first point of Q_i and stopping at the last point of Q_i. Then, for any two distinct sets Q_i and Q_j, either $I(Q_i)$ and $I(Q_j)$ are disjoint, or one of them (say Q_j) is strictly included between two consecutive points of the other (Q_i). In that case, we say that Q_j is* nested *inside Q_i.*

Proof. Suppose that $I(Q_i)$ and $I(Q_j)$ intersect. Recall that all the points in Q_i have the same code C_i by S', and all the points in Q_j have the same code $C_j \neq C_i$ by S'. That is, each interval of S' either contains all points or no point of Q_i and Q_j, respectively, and there is at least one interval I of S' that contains, say, all points of Q_j but no point of Q_i. Then, necessarily, $I(Q_j)$ is included between two consecutive points of Q_i, as claimed. ◀

For a set Q_i of size s, we denote $q_i^1, \ldots, q_i^s$ the points in Q_i. We give a lower bound on $|S'|$.

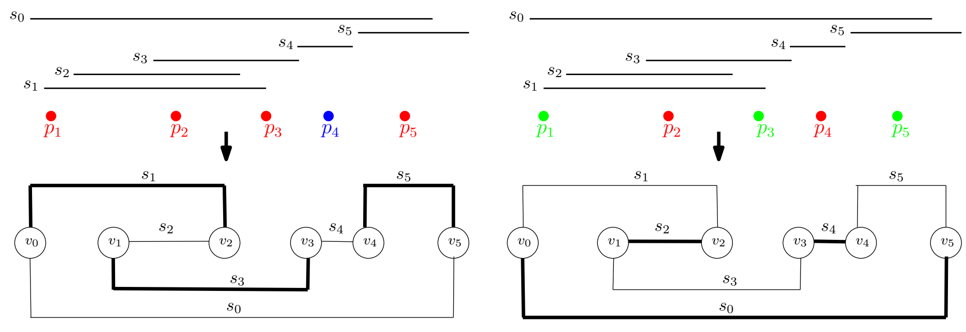

Figure 5 Illustration of Lemma 6 with two different MECs: the points in set U (red), Q_0 (blue) and Q_1 (green).

▶ **Lemma 8** (⋆). *We have* $|S'| \geq \sum_{i=0}^{k}(|Q_i| - 1) + 1$.

Proof. Consider the sets $Q_0, \dots, Q_k$ (possibly $Q_0 = \emptyset$). We will prove that every interval $I(Q_i)$ contains a set S'_i of at least $|Q_i| - 1$ intervals of S' that are included in $I(Q_i)$. Moreover, for every Q_j that is nested inside Q_i, none of the intervals of S'_i are included in $I(Q_j)$.

We proceed by induction on the nested structure of the $I(Q_i)$'s that follows from Lemma 7. As a base case, assume that $I(Q_i)$ has no interval $I(Q_j)$ nested inside. Since by Lemma 6, the points of Q_i are non-consecutive inside P, between each pair q_i^a, q_i^{a+1} of consecutive points of Q_i, there is at least one point p of P. By definition of Q_i, p is discriminated from all points of Q_i by S'. Hence, there is an interval of S' that lies completely between q_i^a and q_i^{a+1}: add it to S'_i. Since there are $|Q_i| - 1$ such consecutive pairs, $|S'_i| \geq |Q_i| - 1$: the base case is proved.

Next, assume by induction that the claim is true for all the intervals Q_j that are nested inside Q_i. Consider a point q_i^a of Q_i that is not the last point of Q_i. Again, between q_i^a and q_i^{a+1}, there is a point of P. Let p be the point of P that comes just after q_i^a. The set S' discriminates the two consecutive points q_i^a and p. However, there cannot be an interval of S' covering q_i^a and ending between q_i^a and p, otherwise it would also discriminate q_i^a and q_i^{a+1}. Thus, there must be an interval I of S' that starts between q_i^a and p. Notice that I is not included in any $I(Q_j)$, for Q_j nested inside Q_i. Thus, we can add I to S'_i. Repeating this for all points of Q_i except the last one, we obtain that $|S'_i| \geq |Q_i| - 1$, as claimed.

