Dominating Set of Rectangles Intersecting a Straight Line

Supantha Pandit*

Abstract

We study the dominating set problem using axis-parallel rectangles and unit squares on the plane. These geometric objects are constrained to be intersected by a straight line which makes an angle with the x-axis. For axis-parallel rectangles, we prove that this problem is NP-complete. When the objects are axis-parallel unit square, we give a polynomial time algorithm. For unit squares which touch the straight line at a single point from either side of the straight line, we give an $O(n \log n)$ time algorithm.

Keywords: Dominating set, Straight line, Rectangles, Squares, NP-complete, Inclined line, Diagonal line, Touching a line, Intersecting a line.

1 Introduction

Dominating Set (DS) problem is a fundamental problem and has applications in diverse setting. This problem is defined as follows. Given a set $\mathcal{O}$ of objects, the objective is to find a subset $\mathcal{O}' \subseteq \mathcal{O}$ of objects such that every object in $\mathcal{O}$ is either in $\mathcal{O}'$ or has a non-empty intersection with an object in $\mathcal{O}'$. This problem is known to be NP-complete even with simple geometric objects like squares, disks, etc. There are many applications where minimum dominating set plays a crucial role, one of them being network routing [17]. In this work, we are interested in a special case of the *DS* problem where the given input objects are forced to intersect a given line which makes an angle with the x-axis. We define this problem formally as follows.

> **Dominating Set Problem with Objects Intersecting a Straight Line:** Given a set of objects $\mathcal{O}$ and a straight line L such that the objects are intersecting the line L. The objective is to find a minimum cardinality subset $\mathcal{O}' \subseteq \mathcal{O}$ of objects such that any object in $\mathcal{O}$ is either belongs to $\mathcal{O}'$ or it has a non-empty intersection with an object in $\mathcal{O}'$.

We are looking at this problem when the objects are axis-parallel rectangles and unit squares in the plane. Further, we assume that the rectangles or squares are either ***intersecting*** or ***touching*** L. Here, we assume that L makes an angle 135° with the x-axis. A set of rectangles is intersecting L if all the rectangles have a non-empty intersection with L (see Figure 1(a)). A set of rectangles is touching L if all the rectangles intersect L only at a corner point and this rectangles lie on the same side of L (see Figure 1(b)). Similarly, we define this two types of intersections for unit squares. In this paper, we consider the following three problems.

- **DS-Rec-IL**: Dominating set problem with rectangles intersecting a straight line.
- **DS-Sq-IL**: Dominating set problem with unit squares intersecting a straight line.
- **DS-Sq-TL**: Dominating set problem with unit squares touching a straight line.

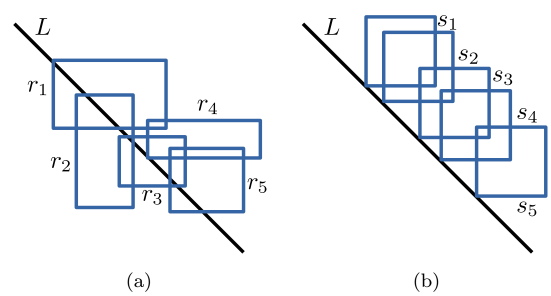

Figure 1: (a) A set of rectangles intersecting a straight line. (b) A set of unit squares touching a straight line.

1.1 Previous Work

The minimum dominating set problem is NP-complete for general graphs [6]. Further, it is $(1-\epsilon)\log n$ hard to approximate this problem for any $\epsilon > 0$ under standard complexity theoretic assumptions [19, 5, 2, 11]. There exists a greedy algorithm which produces an $O(\log n)$ approximation [20] for this problem. Dominating set problem with different classes of graphs like unit disk graphs, growth bounded graphs [10, 17] are

