ANALYSIS OF GREEDY ALGORITHM FOR DOMINATING SET PROBLEM ON ANCHORED RECTANGLES

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE

by

Souborno Choudhury
[Roll No. CS1728]

Under the Guidance of

Dr. Sasanka Roy
Associate Professor
Advance Computing and Microelectronics Unit

INDIAN STATISTICAL INSTITUTE
KOLKATA - 700108, INDIA
July 2019

Dedicated to my parents

Declaration

I hereby declare that the project report entitled **"Analysis of Greedy Algorithm for Dominating Set Problem on Anchored Rectangles"** submitted to Indian Statistical Institute, Kolkata, is a bonafide record of work carried out in partial fulfillment for the award of the degree of **Master of Technology in Computer Science**. The work has been carried out under the guidance of **Dr. Sasanka Roy**, Associate Professor, Advance Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata.

I further declare that this work is original, composed by myself. The work contained herein is my own except where stated otherwise by reference or acknowledgement, and that this work has not been submitted to any other institution for award of any other degree or professional qualification.

Place: Kolkata
Date: July 20, 2019

Souborno Choudhury
Roll No: CS1728

Indian Statistical Institute
Kolkata - 700108, India

CERTIFICATE

This is to certify that the work contained in this project report entitled **"Analysis of Greedy Algorithm for Dominating Set Problem on Anchored Rectangles"** submitted by **Souborno Choudury** of **Roll No:CS1728** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The project report has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Dr. Sasanka Roy
Associate Professor
Advance Computing and Microelectronics Unit
Indian Statistical Institute
KOLKATA - 700108, INDIA

Acknowledgement

I would like to show my highest gratitude to my supervisor, *Dr. Sasanka Roy*, Associate Professor, Advance Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. He has taught me how to do good research and motivated me with great insights and innovative ideas.
I would like to thank *Prof. Subhas Chandra Nandy*, for his valuable suggestions and discussions. I would also like to thank *Binayak Dutta* and *Satyabrata Jana*, for scrutinizing the thesis report.
My deepest thanks to all the teachers of Indian Statistical Institute, Kolkata, for their valuable suggestions which added an important dimension to my research work.
Finally, I am very thankful to my father and *Surosree Mukhopadhyay* for their everlasting support and motivation.
Last but not least, I would like to thank all of my friends for their help. I would also like to thank anyone, that I may have unintentionally missed out above.

Abstract

We are given a set $\mathcal{R}$ of n Axis-Parallel Rectangles in the plane. We study the Dominating set problem on $\mathcal{R}$. The bottom left vertex of each rectangle in set $\mathcal{R}$ is constrained to touch a straight diagonal line of $135°$. We study the performance of greedy algorithm for *Minimum Dominating set (MDS)* problem on the *Intersection Graph* of $\mathcal{R}$. We give a construction, on $\mathcal{R}$, where Greedy technique yields $\Theta(\log n)$-*factor* approximation. This proves that the approximation ratio for Greedy algorithm for MDS problem is $\Theta(\log n)$ even for this constrained version of MDS problem. We also do an experimental study of Greedy algorithm of MDS problem for randomly generated arbitrary rectangles. We compare the performance of greedy algorithm with optimal result obtained by solving *Integer Linear programming (ILP)* formulation of MDS problem.

Keywords: Axis Parallel, Rectangle, Diagonal, Minimum Dominating Set, Intersecting Graph.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

To find the Dominating Set of a given set of geometrical objects is a problem which has been researched intensively in the field of Computer Science. We are given a set $\mathcal{R}$ of n Axis-Parallel Rectangles, in 2-Dimension on which we study the Dominating set problem. The bottom left vertex of each rectangle in set $\mathcal{R}$ is constrained to touch a straight diagonal line of 135°, as depicted in figure 1.1.

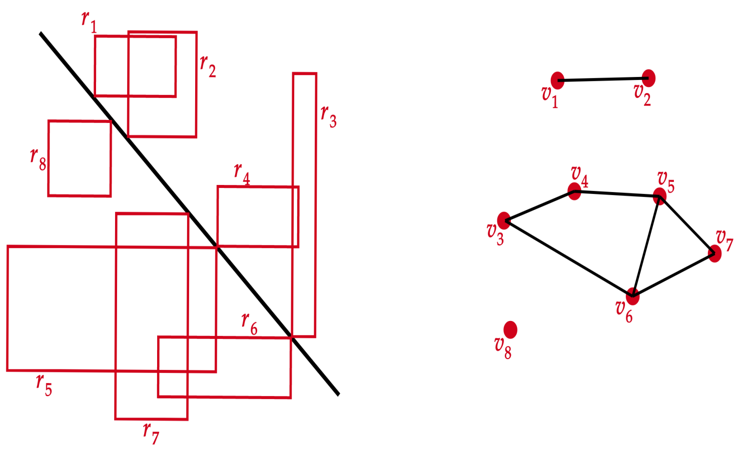

Figure 1.1: (Left) Rectangles Anchored on Straight Diagonal line of slope 135° and (Right) its corresponding Intersection Graph G_I

Definition 1.0.1. *Intersection Graph* $G_I = (V, E)$ of a set of rectangle $\mathcal{R}$ is defined as, each vertex $v_i \in V$ represents a rectangle $r_i \in \mathcal{R}$ and has a one-to-one correspondence with r_i. There is an edge $e(v_i, v_j) \in E$ if and only if, corresponding rectangles r_i and r_j have non-empty intersection, that is $r_i \cap r_j \neq \emptyset$.