We have thus proved that there are at least $\sum_{i=0}^{k}(|Q_i| - 1)$ distinct intervals of S', each of them being included in some $I(Q_i)$. But moreoever, there is at least one interval of S' that is not included in any $I(Q_i)$. Indeed, there must be an interval of S' that corresponds to an edge of E' that covers the first gap g_0. This interval has not been counted in the previous argument. Thus, it follows that $|S'| \geq \sum_{i=0}^{k}(|Q_i| - 1) + 1$. ◀

Next, we will choose additional intervals from $S \setminus S'$ to discriminate the points in $\cup_{j=0}^{k} Q_j$, and add them to S'. The resulting set, S'', will form a discriminating code of (P, S). Consider some set $Q_i = \{q_i^1, \dots, q_i^s\}$. We will choose at most $s - 1$ new intervals so that all points in Q_i are discriminated: call this set S''_i. We start with q_i^1, q_i^2, and we select some interval of S that discriminates q_i^1, q_i^2 (since (P, S) can be assumed to be twin-free, such an interval exists) and add it to S''_i. We then proceed by induction: at each step a $(2 \leq a \leq s - 1)$, we assume that the points $q_i^1, \dots, q_i^a$ are discriminated, and we consider q_i^{a+1}. There is at most one point, say q_i^b, among $q_i^1, \dots, q_i^a$ whose code is the same as q_i^{a+1} by S''_i (since by induction $q_i^1, \dots, q_i^a$ all have different codes). We thus find one interval of S that discriminates q_i^{a+1}, q_i^b and add it to S''_i. In the end we have $|S''_i| \leq |Q_i| - 1$.

After repeating this process for every set Q_i, all pairs of points of P are discriminated by $S' \cup \bigcup_{j=0}^{k} S''_j$. Finally, we may have to add one additional interval in order to cover one point of Q_0, that remains uncovered. Let us call S'' the resulting set: this is a discriminating code of (P,S). Moreover, we have added at most $\sum_{j=0}^{k}(|Q_j|-1)+1$ additional intervals to S', to obtain S''. By Lemma 8, we thus have $|S''| \leq |S'| + \sum_{j=0}^{k}(|Q_j|-1)+1 \leq 2|S'|$.

Hence, denoting by OPT the optimal solution size for (P,S), and recalling that $|S'| \leq OPT$, we obtain that $|S''| \leq 2|S'| \leq 2OPT$. Moreover, the construction of S'' from S can be done in linear time. Thus, we have proved the following:

▶ **Theorem 9.** *The proposed algorithm produces a* 2*-factor approximation for* Discrete-G-Min-Disc-Code *in 1D, and runs in time* $O(\min(n^2, m\sqrt{n}))$.

2.3 A PTAS for the 1D unit interval case

The following observation (which was also made in the related setting of identifying codes of unit interval graphs [10, Proposition 5.12]) plays an important role in designing our PTAS.

▶ **Observation 10.** *In an instance* (P,S) *of* Discrete-G-Min-Disc-Code *in 1D, if the objects in* S *are intervals of the same length, then discriminating all the pairs of* consecutive *points in* P *is equivalant to discriminating* all *the pairs of points in* P.

For a given $\epsilon > 0$, we choose $\lceil \frac{n\epsilon}{4} \rceil$ points, namely $q_1, q_2, \ldots, q_{\lceil \frac{n\epsilon}{4} \rceil} \in P$, called the *reference points*, as follows: q_1 is the $\lceil \frac{2}{\epsilon} \rceil$-th point of P from the left, and for each $i = 1, 2, \ldots, \lfloor \frac{n\epsilon}{4} \rfloor$, the number of points in P between every consecutive pair (q_i, q_{i+1}) is $\lceil \frac{4}{\epsilon} \rceil$, including q_i and q_{i+1} (the number of points to the right of $q_{\lceil \frac{n\epsilon}{4} \rceil}$ may be less than $\lceil \frac{2}{\epsilon} \rceil$). For each *reference point* q_i, we choose two intervals $I_i^1, I_i^2 \in S$ such that both I_i^1, I_i^2 contain (span) q_i, and the left (resp. right) endpoint of I_i^1 (resp. I_i^2) have the minimum x-coordinate (resp. maximum x-coordinate) among all intervals in S that span q_i. Observe that all the points in P that lie in the range $G_i = [\ell(I_i^1), r(I_i^2)]$ are *covered*, where $\ell(I_i^1), r(I_i^2)$ are the x-coordinates of the left endpoint of I_i^1 and the right endpoint of I_i^2, respectively. These ranges will be referred to as *group-ranges*. Since the endpoints of the intervals are distinct, the span of a *group-range* is strictly greater than 1. The span of an interval may be defined as the number of points that lie inside it.