*Indian Statistical Institute, Kolkata, India.
This work is partially supported by Grant No. PDF/2016/002490 for the National Post-doctoral Fellowship of the Science & Engineering Research Board, Department of Science and Technology, Government of India.
pantha.pandit@gmail.com

also studied in the literature. For graphs with polynomially bounded expansion, Har-Peled and Quanrud [9] designed a PTAS using local search algorithm. Gibson and Pirwani [8] designed a PTAS for arbitrary disks. For the intersection graphs of axis-parallel rectangles, ellipses, α-fat objects of constant description complexity, and of convex polygons with r-corners ($r \geq 4$), Erlebach and van Leeuwen [4] proved that the dominating set problem is APX-hard. This implies that, there is no PTAS for these problems unless P=NP. In [14], Marx proved that the problem is W[1]-hard for unit squares, which implies that no efficient-polynomial-time-approximation-scheme (EPTAS) is possible unless FPT = W[1] [15]. Erlebach and van Leeuwen [4] gave a $O(k)$, where $k > 0$, factor approximation factor for homothetic $2k$-regular polygons. They also provided an $O(k^2)$ factor approximation result for homothetic $(2k+1)$-regular polygons. For the homothetic convex polygons where each polygons has k-corners, the best known result is $O(k^4)$-approximation.

Chepoi and Felsner [1] considered the independent set and piercing set problems with rectangles where the rectangles are intersecting an axis-monotone curve. Recently, Correa et al. [3] studied the same problem, however instead of axes-monotone curve they considered a diagonal line. In [16] Their results were extended. Further, in [16], the authors considered the set cover and hitting set problems with other geometric objects as well.

1.2 Our Contributions

We list our contributions as follows.

- DS-REC-IL problem is NP-complete. (Section 2)
- DS-SQ-IL problem can be solved in polynomial time. (Section 3)
- DS-SQ-TL problem can be solved in $O(n \log n)$ time. (Section 4)

1.3 Prerequisites

In this section, we provide some definitions and prerequisites that are used in the subsequent sections. We define **3-SAT** problem as follows. Given a 3-CNF formula F with n variables $x_1, x_2, \ldots, x_n$ and m clauses $C_1, C_2, \ldots, C_m$, where each clause contains exactly 3 literals, the goal is to find a truth assignment to the variables such that F is satisfied. This problem is known to be NP-complete [7]. We now embed the 3-CNF formula F in the plane as follows. For each variable or clause take a vertex in the plane. A literal is present in a clause iff their is an edge from the corresponding variable to that clause. The goal is now to find a satisfying assignment of F. This is **planar 3-SAT** (**P-3-SAT**) problem and Lichtenstein [13] proved that this problem is NP-complete. A further variation of P-3-SAT problem is the **rectilinear planar 3-SAT** (**R-P-3-SAT**) problem which is defined as follows. For each variable or clause we take a horizontal line segment. The variable segments are placed on a horizontal line and clause segments are connected to these variable segments either from above or below by vertical line segments called ***connections*** such that none of these line segments and connections intersect. The goal is to find a satisfying assignment of F. See Figure 2 for an instance of R-P-3-SAT problem. Knuth and Raghunathan [12] proved that R-P-3-SAT problem is NP-complete. Observe that the variable segments are ordered in the increasing x direction. Let $C_t = (x_i \vee \overline{x}_j \vee x_k)$ be a clause where x_i, x_j, x_k are in increasing order. Then we say that, x_i is the ***left*** variable, x_j is the ***middle*** variable, x_i is the ***right*** variable.

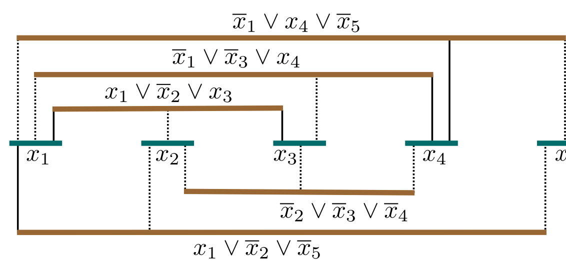

Figure 2: An instance of R-P-3-SAT problem. Solid (resp. dotted) clause vertical segments represent that the variable is positively (resp. negatively) present in the corresponding clauses.

Let us now consider the graph G given in Figure 3. The following claim can be easily proved.