It has been proved that finding the minimum dominating set on *intersection graph* G_I for any set of n rectangles $\mathcal{R}$,defined above is a NP-Hard problem in 2017 [4]. Till date no constant factor approximation algorithm has been provided. For the first time it was shown that for an $\epsilon > 0$, there exists an $(2+\epsilon)-$approximation algorithm in 2018 [1]. In this thesis report we consider this problem.

• **MDS-R-LB-DL:** Finding Minimum Dominating Set on set of n rectangles, when bottom left vertex of each rectangle touches a diagonal straight line of slope 135°, as depicted in figure 1.2 where $n = 7$.

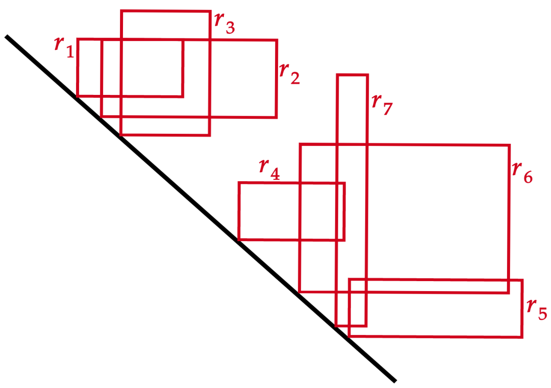

Figure 1.2: Rectangles Anchored only one one side of Straight Diagonal line of slope 135°

We generate set $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$ of n axis parallel L-frames anchored on diagonal line of slope 135°, as depicted in figure 3.5. These L-frames will represent the left and bottom edge of set $\mathcal{R}$ of n rectangles. Since, any two L-frame l_i and l_j will only intersects if and only if there corresponding rectangles r_i and r_j intersects, respectively[1].

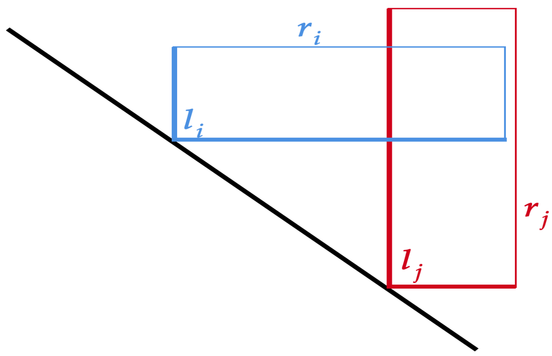

Figure 1.3: One-To-One correspondence of L-frames with Rectangles

Thus, *Intersection Graph* G_I of corresponding set $\mathcal{L}$ of n L-frames and set $\mathcal{R}$ of n rectangles will be equal.

Chapter 2

Preliminaries

2.1 Set Cover Problem

We are given a *set system*, $\Pi = (\mathcal{U}, \mathcal{S})$, where $\mathcal{U}$ is a finite set of elements called *universe* and $\mathcal{S} = \{s_1, s_2, s_3, \ldots, s_n\}$ is a collection of n subset of $\mathcal{U}$, such that $\bigcup_{i=1}^{n} s_i = \mathcal{U}$.
Now, we have to compute a *set cover*, $\mathcal{D} \subset \{1, 2, 3, \ldots, n\}$ of minimum cardinality such that,

$$\bigcup_{j \in \mathcal{D}} s_j = \mathcal{U}, \quad \forall s_j \in \mathcal{S} \tag{2.1}$$

This is an optimization problem which belongs to the class of NP-Hard Problems[3]. The corresponding decision problem, that is, whether there exits a *set cover* $\mathcal{D}$ of size k, is NP-Complete Problem [3].

2.1.1 Greedy Approximation for finding a Set Cover

We give a brief summary of greedy approach that can be used to get a *set cover*.
Start with a empty *set cover*, G, and mark all elements of *universe* as uncovered.
Choose the $s_x \in \mathcal{S}$, such that s_x contains the most number of elements of *universe* which are uncovered, and put s_x in G.

Mark all elements of *universe* as covered which also belongs to s_x.
Keep repeating till all elements of the *universe* is covered.
The set G will be our *set cover* of the *universe*.

2.1.2 Assumption

In the input *set system*, $\Pi = (\mathcal{U}, \mathcal{S})$ provided for our Algorithm 1, the condition $\bigcup_i s_i = \mathcal{U}$, $\forall s_i \in \mathcal{S}$ holds.