We now define a *block* as follows. Observe that the ranges G_i and G_{i+1} may or may not overlap. If several consecutive ranges $G_i, G_{i+1}, \ldots, G_k$ are pairwise overlapping, then the horizontal range $[\ell(I_i^1), r(I_k)]$ forms a block. The region between a pair of consecutive blocks will be referred to as a *free region*. We use $B_1, B_2, \ldots, B_l$ to name the blocks in order, and $F_0, F_1, \ldots, F_l$ to name the free regions (from left to right). The points in each block are covered. Here, the remaining tasks are (i) for each block, choose intervals from S such that consecutive pairs of points in that block are discriminated, and (ii) for each free region, choose intervals from S such that all its points are covered, and the pairs of consecutive points are discriminated. Observe that no interval $I \in S$ can contain both a point in F_i and a point in F_{i+1} since F_i and F_{i+1} are sepatated by the block B_{i+1}. The reason is that if there exists such an interval I, then it will contain the reference point $q_j \in B_{i+1}$ just to the right of F_i[2]. This contradicts the choice of I_j^1 for q_j. Thus, the discriminating code for a free region F_i is disjoint from that of its neighboring free region F_{i+1}. So, we can process the free regions independently.

[2] the reference point of the leftmost group-range G_j of the block B_{i+1}.

Processing of a free region. Let the neighboring group-ranges of a free region F_i be G_a and G_{a+1}, respectively. There are at most $\frac{4}{\epsilon}$ points lying between the reference points of G_a and G_{a+1}. Among these, several points of P to the right (resp. left) of the reference point of G_a (resp. G_{a+1}) are inside *block* B_i (resp. B_{i+1}). Thus, there are at most $\frac{4}{\epsilon}$ points in F_i. We collect all the members in $\mathcal{I}_{F_i} \subseteq S$ that cover at least one point of F_i. Note that, though we have deleted all the redundant intervals of S, there may be several intervals in S with an endpoint lying in a gap inside that free region, and their other endpoint lies in distinct gaps of the neighboring block. There are some blue intervals which are redundant with respect to the points $F_i \cap P$, but are non-redundant with respect to the whole point set P (see Figure 6). However, the number of such intervals is at most $\frac{4}{\epsilon}$ due to the definition of (I_i^1, I_i^2) of the right-most group-range of the neighboring block B_i and left-most group-range of B_{i+1}.

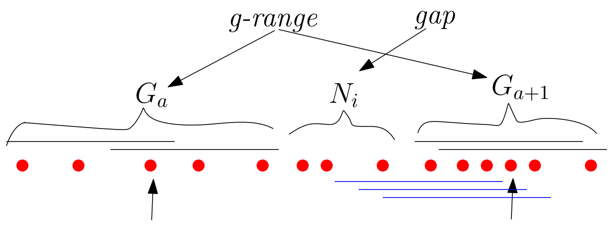

Figure 6 Demonstration of redundant edges in a free region which are non-redundant in the problem instance (P, S).

Thus, we have $|\mathcal{I}_{F_i}| = O(1/\epsilon^2)$. We consider all possible subsets of intervals of $\mathcal{I}_{F_i}$, and test each of them for being a discriminating code for the points in F_i. Let $\mathcal{D}_i$ be all possible different discriminating codes of the points in F_i, with $|\mathcal{D}_i| = 2^{O(1/\epsilon^2)}$ in the worst case.

Processing of a block. Consider a block B_i; its neighboring free regions are F_i and F_{i+1}. Consider two discriminating codes $d \in \mathcal{D}_i$ and $d' \in \mathcal{D}_{i+1}$. As in Section 2.2, we create a graph $G_i = (V_i, E_i)$ whose nodes V_i correspond to the gaps of B_i which are not discriminated by the intervals used in $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$. Each edge $e \in E_i$ corresponds to an interval in S that discriminates pairs of consecutive points corresponding to two different nodes of V_i. Now, we can discriminate each non-discriminated pair of consecutive points in B_i by computing a minimum edge-cover of G_i in $O(|V_i|^2)$ time [19]. As mentioned earlier, all the points in B_i are covered. Thus, the discrimination process for the block B_i is over. We will use $\theta(d, d')$ to denote the size of a minimum edge-cover of B_i using $d \in \mathcal{D}_i$ and $d' \in \mathcal{D}_{i+1}$.

