A Defect Tolerance Scheme for Nanotechnology Circuits

Under the guidance of :
Prof. Malay Kule

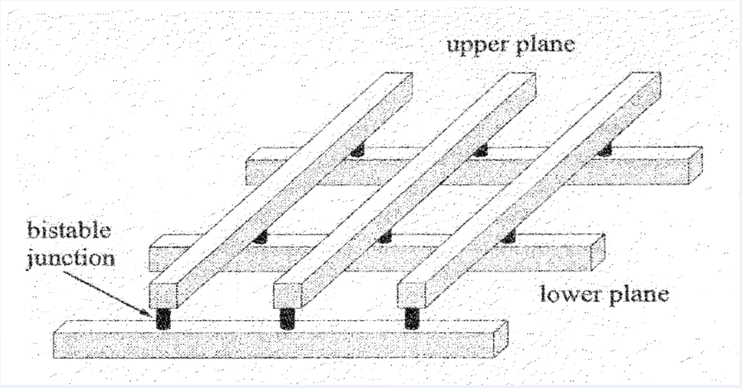
Prepared by:
Sayantan Pandit (Exam Roll - 111205022)
Prameya Mandal (Exam Roll - 111205023)
Saptarshi Nag (Exam Roll - 111205039)
Subha Banerjee (Exam Roll – 111105056)