2.1.3 Algorithm

Algorithm 1 Find Smallest Set Cover using Greedy

```
procedure SetCover(𝒰,𝒮)                        ▷ Find set cover of universal set
    uncovered ← 𝒰
    𝒞 ← ∅                                       ▷ Set Cover
    while uncovered ≠ ∅ do                      ▷ Elements of universe uncovered
        A(x) = {x ∈ 𝒮 | cardinality of (x ∩ uncovered) is largest}
        𝒞 ← 𝒞 ∪ {A(x)}
        uncovered ← uncovered \ A(x)
    return 𝒞                                    ▷ 𝒞 is the required Set Cover
```

2.1.4 Correctness

Theorem 2.1.1. *The set cover* $\mathcal{C}$ *returned by Algorithm* 1*, is indeed the set cover the universe* $\mathcal{U}$*.*

Proof. The while loop stops only when there is no more elements of the *universe* $\mathcal{U}$ is left uncovered. The *universe* $\mathcal{U}$ can be covered which follows from the assumption considered above. □

2.1.5 Complexity

Theorem 2.1.2. *The Algorithm* 1 *runs in polynomial time.*

Proof. Suppose we consider m is the size of the *universe*, each elements in the family of subset can be at most of size k, and $n = |\mathcal{S}|$, that is cardinality of set $\mathcal{S}$.
The algorithm runs for $\mathcal{O}(mkn)$ steps. Since in each iteration of the while loop, at least one element of the universe must be covered. So the while loop runs for at most m times in worst case if exactly one element of *universe* is covered in each iteration. Finding $A(x)$ can take at most kn steps to check all n elements of $\mathcal{S}$, which are at most of size k.
Further since k is bounded by m, as elements of $\mathcal{S}$ are subsets of *universe* $\mathcal{U}$, thus we can say it as $\mathcal{O}(m^2 n)$. □

2.1.6 Upper Bound of Worst Case Approximation

We know in worst case greedy technique yields $O(\log n)$-*factor* approximate of the optimal result.
Let $\mathcal{P}$ be the optimal cover, $\mathcal{G}$ be the greedy set cover, and n is the size of the $\mathcal{S}$, then $|\mathcal{G}| = |\mathcal{P}| (\log n)$.

Theorem 2.1.3. $|\mathcal{G}| \leqslant |\mathcal{P}| (\log n)$, *for any set system* $\Pi = (\mathcal{U}, \mathcal{S})$. *Where* $n = |\mathcal{U}|$, $\mathcal{G}$ *is the set cover returned by Greedy algorithm and* $\mathcal{P}$ *is the minimum set cover* [5].

2.2 Minimum Dominating Set Problem(MDS)

We are given a graph $G = (V, E)$, where V is the set of *vertices* and $E \subset V \times V$ is the set of *edges*.

Definition 2.2.1. Dominating Set: $\mathcal{D} \subset V$ is a *dominating set* if and only if $\forall v \in V$ either $v \in \mathcal{D}$ or $\exists e(v, u) \in E$ and $u \in \mathcal{D}$.

Now, we have to compute a minimum dominating set $\mathcal{M}$, such that $\forall D$, D is a *dominating set* of G, $|M| \leqslant |D|$.

For a general graph this is an optimization problem which belongs to the class of NP-Complete Problems[2].

2.2.1 Integer Linear Programming(ILP) formulation of MDS Problem

Definition 2.2.2. Adjoined of a vertex v in a graph $G = (V, E)$ is,
$Adj(v) = \{u \mid \forall u \in V, \exists e(u, v) \in E\}$.

Definition 2.2.3. Neighbour of vertex v is, $N(v) = Adj(v) \cup \{v\}$.

Given a graph $G = (V, E)$, $\mid V \mid= n$:

$$\begin{array}{ll} \min & \sum_{v \in V} x_v \\ \forall v \in V & \sum_{u \in N(v)} x_u \geq 1 \\ \forall v \in V & x_v \in \{0, 1\} \end{array} \tag{2.2}$$

2.2.2 Reduction of MDS to Set Covering Problem

Given a graph $G = (V, E)$, where V is the set of *vertices* and $E \subset V \times V$ is the set of *edges*. We have to reduce it to a *set system*, $\Pi = (\mathcal{U}, \mathcal{S})$, and then solve the *Set Cover Problem* on Π

$\mathcal{U}$*(universe)* $= \{v_1, v_2, v_3, \ldots, v_n\}$, $\forall v_i \in V$.
$\mathcal{S} = \{s_1, s_2, s_3, \ldots, s_n\}$, $s_i = N(v_i)$, $\forall i \in [1, 2, \ldots, n]$.

Now the procedure **SetCover**($\mathcal{U}$,$\mathcal{S}$) provides us with the required *minimal dominating set* of the provided graph $G = (V, E)$, which runs in $O(n^3)$, from Theorem 2.1.2.