Computing a discriminating code for P. We now create a multipartite directed graph $H = (\mathcal{D}, \mathcal{F})$. Its i-th partite set corresponds to the discriminating codes in $\mathcal{D}_i$, and $\mathcal{D} = \cup_{i=0}^{l} \mathcal{D}_i$. Each node $d \in \mathcal{D}$ has its weight equal to the size of the discriminating code d. A directed edge $(d, d') \in \mathcal{F}$ connects two nodes d and d' of two adjacent partite sets, say $d \in \mathcal{D}_i$ and $d' \in \mathcal{D}_{i+1}$, and has its weight equal to $\theta(d, d')$. For every pair of partite sets $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$, we connect every pair of nodes (d, d') $d \in \mathcal{D}_i$ and $d' \in \mathcal{D}_{i+1}$, where $i = 0, 1, \ldots, l-1$. Every node of $\mathcal{D}_0$ is connected to a node s with weight 0, and every node of $\mathcal{D}_l$ is connected to a node t with weight 0.

▶ **Lemma 11** (⋆). *The shortest weight of an s-t path*[3] *in H is a lower bound on the size of the optimum discriminating code for (P, S).*

[3] The weight of a path is equal to the sum of costs of all the vertices and edges on the path.

Let S' denote the set of intervals of S in a shortest s-t path in H. The intervals in S' may not form a discriminating code for P, as the points in a block may not all be covered. However, the additional intervals $\{(I_i^1, I_i^2), i = 1, 2, \ldots, \lceil \frac{n\epsilon}{2} \rceil\}$ ensure the covering of the points in all blocks $B_i, i = 1, 2, \ldots, \lceil \frac{n\epsilon}{2} \rceil$. Thus, $SOL = S' \cup \{(I_i^1, I_i^2), i = 1, 2, \ldots, \lceil \frac{n\epsilon}{4} \rceil\}$ is a discriminating code for (P, S). Moreover, the optimum size of the discriminating code, denoted OPT, satisfies $OPT \geq \lceil \frac{n+1}{2} \rceil$ due to the fact that we have $(n + 1)$ gaps, and each interval in S covers exactly 2 gaps. This fact, along with Lemma 11 implies:

▶ **Lemma 12** ($\star$). *$|SOL| \leq (1 + \epsilon)OPT$.*

Proof. By Lemma 11, $|\mathcal{I}'| \leq \mathcal{I}_{opt}$. The number of extra intervals to cover the *blocks* is $\frac{n\epsilon}{2}$. Again, $\frac{n}{2} \leq EC(P) \leq \mathcal{I}_{opt}$, where $EC(P)$ is the size of minimum edge-cover of the graph G created with the points in P and the intervals in $\mathcal{I}$. Thus, $|SOL| \leq (1 + \epsilon)\mathcal{I}_{opt}$. ◀

The number of possible discriminating codes in a free region is $2^{O(1/\epsilon^2)}$. Thus, we may have at most $2^{O(1/\epsilon^2)}$ edges between a pair of consecutive sets $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$. As the computation of the cost of an edge between the sets $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$ invokes the edge-cover algorithm of an undirected graph, it needs $O(|B_i|^2)$ time [19]. Thus, the total running time of the algorithm is $A + B$, where A is the time of generating the edge costs, and B is the time for computing a shortest path of H. We have $A \leq \sum_{i=1}^{\lceil \frac{n\epsilon}{4} \rceil} 2^{O(1/\epsilon^2)} \times O(|B_i|^2)$. As the B_i's are mutually disjoint, we get $A = O(n^2 \times 2^{O(1/\epsilon^2)})$. Moreover, $B = O(|\mathcal{F}|) = O(\frac{n}{\epsilon} \times 2^{O(1/\epsilon^2)})$ [25].

Moreover, we can easily reduce Continuous-G-Min-Disc-Code to Discrete-G-Min-Disc-Code by first computing the $O(n^2)$ possible non-redundant unit intervals. Thus:

▶ **Theorem 13.** *Discrete-G-Min-Disc-Code and Continuous-G-Min-Disc-Code in 1D for unit interval objects have a PTAS: for every $\epsilon > 0$, they admit a $(1 + \epsilon)$-factor approximation algorithm with time complexity $2^{O(1/\epsilon^2)} n^2$.*

3 The two-dimensional case: axis-parallel unit squares

In [13], it was shown that Continuous-G-Min-Disc-Code for bounded-radius disks is NP-complete. The same proof technique, a reduction from the NP-complete P_3-Partition-Grid problem [27], can be adapted to show the following.