Claim 1 *There are exactly two optimal dominating sets,* $D_0 = \{v_4, v_8, \ldots, v_{8\tau}\}$ *and* $D_1 = \{v_2, v_6, \ldots, v_{8\tau-2}\}$ *of vertices each with cost exactly* 2τ *for graph* G.

Proof. We already know that D_0 and D_1 are dominating sets. Thus the size of a minimum dominating set is atmost 2τ. In a triangle, e.g., vertices v_2, v_3, v_4, to dominate v_3 we should choose one of the vertices v_2, v_3, and v_4. Since there are 2τ such triangles and they are separated by a degree 2 vertex, the size of the minimum dominating set is at least 2τ and thus D_0 and D_1 are minimum dominating sets. Around a degree 2 vertex with non-adjacent neighbours, e.g., vertices v_4, v_5, v_6, we should choose one of the vertices v_4, v_5, and v_6. This means that we cannot choose any degree 2 vertex in a minimum dominating set, and D_0 and D_1 are the only minimum dominating sets. $\square$

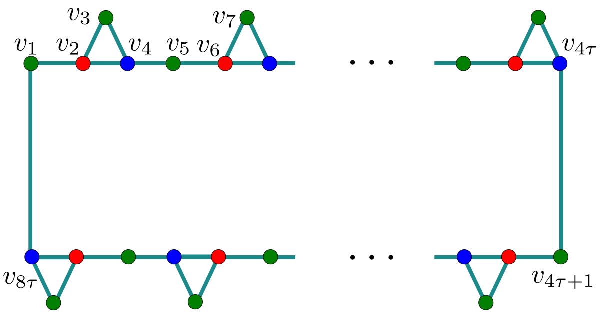

Figure 3: The graph G.

2 Intersecting Rectangles

In this section, we prove that DS-REC-IL problem is NP-complete by giving a reduction from the R-P-3-SAT problem. We first modify R-P-3-SAT problem as follows. Instead of placing the variables on a horizontal line, place them on a diagonal line. Modify the clause vertical connections as follows (see Figure 4). For clauses which connect to the variables from above, remove the clause vertical connection for its left variable and directly connect the clause horizontal segments to the corresponding variables. To distinguish between the negative and positive connections we assume that at the meeting point of this clause horizontal segment and variable segment there is a spare vertical connections. Similar construction can be done for clauses which connect to the variables from below. Now we describe the reduction as follows.

Reduction: Given an R-P-3-SAT instance, we denote α to be the maximum number of clause vertical segments that connect to a single variable segment via connections either from above or below. For each variable x_i, we take 8α rectangles (4 rectangles are considered for each clause vertical connection) $R_i = \{r^i_1, r^i_2, \ldots, r^i_{8\alpha}\}$ as shown in Figure 5. The 4α rectangles $\{r^i_1, r^i_2, \ldots, r^i_{4\alpha}\}$ are above and the 4α rectangles $\{r^i_{4\alpha+1}, r^i_{4\alpha+2}, \ldots, r^i_{8\alpha}\}$ are below the line L. Note that, here we encode the graph in Figure 3 as a variable gadget of DS-REC-IL with $\tau = \alpha$ where vertices represent the rectangles and there is an edge between two vertices if the two rectangles corresponding to these two vertices intersect. Therefore, by Claim 1 we conclude that for each variable gadget there are exactly two optimal dominating set of rectangles $R^1_i = \{r^i_2, r^i_6, \ldots, r^i_{8\alpha-2}\}$ and $R^0_i = \{r^i_4, r^i_8, \ldots, r^i_{8\alpha}\}$ each with cost 2α. The rest of