Theorem 2.2.4. *For given a graph $G = (V, E)$, where V is the set of vertices and $E \subset V \times V$ is the set of edges, reduction* *of the graph G to a set system, $\Pi = (\mathcal{U}, \mathcal{S})$, can be done in polynomial time.*

Proof. Let $\mid V \mid= n$, then $\mathcal{U} = \{1, 2, \ldots, n\}$, the time complexity of creating $\mathcal{U}$ is $O(n)$.
Now suppose the graph $G = (V, E)$ is given as input in adjacency list format, for example *adjacency list* of *vertex* v_i is $Adj(v_i) = \{u_1, u_2, \ldots, u_k\}$ $\forall v_i \in V$, we create *neighbourhood* of *vertex* v_i as $N(v_i) = \{v_i, u_1, u_2, \ldots, u_k\}$ $\forall v_i \in V$.
That is, put v_i as the first element in $Adj(v_i)$ $\forall v_i \in V$, to get $N(v_i)$. Now $\mathcal{S} = \{N(v_i) : \forall v_i \in V\}$, thus the time complexity of creating $\mathcal{S}$ is also in $O(n)$.
Hence, for a given graph $G = (V, E)$, where V is the set of *vertices* and $E \subset V \times V$ is the set of *edges*, reducing it to a *set system*, $\Pi = (\mathcal{U}, \mathcal{S})$, can be done in polynomial time. □

Theorem 2.2.5. *The minimal dominating set can be produced from the minimal set cover returned by the procedure* **SetCover**($\mathcal{U}$,$\mathcal{S}$) *in polynomial time.*

Proof. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ be the minimal set cover returned by the procedure **SetCover**($\mathcal{U}$,$\mathcal{S}$). Now produce a set $\mathcal{M} = \{m_i : 1^{st} \textit{element of } \ c_i, \ \forall c_i \in \mathcal{C}\}$. Retracing back from the steps defined in Theorem 2.2.4. Now set $\mathcal{M}$ is our required minimal dominating set, thus the time complexity of creating $\mathcal{M}$ is $O(k)$ and, $\mid \mathcal{S} \mid= n$ and $k \leqslant n$. Hence, minimal dominating set can be produced from the minimal set cover returned by the procedure **SetCover**($\mathcal{U}$,$\mathcal{S}$) in polynomial time. □

Chapter 3

Our Contribution

We list our contributions as follows:
• We prove that, Greedy technique yields $\Theta(\log n)$-*factor* approximation for **MDS-R-LB-DL**.
• We give an experimental comparison of solution of **MDS-R-LB-DL** obtained by Greedy technique with optimal result on randomly generated arbitrary rectangle set $\mathcal{R}$, as the data set for **MDS-R-LB-DL**.

3.1 Worst Case of Greedy Heuristic

Theorem 3.1.1. *Greedy technique yields* $\Theta(\log n)$-*factor approximation for* ***MDS-R-LB-DL****.*

Proof. We will prove this in two parts p_1 and p_2.

p_1 = Greedy technique yields $O(\log n)$-*factor* approximation for **MDS-R-LB-DL**.
p_2 = Greedy technique yields $\Omega(\log n)$-*factor* approximation for **MDS-R-LB-DL**.

By proving p_1 and p_2, we can conclude that indeed, Greedy technique yields $\Theta(\log n)$-*factor* approximation for **MDS-R-**

LB-DL and hence proving Theorem 3.1.1.

p_1 can be concluded from Theorem 2.1.3, since we are reducing MDS to set cover problem, hence, greedy technique yields $O(\log n)$-*factor* approximation for **MDS-R-LB-DL**.

For the proof of p_2 we are going to construct a particular case, thus concluding indeed Greedy technique yields $\Omega(\log n)$-*factor* approximation for **MDS-R-LB-DL**.

Consider the figure 3.1 given below for $n = 3$, as an example case where there is at least one instance for which greedy solution is $\Omega(\log n)$-*factor* approximation. The corresponding intersection graph is given in figure 3.2, thus Greedy technique yields $\Omega(\log n)$-*factor* approximation for **MDS-R-LB-DL**.
Hence, Greedy technique yields $\Omega(\log n)$-*factor* approximation for **MDS-R-LB-DL**.