▶ **Theorem 14** ($\star$). *Continuous-G-Min-Disc-Code and Discrete-G-Min-Disc-Code for axis-parallel unit squares in 2D are NP-complete.*

3.1 A $(4 + \epsilon)$-approximation algorithm for the continuous problem

We formulate our algorithm by extending the ideas for the 1D case in Section 2.2. Here, our goal is to choose a set Q of points in $\mathbb{R}^2$ of minimum cardinality such that every point of P is covered by at least one axis-parallel unit square centered at Q, and for every pair of points $p_i, p_j \in P$ $(i \neq j)$, there exists at least one square whose boundary intersects the interior of the segment $\overline{p_i p_j}$ exactly once. We define the set of line segments $L(P) = \{\overline{p_i p_j}$ for all $p_i, p_j \in P, i \neq j\}$, where $\overline{p_i p_j}$ is the line segment joining p_i and p_j. We will thus use the following problem:

Segment-Stabbing
Input: A set L of segments in 2D.
Output: A minimum-size set S of axis-parallel unit squares in 2D such that each segment is intersected exactly once by some square of S.

In fact, SEGMENT-STABBING for the input $L(P)$ is equivalent to the TEST COVER problem for P using axis-parallel unit squares as tests. As in the edge-cover formulation of DISCRETE-G-MIN-DISC-CODE in 1D from Section 2.2, here also a feasible solution of SEGMENT-STABBING ensures that the two endpoints of each line segment of $L(P)$ are discriminated, but one point may remain uncovered. Thus, we have the following:

▶ **Observation 15.** *For a feasible solution Φ of* SEGMENT-STABBING*,*

(a) *Φ discriminates every point-pair in P and*

(b) *at most one point is not covered by any square in Φ.*

In order to discriminate the two endpoints of a member $\ell = [a, b] \in L(P)$, we need to consider the two cases: $\lambda(\ell) \geq 1$ and $\lambda(\ell) < 1$, where $\lambda(\ell)$ denotes the length of ℓ. In the former case, if a center is chosen in any one of the unit squares centered at a and b, the segment ℓ is stabbed. However, more generally in the second case, to stab ℓ, we need to choose a center in the region $(D(a) \setminus D(b)) \cup (D(b) \setminus D(a))$, where $D(q)$ is the axis parallel unit square centered at q (see Figure 7). Let us denote the set of all such objects corresponding to the members in $L(P)$ as $\mathcal{O}$. We now need to solve the HITTING SET problem, where the objective is to choose a minimum number of center points in $\mathbb{R}^2$, such that each object in $\mathcal{O}$ contains at least one of those chosen points. We solve this problem using a technique followed in [1] for covering a set of segments using unit squares.

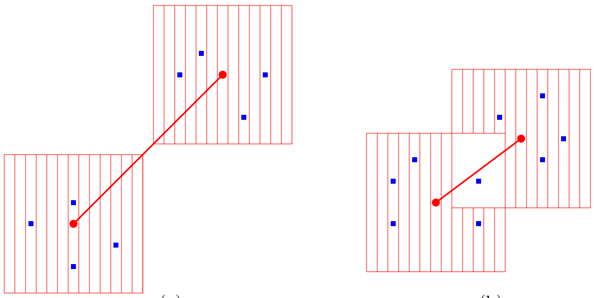

Figure 7 Object for segment $\ell = [a, b]$, where (a) $\lambda(\ell) \geq 1$ and (b) $\lambda(\ell) < 1$.

The Seg-HIT problem. Consider the *arrangement* [8] $\mathcal{A}$ of the objects in $\mathcal{O}$. Create a set Q of points by choosing one point in each cell of $\mathcal{A}$. A square centered at a point q inside a cell $A \in \mathcal{A}$ will stab all the segments whose corresponding objects have common intersection A. For each point $q \in Q$, we use an indicator variable x_q. Thus, we have an integer linear programming (ILP) problem, whose objective function is:

$$
\begin{aligned}
&Z_0 : \min \sum_{\alpha=1}^{|Q|} x_\alpha, \\
&\text{subject to} \quad \sigma_1(\ell) + \sigma_2(\ell) \geq 1 \quad \text{for all segments} \quad \ell = [a, b] \in L(P), \\
&\quad \text{where} \quad \sigma_1(\ell) = \sum_{q_\alpha \in Q \cap (D(a) \setminus D(b))} x_\alpha, \\
&\quad \text{and} \quad \sigma_2(\ell) = \sum_{q_\alpha \in Q \cap (D(b) \setminus D(a))} x_\alpha, \\
&\quad \text{and} \quad x_\alpha \in \{0, 1\} \text{ for all points } q_\alpha \in Q. \qquad (1)
\end{aligned}
$$

As the ILP is NP-hard [23], we relax the integrality condition of the variables x_q for all $q \in Q$ from Z_0, and solve the corresponding LP problem $\overline{Z}_0$:

$$\overline{Z}_0 : \min \sum_{\alpha=1}^{|Q|} x_\alpha$$
$$\text{subject to } \sigma_1(\ell) + \sigma_2(\ell) \geq 1 \ \forall \ \ell = [a,b] \in L(P),$$
$$\text{and} \quad 0 \leq x_\alpha \leq 1 \ \forall \ q_\alpha \in Q \tag{2}$$

in polynomial time.