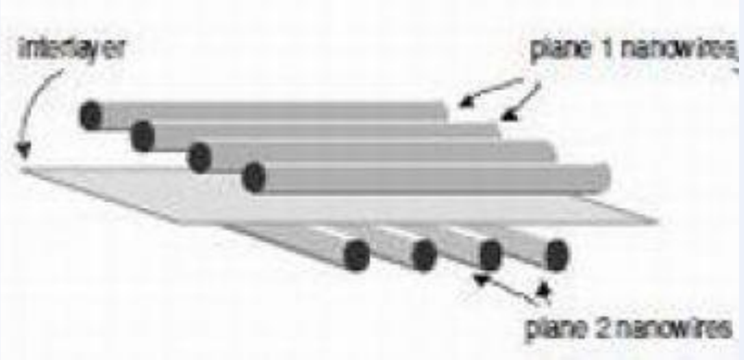
INTRODUCTION

1. Two parallel planes of nanoscale wire arrays

2. The region where two wires cross is called a junction.

3. Properties of the crossbar is determined by inter-connected junction





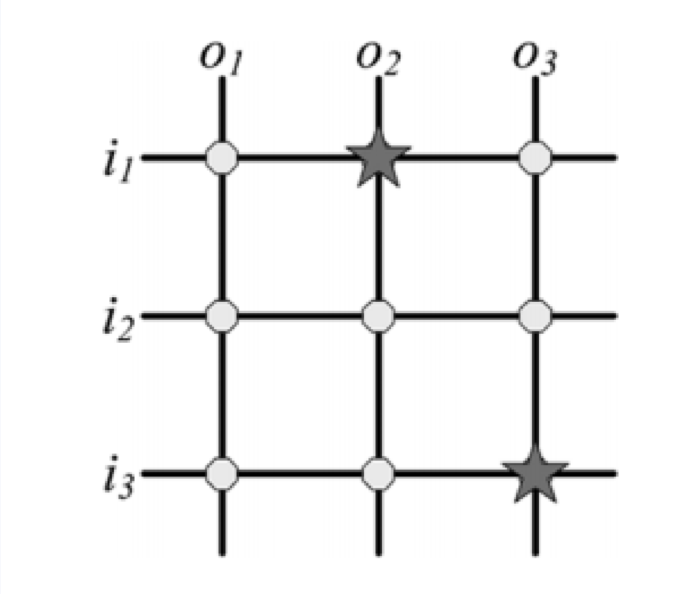
INTRODUCTION

1. LITHOGRAPHY-BASED silicon VLSI technology

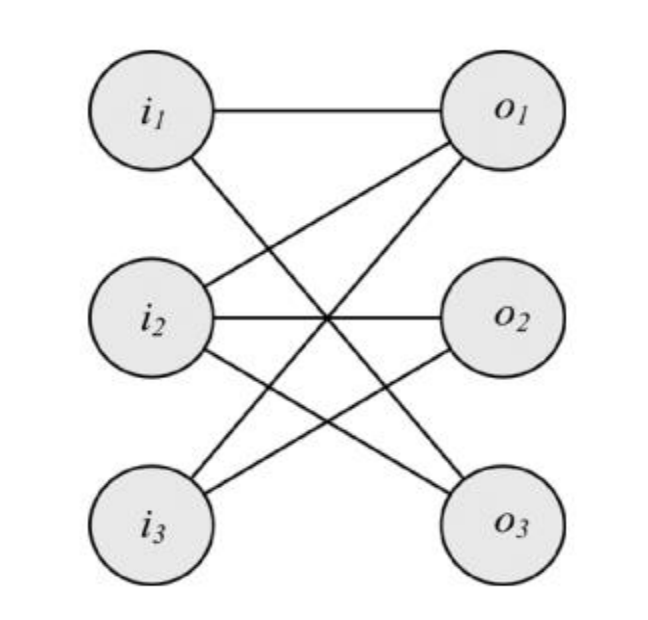
2. Very high density and high frequencies

3. Programmable logic (PLA) planes

CROSSBAR DEFECTS



Crossbar with two defects



Graph representation of the crossbar

LITERATURE SURVEY

1. Faults can be repaired

2. Polynomial time using Bipartite matching algorithm

3. HP observed that 85% are programmable.

PROBLEMS

1. Total number of switch is n x n with some k x k defects switch
2. Maximizing k for a defect switch leads to higher densities on the same defective circuit
3. But, the problem of finding the maximum k x k is NP Complete

Our Proposed Algorithm

Step 1: Choose an element of the ON-set

Step 2: Find "maximal" groupings of 1s and Xs adjacent to that element

- consider top/bottom row, left/right column, and corner adjacencies
- this forms prime implicants Repeat Steps 1 and 2 to find all prime implicants

Step 3: Revisit the 1s in the K-map

- if covered by single prime implicant, it is essential, and participates in final cover
- 1s covered by essential prime implicant do not need to be revisited

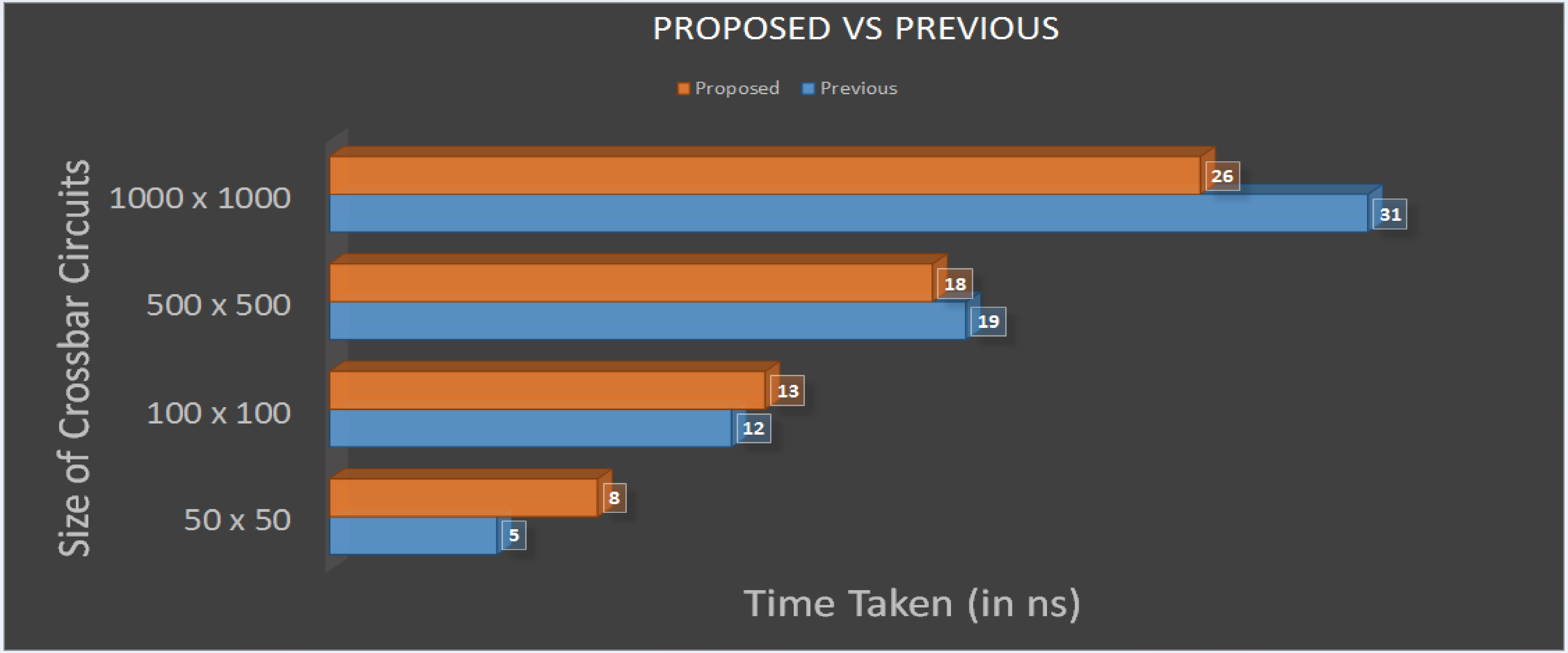
Step 4: if there remain 1s not covered by essential prime implicants