□

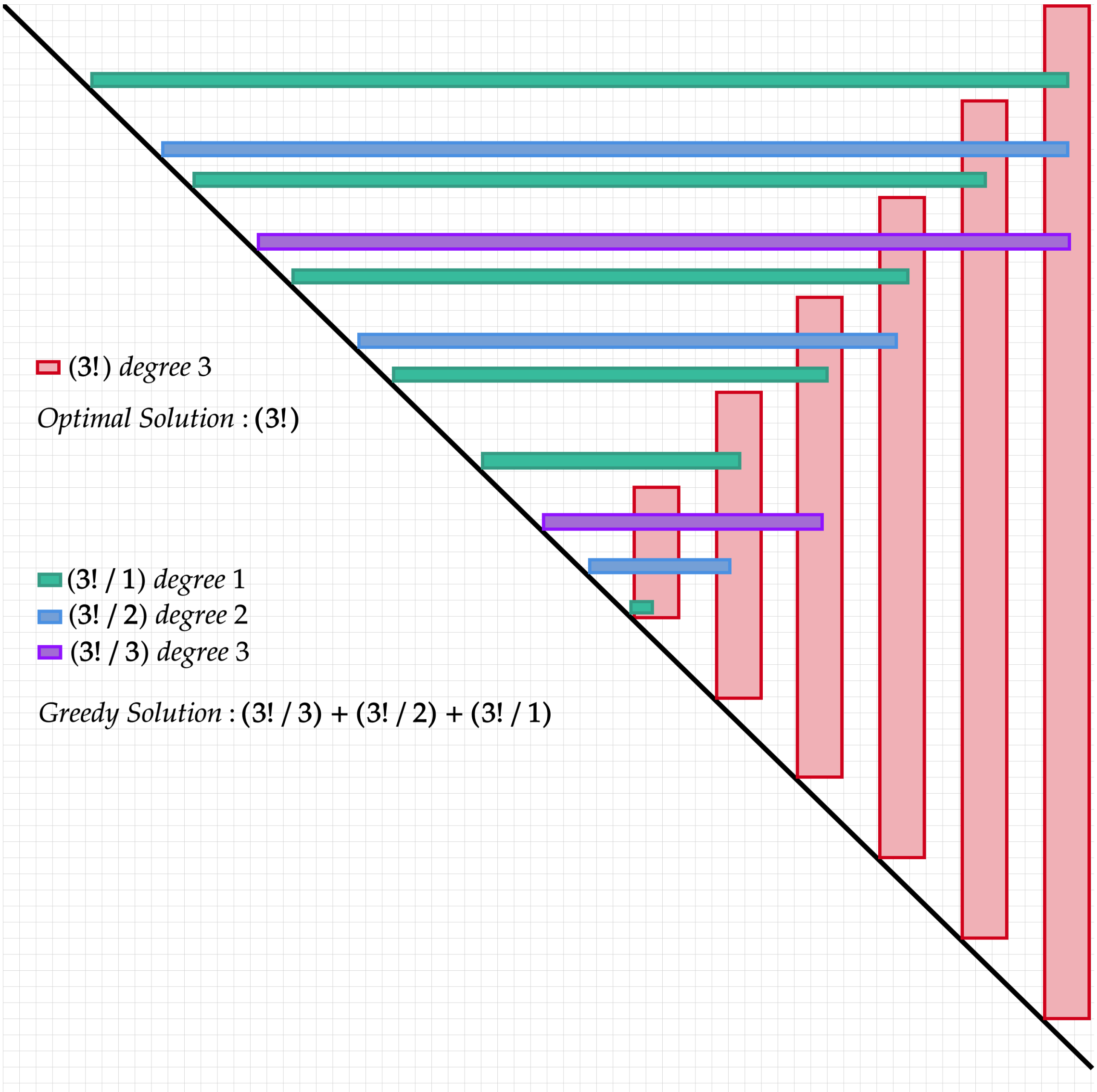

Figure 3.1: Greedy Worst-case 1-Sided Diagonal-Anchored Rectangle Instance

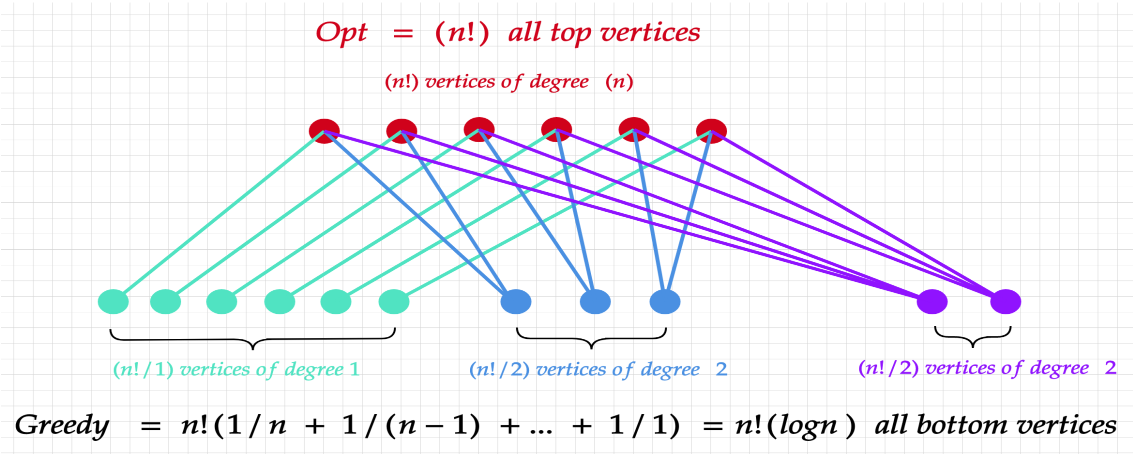

Figure 3.2: Corresponding Intersection Graph

3.1.1 Idea or Construction for any n for Figure 3.1

STEP 1: Draw $n!$ number of rectangles aligned on the diagonal as illustrated in figure 3.3, and lets name them $\mathcal{E} = \{e_1, e_2, \ldots, e_{n!}\}$.

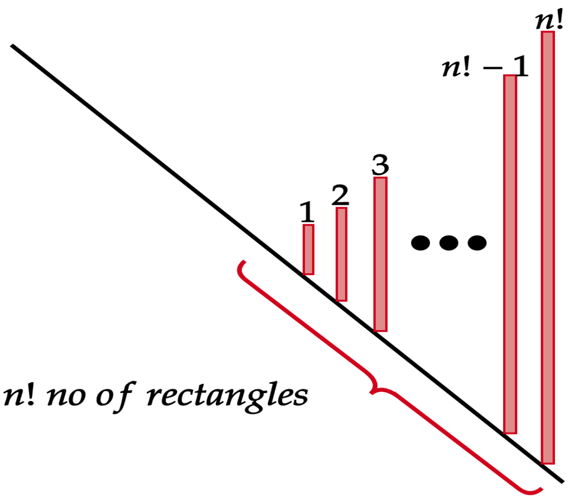

Figure 3.3: $n!$ Vertical Rectangles

STEP 2: Now draw $n!/1$ number of rectangles intersecting exactly 1 rectangle of $\mathcal{E}$, $n!/2$ number of rectangles intersecting exactly 2 rectangle of $\mathcal{E}$, keep doing it till, $n!/n$ number of rectangles intersecting exactly n rectangle of $\mathcal{E}$, aligned on the diagonal as illustrated in figure 3.4.