Observe that for each constraint, at least one of $\sigma_1(\ell)$ or $\sigma_2(\ell)$ will be greater than $\frac{1}{2}$. We choose either $(D(a) \setminus D(b))$ or $(D(b) \setminus D(a))$ or both in a set $\mathcal{O}_1$ depending on whether $\sigma_1(\ell) >$ or $=$ or $< \sigma_2(\ell)$, and form an ILP Z_1 for the hitting set problem with the objects in $\mathcal{O}_1$ as stated above. Observe that, if $\overline{x}$ is an optimum solution for $\overline{Z}_0$, then $2\overline{x}$ is a feasible solution of $\overline{Z}_1$. Denoting by OPT_θ and $\overline{OPT}_\theta$ as the optimum solutions of the problem Z_θ and $\overline{Z}_\theta$ respectively, we have

$$\overline{OPT}_1 \leq 2 \sum_{\alpha=1}^{|Q|} x_\alpha = 2\overline{OPT}_0 \leq 2OPT_0, \tag{3}$$

The L-HIT problem. Now, we solve Z_1, where each object is either a unit square or an L-shape object whose length and width of the outer side are 1. Such an object is the union of two rectangles of type A and B, where the one of type A has height 1 and width ≤ 1, and the one of type B has width 1 and height ≤ 1 (see Figure 8).

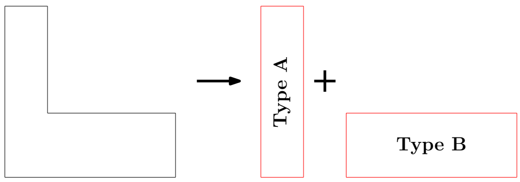

Figure 8 L-shaped object which is the union of a type A and a type B object.

While solving Z_1, for each constraint, any (or both) of these cases must happen: (a) the sum of variables whose corresponding points lie in a type A rectangle is ≥ 0.5, (b) the sum of variables whose corresponding points lie in a type B rectangle is ≥ 0.5. We accumulate all type A (resp. B) rectangles for which condition (a) (resp. (b)) is satisfied in set $\mathcal{A}$ (resp. $\mathcal{B}$).

The ILP formulation $Z_2^{\mathcal{A}}$ of the hitting set problem for the rectangles in $\mathcal{A}$ can be done as follows. Consider the arrangement of the rectangles in $\mathcal{A}$. In each cell of the arrangement, we can choose a point to form a set of points $Q_{\mathcal{A}}$ considering all the cells in $\mathcal{A}$. Now,

$$Z_2^{\mathcal{A}} : \min \sum_{q \in Q_{\mathcal{A}}} x_q,$$
$$\text{subject to } \sum_{q \in A_\alpha} x_q \geq 1 \ ,$$
$$\text{for each rectangle} \quad A_\alpha \in \mathcal{A} \ ,$$
$$\text{and} \quad x_q \in \{0,1\}, \forall \ q \in Q_{\mathcal{A}}. \tag{4}$$

Similarly, we can have an ILP formulation $Z_2^{\mathcal{B}}$ for the hitting set problem of the rectangles in $\mathcal{B}$. The corresponding LP problems are $\overline{Z}_2^{\mathcal{A}}$ and $\overline{Z}_2^{\mathcal{B}}$ respectively. Following the notations introduced earlier, we have

$$\overline{OPT}_2^{\mathcal{A}} + \overline{OPT}_2^{\mathcal{B}} \leq OPT_2^{\mathcal{A}} + OPT_2^{\mathcal{B}} \leq 2\overline{OPT}_1. \quad (5)$$

The right-hand inequality follows from the fact that if we multiply the solution of the variables in $\overline{OPT}_1$ by 2, and then round the fractional part of each non-zero x_α, we can get a feasible solution for $Z_2^{\mathcal{A}}$ and $Z_2^{\mathcal{B}}$.