- select the smallest number of prime implicants that cover the remaining 1s

Our Proposed Algorithm

		1				
	1	1				
	1	X				

COMPARISON



PROPOSED VS PREVIOUS
Proposed
Previous
Size of Crossbar Circuits
1000 x 1000
500 x 500
100 x 100
50 x 50
26
31
18
19
13
12
8
5
Time Taken (in ns)

ADVANTAGES

1. Closed junction error

2. Time Complexity is better

DISADVANTAGE

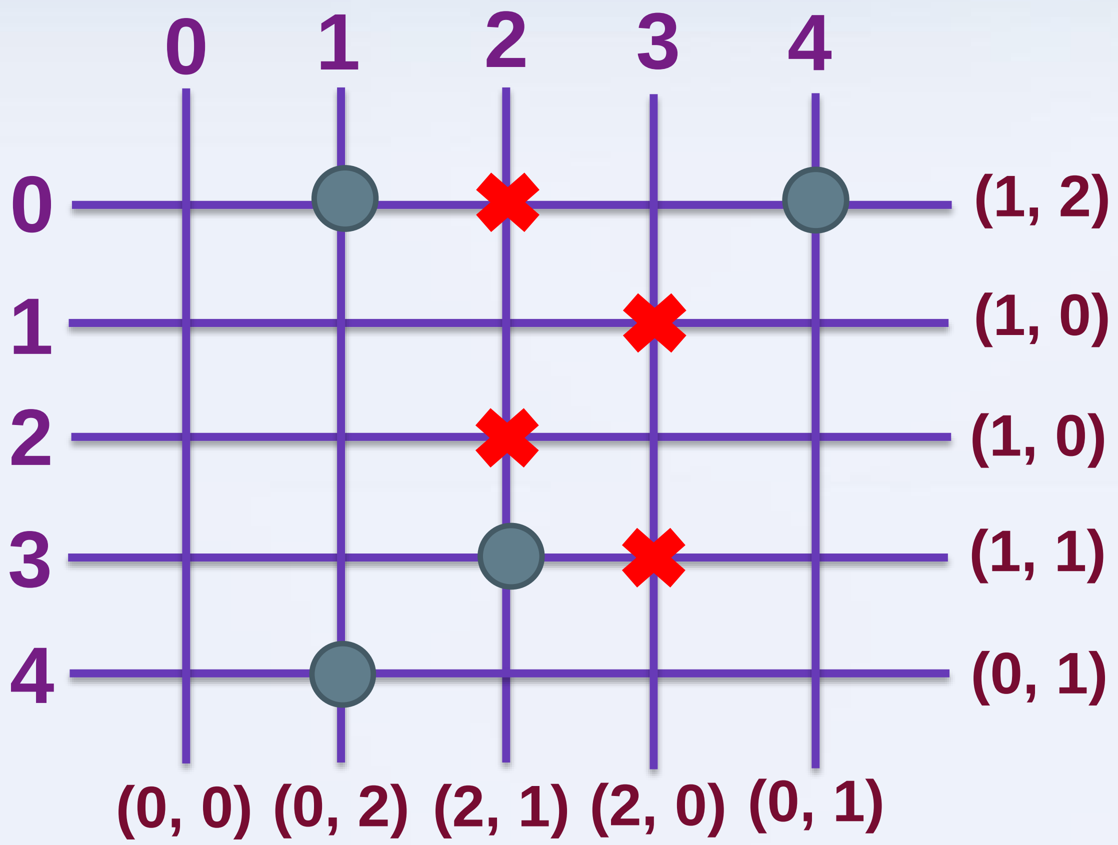
1. Time complexity increases if the junction error is not adjacent
2. Space Complexity remains same i.e still uses $O(n^2)$