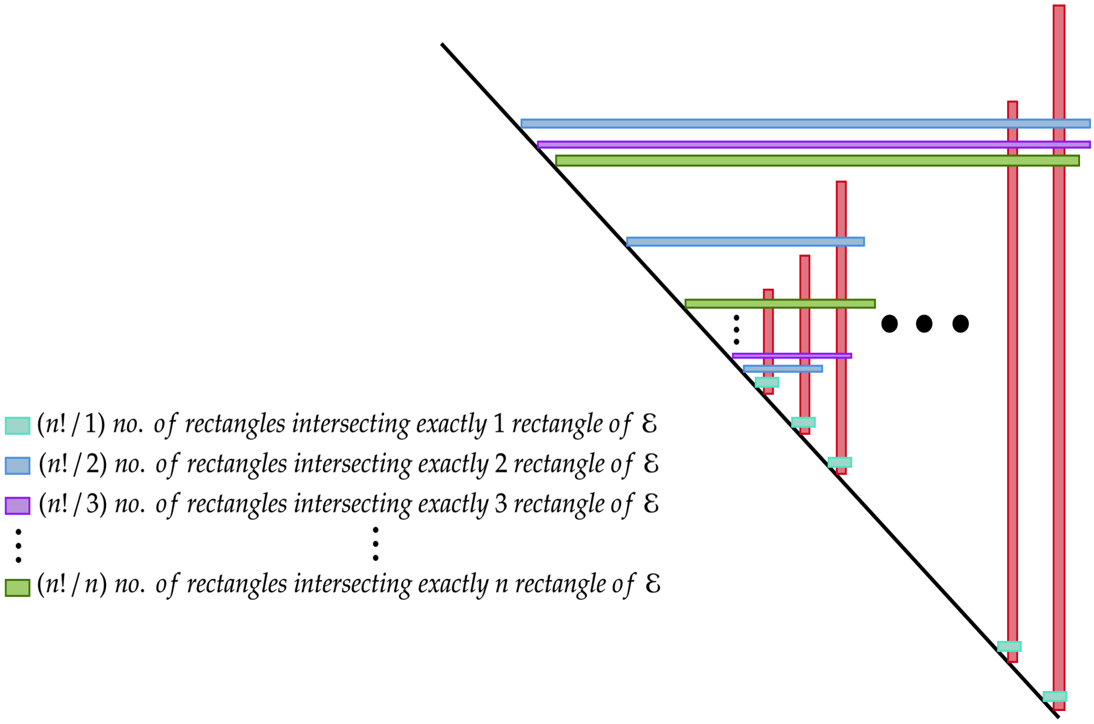

Figure 3.4: $n!(1/1 + 1/2 + \ldots + 1/n)$ Horizontal Rectangles

3.2 Comparison of Greedy Solution vs Optimal Solution for MDS-R-LB-DL

We have given a detailed empirical study which shows that our greedy algorithm produces a very close result to optimal result for randomly generated arbitrary graph. In fact we can clearly see from our experimental studies that greedy algorithm produces at most 1.20 times worse than the optimal result from table 4.1.

We generate set $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$ of n axis parallel L-frames anchored on diagonal line of slope 135°, as depicted in figure 3.5 for $n = 6$. These L-frames will represent the left and bottom edge of set $\mathcal{R}$ of n rectangles.