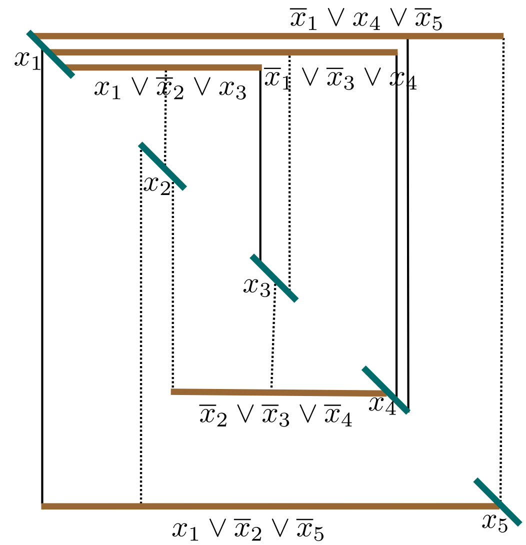

Figure 4: Modified R-P-3-SAT problem instance of the R-P-3-SAT problem instance in Figure 2.

the construction for the clauses connecting to the variables from above is similar for clauses connecting to the variables from below. Therefore, here we only describe the construction for clauses connecting to the variables from above.

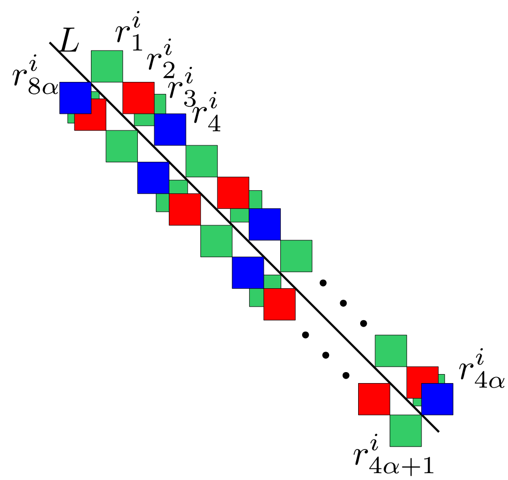

Figure 5: Structure of a variable gadget.

For each clause C_t, we take a thin rectangles r^t (see Figure 6). The bottom boundary of r^t are on the horizontal segment of C_t. We now describe how the rectangle r^t interact with the variable rectangles.

For each variable x_i, $1 \leq i \leq n$, sort the vertical connections from left to right which connect to x_i from clauses connecting from above. Let clause C_t connects to x_i through the l_j-th connection, then we say that C_t is the l_j-th clause for variable x_i.

Let us assume that, C_t contains three variables x_i, x_j, and x_k in this order.

- Here x_i is a left variable in the clause C_t and C_t is the l_1-th clause for x_i. If x_i occurs as a positive literal in C_t, then r^t will intersect with the rectangle $r^i_{4l_1+4}$ only. Otherwise, r^t will intersect with the rectangle $r^i_{4l_1+2}$ only.
- Here x_j is a middle variable in the clause C_t and C_t is the l_2-th clause for x_j. If x_j occurs as a positive literal in C_t, then we extend the rectangle $r^j_{4l_2+4}$ upward such that it will intersect with the rectangle r^t. Otherwise, extend the rectangle $r^j_{4l_2+2}$ upward.
- Here x_k is a right variable in the clause C_t and C_t is the l_3-th clause for x_k. If x_k occurs as a positive literal in C_t, then we extend the rectangle $r^k_{4l_3+4}$ upward such that it will intersect with the rectangle r^t. Otherwise, extend the rectangle $r^k_{4l_3+2}$ upward.

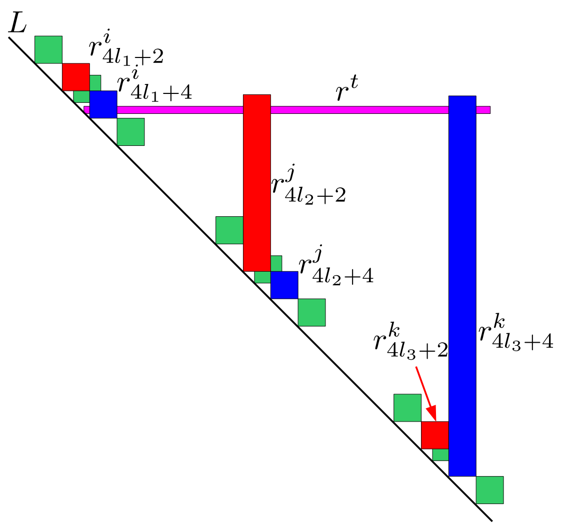

Figure 6: Clause gadget for the clause $C_t = (x_i \vee \overline{x}_j \vee x_k)$ and connection with the variable gadgets of x_i, x_j, x_k.