Algorithm – K1 x K2

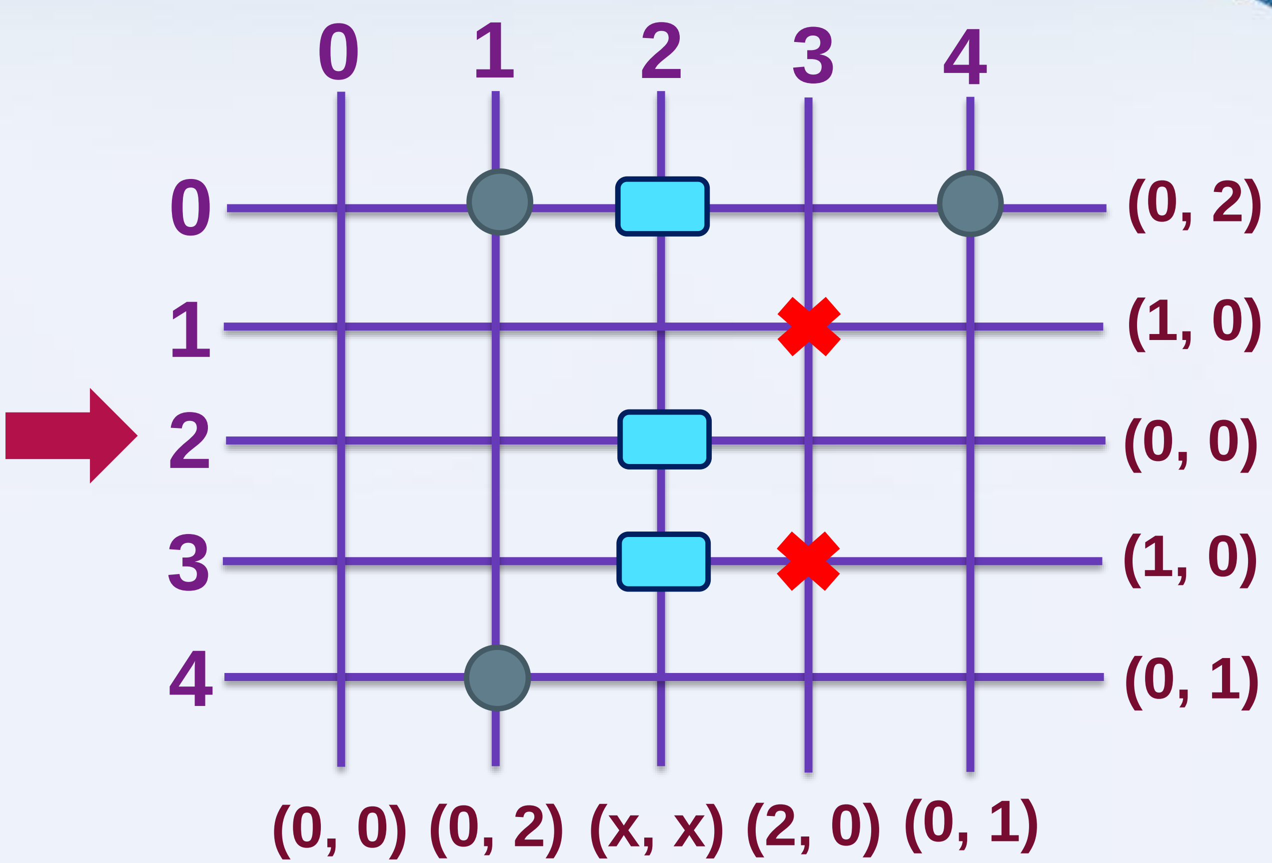
Objective :