The U-HIT problem. We now compute the optimum solution $\overline{OPT}_2^{\mathcal{A}}$ of $\overline{Z}_2^{\mathcal{A}}$ and $\overline{OPT}_2^{\mathcal{B}}$ of $\overline{Z}_2^{\mathcal{B}}$, where all rectangles in $\mathcal{A}$ are of unit height and all rectangles in $\mathcal{B}$ are of unit width. Mustafa and Ray [21] proposed a PTAS for the U-HIT problem that runs in $O(mn^{\frac{1}{\epsilon^2}})$ time, where n and m are the number of points and the number of unit-height rectangles.

Equations 3 and 5 and the PTAS for U-HIT lead to the following:

▶ **Lemma 16.** *For a given set of line segments L, the aforesaid algorithm computes a $(4+\epsilon')$-factor approximation for* Segment-Stabbing*, for every fixed $\epsilon' > 0$.*

After solving Segment-Stabbing, by Observation 15, at most one point in P may not be covered. Thus, we may add at most one extra square to cover that point, and obtain a solution of size at most $(4+\epsilon')OPT + 1$, which implies:

▶ **Theorem 17.** *Continuous-G-Min-Disc-Code for axis-parallel unit squares in 2D has a polynomial-time $(4+\epsilon)$-factor approximation algorithm, for every fixed $\epsilon > 0$.*

Proof. It remains only to show that having a solution of size at most $(4+\epsilon')OPT + 1$ gives a $(4+\epsilon)$-approximation, for every fixed $\epsilon > 0$. To see this, note that $OPT \geq \log_2(n+1)$ (where n is the number of points), since every point is assigned a distinct nonempty subset of the solution SOL, and there can be at most $2^{|SOL|} - 1$ such subsets. The solution of size $(4+\epsilon')OPT + 1$ gives an approximation factor of $4 + \epsilon' + \frac{1}{OPT}$ which is thus at most $4 + \epsilon' + \frac{1}{\log_2(n+1)}$. Thus, if $\epsilon' + \frac{1}{\log_2(n+1)} \leq \epsilon$, we are done. Otherwise, $n \leq 2^{1/\epsilon}$ and hence we can solve the problem by brute-force in constant time (since ϵ is fixed). ◀

3.2 A $(32+\epsilon)$-approximation algorithm for the discrete problem

As for Continuous-G-Min-Disc-Code (Section 3), we reduce Discrete-G-Min-Disc-Code to a special version of Hitting Set, where a set $\mathcal{O}$ of unit height rectangles and a set Q of points are given. The set Q contains the centers of the squares in S, and the objective is to find a minimum cardinality subset of Q that hits all the objects in $\mathcal{O}$. Thus, using an α-factor approximation algorithm for the discrete version of this hitting set problem, we obtain a 4α-factor approximation algorithm for the Discrete-G-Min-Disc-Code.

▶ **Theorem 18** ($\star$). *Discrete-G-Min-Disc-Code for axis-parallel unit squares in 2D has a polynomial-time $(32+\epsilon)$-factor approximation algorithm, for every fixed $\epsilon > 0$..*

4 Concluding remarks and open problems

We have seen that Discrete-G-Min-Disc-Code is NP-complete, even in 1D. This is in contrast to most covering problems and to Continuous-G-Min-Disc-Code, which are polynomial-time solvable in 1D [13, 16]. We believe that our simple reduction can be adapted to the graph problem Min-ID-Code on interval graphs, proved to be NP-complete in [11], but via a much more complex reduction. We also proposed a 2-factor approximation algorithm for the Discrete-G-Min-Disc-Code problem in 1D, and a PTAS for a special case where each interval in the set S is of unit length. It seems challenging to determine whether Discrete-G-Min-Disc-Code in 1D becomes polynomial-time for unit intervals. As noted in [13], this would be related to Min-ID-Code on *unit* interval graphs, which also remains unsolved [11]. In fact, it also seems to be unknown whether Continuous-G-Min-Disc-Code in 1D remains polynomial-time solvable with this restriction.

References

1 Ankush Acharyya, Subhas C. Nandy, Supantha Pandit, and Sasanka Roy. Covering segments with unit squares. *Comput. Geom.*, 79:1–13, 2019.

2 Kaustav Basu, Sanjana Dey, Subhas C. Nandy, and Arunabha Sen. Sensor networks for structural health monitoring of critical infrastructures using identifying codes. *15th Int. Conf. on the Design of Reliable Communication Networks (DRCN)*, pages 43–50, 2019.

3 Ralph P. Boland and Jorge Urrutia. Separating collections of points in euclidean spaces. *Inform. Process. Lett.*, 53:177–183, 1995.