See Figure 6 for the above construction. Thus, from an instance F of the R-P-3-SAT problem, we created an instance $\mathscr{D}$ of the DS-Rec-IL problem. It is observe that the number of rectangles in $\mathscr{D}$ are $8\alpha n + m$ which is polynomial with respect to the number of variables n and clauses m of the formula F. Hence, this construction can be performed in polynomial time. The correctness of the above construction is shown in following lemma.

Lemma 1 *Formula F is satisfiable iff $\mathscr{D}$ has a solution with cost at most $2\alpha n$.*

Proof. Assume that F is satisfiable and let $A : \{x_1, x_2, \ldots, x_n\} \to \{true, false\}$ be a satisfying assignment. For the i-th variable gadget, take the solution R^0_i if $A(x_i) = true$ and R^1_i if $A(x_i) = false$. We choose a total of $2\alpha n$ rectangles and these rectangles dominates all the variable and clause rectangles.

On the other hand, suppose that there is a solution to $\mathscr{D}$ with cost at most $2\alpha n$. To dominate all the rectangles in a variable gadget requires at least 2α rectangles (see Claim 1). Note that all the variable gadgets are disjoint. Therefore, from each variable gadget we must choose exactly 2α rectangles (either set R^0_i or set R^1_i). We now show that $\mathscr{D}$ contains no rectangle r^t corresponding to the clause C_t. Assume that $r^t \in \mathscr{D}$. Let C_t contains the variable x_i. From the construction described above we say that r^t dominates a single vertex from the variable gadget of x_i and to dominate the remaining vertices from this gadget at least 2α rectangles are required. We now set the variable x_i to *true* if R^0_i is chosen from its variable gadget, otherwise set it to *false*. Since the solution dominates all the clause rectangles, hence by the construction we say that each clause is satisfied by this assignment. Therefore, the above assignment is a satisfying assignment. □

Clearly, DS-Rec-IL problem is in NP. Further, from Lemma 1, we conclude the following theorem.

Theorem 2 DS-Rec-IL *problem is* NP*-complete.*

Remark 1 *We prove that* DS-Rec-IL *problem is* NP*-complete even when each of the rectangles touches the straight line L at a single point from both sides of L.*

3 Intersecting Unit Squares

In this section, we show that DS-Sq-IL problem can be solved in polynomial time using dynamic programming. Let $S = \{s_1, s_2, \cdots, s_n\}$ be a set of n axis-parallel unit squares on the plane. The squares are intersecting a straight line L. We first rotate the given input configuration to make the straight line L parallel to the x-axis. Consider a horizontal strip T of height $\sqrt{2}$ such that the line L horizontally divides T into two equal parts above and below the line L. Since the squares are intersecting the line L, the center of all the squares in S are inside the strip T. The strip T is further partitioned into ***rectangles*** of width $\sqrt{2}$ and height $\sqrt{2}$. We remove all the rectangles that do not have any intersection with the given input squares. Clearly, there are at most $2n$ such rectangles that remain after the removal, since each square can intersect at most 2 rectangles.

Let $T_1, T_2, \ldots, T_k$ be these rectangles which are ordered from left to right. Add two additional rectangles T_0 and T_{k+1} such that, (i) T_0 is to the left of T_1, (ii) T_{k+1} is to the right of T_k, and (iii) no square in S intersects either T_0 or T_{k+1}.

Let $S_i \subseteq S$ be the set of squares which intersect the rectangle T_i. Further, let $S_i^c \subseteq S_i$ be the set of squares whose centers are inside T_i and $S_i^{nc} \subseteq S_i$ be the set of squares whose centers are outside T_i (see Figure 6). Clearly, $S_i^{nc} = S_i \setminus S_i^c$. More precisely, $S_i^{nc} \subseteq S_{i-1}^c \cup S_{i+1}^c$. We now prove the following result.