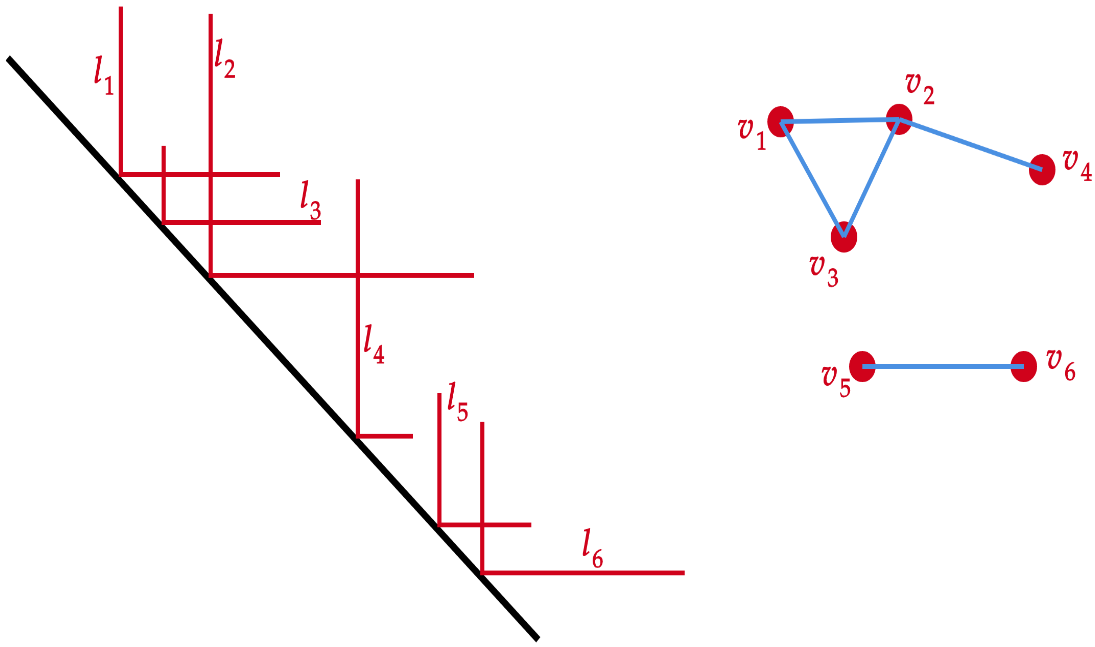

Figure 3.5: (Left) L-frames Anchored on Straight Diagonal line of slope 135° and (Right) its corresponding Intersection Graph G_I.

3.2.1 Detailed Explanation of Data Set Generation

We set line $x+y = 100$ for $n \in \{10, 50, 100, 500, 1000, 5000, 10000\}$, as our diagonal straight line on which the set $\mathcal{L}$ of n L-frames will be anchored as in figure 3.5.

We have used pseudo-random generating function to generate the *anchoring vertex*(x_v, y_v), such that $x_v + y_v = 100$ for $n \in \{10, 50, 100, 500, 1000, 5000, 10000\}$, the length of bottom edge b and the length of left edge h of each L-frame, as depicted in figure 3.6.

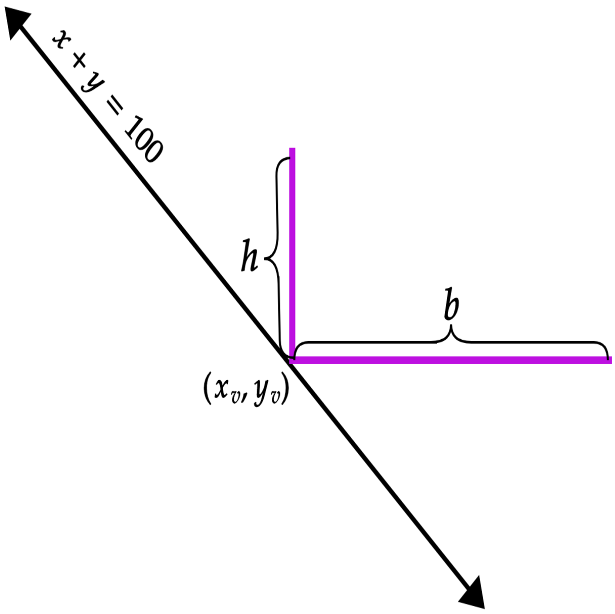

Figure 3.6: Single Randomly Generated L-frame

We have created *Intersection Graph*, G_I of corresponding set $\mathcal{L}$ of n L-frames, as depicted in figure 3.5(Right). On the *Intersection Graph*, G_I we have done our experiment, and accounted result of comparison between greedy solution vs optimal solution for **MDS-R-LB-DL**.

Chapter 4

Experimental Studies

We have performed a detailed experiment, and done a comparison on the size of minimal dominating set produced by our greedy algorithm 1 and optimal result on **MDS-R-LB-DL** problem. We created *Intersection Graph* G_I on randomly generated set of n axis parallel L-frames $\mathcal{L} = \{l_1, l_2, \ldots, l_n\}$ anchored on diagonal line of slope 135° away from origin. Values of $n \in \{10, 50, 100, 500, 1000, 5000, 10000\}$. We created 5 instances for each value of n and accounted the ratio of size of dominating set returned by greedy approach to the size of optimal dominating set *(which, in this case is minimum)*, presented in table 4.1.

$f(n, m) = \{\mid G \mid / \mid P \mid\}$, for n number of L-Frames for m^{th} instance, where G is minimal dominating set returned by greedy approach and P is optimal dominating set *(which, in this case is minimum).*
We have obtained $\mid P \mid$, by running ILP on MDS.
$\text{A}(n) = (f(n,1)+f(n,2)+f(n,3)+f(n,4)+f(n,5))/5 \ \forall n$, that is the *average* ratio of all 5 instances.

No. of L-frames	$f(n,1)$	$f(n,2)$	$f(n,3)$	$f(n,4)$	$f(n,5)$	A(n)
10	3/2=1.50	2/2=1.00	2/2=1.00	2/2=1.00	3/2=1.50	1.20
50	5/4=1.25	6/5=1.20	4/4=1.00	4/4=1.00	4/4=1.00	1.09
100	18/16=1.13	12/11=1.09	11/11=1.00	17/15=1.13	15/13=1.15	1.10
500	27/24=1.13	30/25=1.20	27/23=1.17	30/24=1.25	28/24=1.17	1.18
1000	30/26=1.15	30/26=1.15	33/27=1.22	32/27=1.19	34/28=1.21	1.18
5000	43/34=1.26	39/34=1.15	40/34=1.18	41/34=1.21	40/34=1.18	1.20
10000	27/24=1.13	30/25=1.20	27/23=1.17	30/24=1.25	28/24=1.17	1.08

Table 4.1: Observation Data of Ratio of 5 different instances for Greedy vs Optimal result for n number of our Experiment.