4 Nicolas Bousquet, Aurélie Lagoutte, Zhentao Li, Aline Parreau, and Stéphan Thomassé. Identifying codes in hereditary classes of graphs and vc-dimension. *SIAM J. Discrete Math.*, 29:2047–2064, 2015.

5 Emmanuel Charbit, Irène Charon, Gérard D. Cohen, and Olivier Hudry. Discriminating codes in bipartite graphs. *Electronic Notes in Discrete Mathematics*, 26:29–35, 2006.

6 Irène Charon, Gérard D. Cohen, Olivier Hudry, and Antoine Lobstein. Discriminating codes in (bipartite) planar graphs. *Eur. J. Comb.*, 29:1353–1364, 2008.

7 Gruia Călinescu, Adrian Dumitrescu, Howard Karloff, and Peng-Jun Wan. Separating points by axis-parallel lines. *Int. J. Comput. Geom. Appl.*, 15(06):575–590, 2005.

8 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications.* Springer-Verlag TELOS, USA, 2008.

9 Koen M. J. de Bontridder, Bjarni V. Halldórsson, Magnús M. Halldórsson, A. J. Hurkens, Jan Karel Lenstra, R. Ravi, and Leen Stougie. Approximation algorithms for the test cover problem. *Math. Program.*, 98:477–491, 2003.

10 Florent Foucaud. Combinatorial and algorithmic aspects of identifying codes in graphs. data structures and algorithms, phd thesis. *Université Bordeaux 1, France*, 2012. URL: `http://tel.archives-ouvertes.fr/tel-00766138`.

11 Florent Foucaud, George B. Mertzios, Reza Naserasr, Aline Parreau, and Petru Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. II. Algorithms and complexity. *Algorithmica*, 78:914–944, 2017.

12 M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman, first edition edition, 1979.

13 Valentin Gledel and Aline Parreau. Identification of points using disks. *Discrete Math.*, 342:256–269, 2019.

14 Sariel Har-Peled and Mitchell Jones. On separating points by lines. *Discrete and Computational Geometry*, 63:705–730, 2020.

15 Konstantin Kobylkin. Efficient constant factor approximation algorithms for stabbing line segments with equal disks. *CoRR*, abs/1803.08341, 2018. `arXiv:1803.08341`.

16 Datta Krupa R., Aniket Basu Roy, Minati De, and Sathish Govindarajan. Demand hitting and covering of intervals. In *Algorithms and Discrete Applied Mathematics (CALDAM'17)*, pages 267–280, Cham, 2017. Springer International Publishing.

17 Moshe Laifenfeld and Ari Trachtenberg. Identifying codes and covering problems. *IEEE Trans. Information Theory*, 54:3929–3950, 2008.

18 Moshe Laifenfeld, Ari Trachtenberg, Reuven Cohen, and David Starobinski. Joint monitoring and routing in wireless sensor networks using robust identifying codes. *Mob. Netw. Appl.*, 14:415–432, 2009.

19 Silvio Micali and Vijay V. Vazirani. An O(sqrt(|V|) |E|) algorithm for finding maximum matching in general graphs. In *21st Symp. on Foundations of Computer Science*, pages 17–27, 1980.

20 Tobias Müller and Jean-Sébastien Sereni. Identifying and locating-dominating codes in (random) geometric networks. *Comb. Probab. Comput.*, 18(6):925–952, 2009. doi:10.1017/S0963548309990344.

21 Nabil H. Mustafa and Saurabh Ray. Improved results on geometric hitting set problems. *Discrete and Computational Geometry*, 44:883–895, 2010.

22 Subhas C. Nandy, Tetsuo Asano, and Tomohiro Harayama. Shattering a set of objects in 2D. *Discr. Appl. Math.*, 122(1):183–194, 2002.

23 Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

24 Saikat Ray, David Starobinski, Ari Trachtenberg, and Rachanee Ungrangsi. Robust location detection with sensor networks. *IEEE J. Selected Areas Communications*, 22:1016–1025, 2004.

25 Mikkel Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *Journal of the ACM*, 46(3):362–394, 1999.

26 Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, 1984. doi:10.1016/0166-218X(84)90081-7.

27 René van Bevern, Robert Bredereck, Laurent Bulteau, Jiehua Chen, Vincent Froese, Rolf Niedermeier, and Gerhard J. Woeginger. Star partitions of perfect graphs. In *Automata, Languages, and Programming (ICALP'14)*, pages 174–185, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.