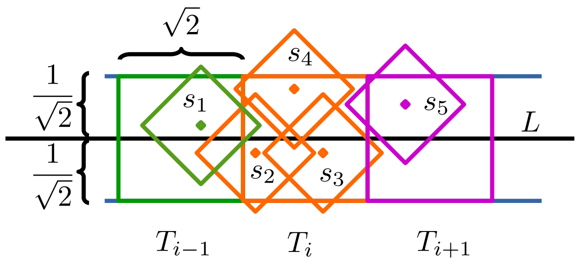

Figure 7: $S_i^c = \{s_2, s_3, s_4\}$ and $S_i^{nc} = \{s_1, s_5\}$.

Lemma 3 *The size of the optimal dominating set of squares for S_i is at most 12.*

Proof. We first prove that, at most 4 unit squares are sufficient to dominate all the squares in S_i^c. Observe that, both the width and height of the rectangle T_i are $\sqrt{2}$. Take 4 congruent squares $T_i^1, T_i^2, T_i^3, T_i^4$ such that each T_i^j, for $1 \leq j \leq 4$, is of length $\frac{1}{\sqrt{2}}$. If we arrange these 4 squares such that exactly 2 squares are in a column and exactly 2 squares are in a row, then their union fully cover the rectangle T_i. The center c_i^j of T_i^j is at most $\frac{1}{2}$ unit far from any other point inside T_i^j and hence at most one square with center inside T_i^j will dominate all the squares whose centers are inside T_i^j. Thus, any dominating set for the squares in S_i^c has size at most 4.

Observe that, the centers of the squares in S_i whose centers are outside T_i must belongs to T_{i-1} and T_{i+1}. This implies that, $S_i^{nc} \subseteq S_{i-1}^c \cup S_{i+1}^c$. Therefore, by the above argument we say that, the squares in S_i can be dominated by at most 12 squares, 4 squares each from rectangles T_{i-1}, T_i, and T_{i+1}. Since the squares whose center are inside T_i can only dominate a subset of squares in S_i, if an optimal solution OPT contains more than 12 square whose center are inside the rectangle T_i, we can replace them by 12 squares whose centers are in rectangles T_{i-1}, T_i, T_{i+1} without leaving any square to be dominated. This contradicts the assumption that OPT was an optimal dominating set. $\square$

For $0 \leq i \leq k+1$, let $D(S_i', S_{i-1}')$ where $S_i' \subseteq S_i$ and $S_{i-1}' \subseteq S_{i-1}$ denote the size of an optimal dominating set δ for the squares which lie completely inside $\cup_{j=0}^{i} T_j$ such that $\delta \cap S_i = S_i'$ and $\delta \cap S_{i-1} = S_{i-1}'$. Note that by Lemma 3, we can assume that both S_i' and S_{i-1}' have at most 24 squares. $D(S_i', S_{i-1}')$ satisfies the following recurrence:

- If $S_i' \cup S_{i-1}'$ does not dominate all squares which lie completely inside $T_i \cup T_{i-1}$, then $D(S_i', S_{i-1}') = \infty$.
- Otherwise,

$$D(S_i', S_{i-1}') = \min_{\substack{S_{i-2}' \subseteq S_{i-2}, \\ |S_{i-2}'| \leq 12}} D(S_{i-1}', S_{i-2}') + |S_i'|$$

We calculate the minimum dominating set by evaluating the function $D(S_{k+1}', S_k')$.

Running Time: We now calculate the time required to compute the optimal dominating set. There are at most $O(n^{24})$ subproblems and each subproblem depends on $O(n^{12})$ smaller subproblems. Hence, the total time required is $n^{O(1)}$.

Therefore, we have the following theorem.

Theorem 4 DS-SQ-IL *Problem can be solved in polynomial time.*

4 Touching Unit Squares

In this section, we prove that DS-SQ-TL problem can be solved in $O(n \log n)$ time. We reduce this problem to the minimum dominating set problem with uniform intervals (all intervals have same length) on real line. Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of axis-parallel unit squares. The squares touches a straight line L from above (see Figure 1(b)). Observe that, all the centers of the squares are on a straight line parallel to the line L. We move the line L to a position L' in the orthogonal direction of L until it passes through all the centers of all the squares in S (see Figure 8(a)).