We also plot a graph for A(n) versus n for all values of n, for $n \in \{10, 50, 100, 500, 1000, 5000, 10000\}$, in figure 4.1. Values of A(n) is taken form the table 4.1.

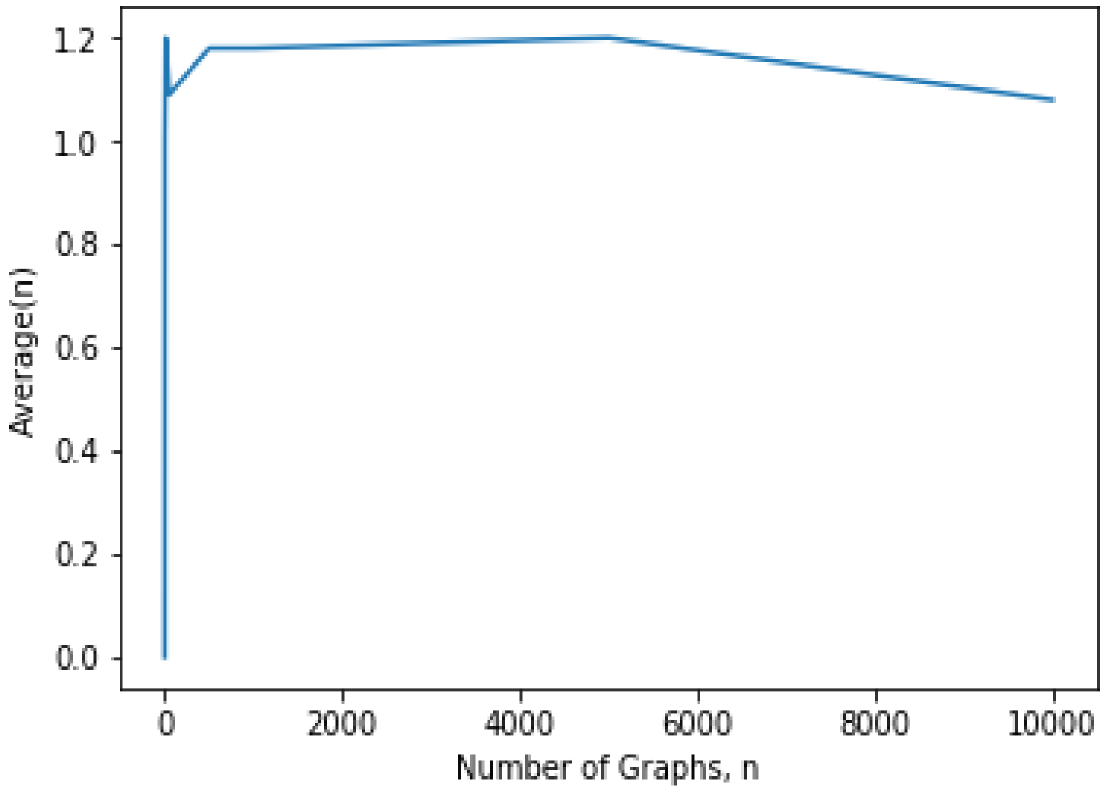

Figure 4.1: Plot of Average Ratio of 5 different instances for Greedy vs Optimal result for n number of our Experiment.

Chapter 5

Conclusion and Future Work

Thus, we draw a conclusion that, finding Minimum Dominating Set on set $\mathcal{R}$ of n Axis-Parallel Rectangles in the plane, when the bottom left vertex of each rectangle in set $\mathcal{R}$ is constrained to touch a straight diagonal line of $135°$, as depicted in figure 1.2, using greedy technique yields $\Theta(\log n)$ factor approximation. We also experimentally analyzed the performance of Greedy algorithm on randomly generated test samples.

Till date it is not proved or disproved if, finding Minimum Dominating Set on set $\mathcal{R}$ of n Axis-Parallel Rectangles in the plane, when the bottom left vertex of each rectangle in set $\mathcal{R}$ is constrained to touch a straight diagonal line of $135°$, as depicted in figure 1.2 is actually a NP-Hard problem. Hopefully this may be proved in near future, if so then it will be interesting to see if a *constant-factor* approximation is possible. And if disproved then a polynomial time algorithm to obtain the optimal solution may be designed.

Bibliography

[1] Sayan Bandyapadhyay, Anil Maheshwari, Saeed Mehrabi, and Subhash Suri. Approximating dominating set on intersection graphs of rectangles and l-frames. *Computational Geometry*, 82:32–44, 2019.

[2] Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[3] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[4] Supantha Pandit. Dominating set of rectangles intersecting a straight line. In *CCCG*, pages 144–149, 2017.

[5] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.