We create an instance I of the minimum dominating set problem with uniform intervals on real line from an instance of DS-SQ-TL problem as follows. Let $s \in S$ be a square touching the line L from above. We take an interval $i_s \in I$ as the intersection of the square s and the line L' (see Figure 8(b)). It is easy to observe that, two square s_1 and s_2 intersect if and only if the corresponding two intervals i_{s_1} and i_{s_2} of s_1 and s_2 respectively intersect.

We now solve the minimum dominating set problem on I. Let $\{i_{s_1}, i_{s_2}, \ldots, i_{s_k}\}$ be the set of intervals returned by the algorithm. We return the squares $\{s_1, s_2, \ldots, s_k\}$

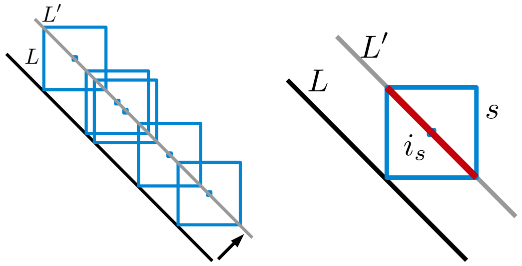

Figure 8: (a) Moving straight line L to L'. (b) A square s and its corresponding interval i_s.

as a solution of the DS-SQ-TL problem. The time required to solve the minimum dominating set problem is $O(n \log n)$ (greedy algorithm is enough, however one can look at [18]). Hence, we have the following theorem.

Theorem 5 *The* DS-SQ-TL *problem can be solved in* $O(n \log n)$ *time.*

References

[1] V. Chepoi and S. Felsner. Approximating hitting sets of axis-parallel rectangles intersecting a monotone curve. *Computational Geometry*, 46(9):1036 – 1041, 2013.

[2] M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation*, 206(11):1264 – 1275, 2008.

[3] J. Correa, L. Feuilloley, P. Pérez-Lantero, and J. A. Soto. Independent and hitting sets of rectangles intersecting a diagonal line: Algorithms and complexity. *Discrete & Computational Geometry*, 53(2):344–365, 2015.

[4] T. Erlebach and E. J. Van Leeuwen. Domination in geometric intersection graphs. In *LATIN*, pages 747–758, 2008.

[5] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[6] M. R. Garey and D. S. Johnson. The rectilinear steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[8] M. Gibson and I. A. Pirwani. Algorithms for dominating set in disk graphs: Breaking the $\log n$ barrier - (extended abstract). In *ESA*, pages 243–254, 2010.

[9] S. Har-Peled and K. Quanrud. Approximation algorithms for polynomial-expansion and low-density graphs. In *ESA*, pages 717–728, 2015.

[10] H. B. Hunt, M. V. Marathe, V. Radhakrishnan, S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *Journal of Algorithms*, 26(2):238 – 274, 1998.

[11] V. Kann. *On the approximability of NP-complete optimization problems*. PhD thesis, 1992.

[12] D. E. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427, 1992.

[13] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

[14] D. Marx. Parameterized complexity of independence and domination on geometric graphs. In *IWPEC*, pages 154–165, 2006.

[15] D. Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008.

[16] A. Mudgal and S. Pandit. Covering, hitting, piercing and packing rectangles intersecting an inclined line. In *COCOA*, pages 126–137, 2015.

[17] T. Nieberg, J. Hurink, and W. Kern. Approximation schemes for wireless networks. *ACM Trans. Algorithms*, 4(4):49:1–49:17, 2008.

[18] G. Ramalingam and C. Rangan. A unified approach to domination problems on interval graphs. *Information Processing Letters*, 27(5):271 – 274, 1988.

[19] E. J. van Leeuwen. *Optimization and Approximation on Systems of Geometric Objects*. PhD thesis, 2009.

[20] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., 2001.