1. Find the maximum size of rectangle including Open Defect
2. Eliminate Closed Defect

Example

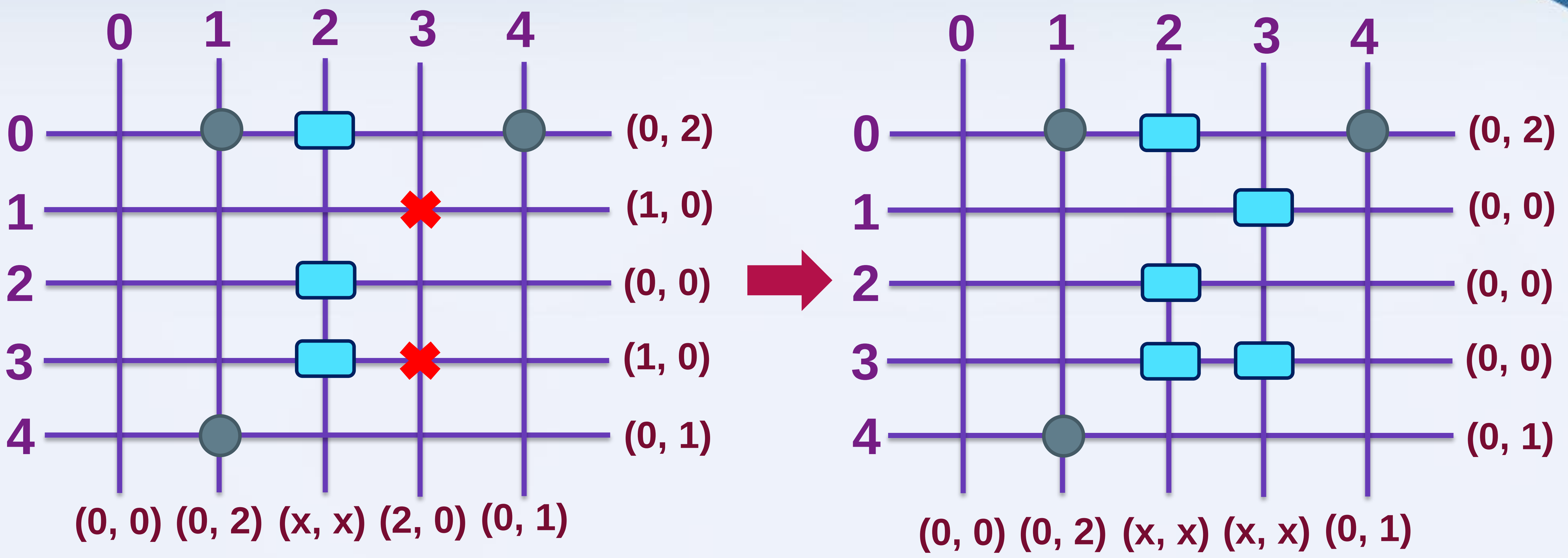


0
1
2
3
4
0
1
2
3
4
(1, 2)
(1, 0)
(1, 0)
(1, 1)
(0, 1)
(0, 0) (0, 2) (2, 1) (2, 0) (0, 1)



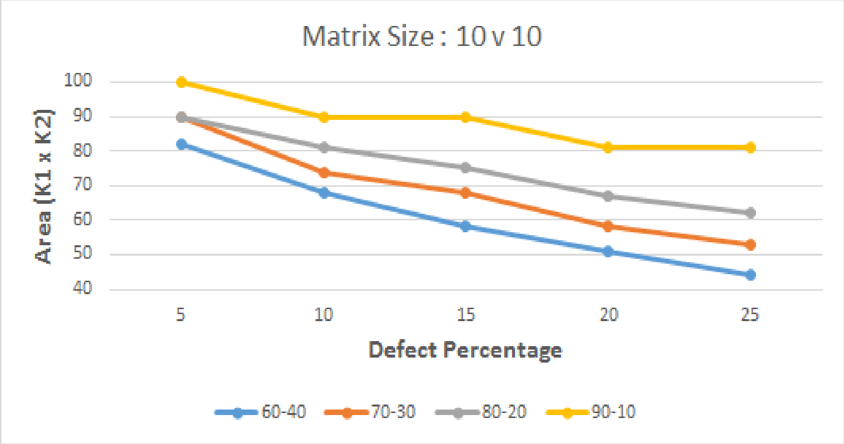
0
1
2
3
4
0
1
2
3
4
(0, 2)
(1, 0)
(0, 0)
(1, 0)
(0, 1)
(0, 0) (0, 2) (x, x) (2, 0) (0, 1)

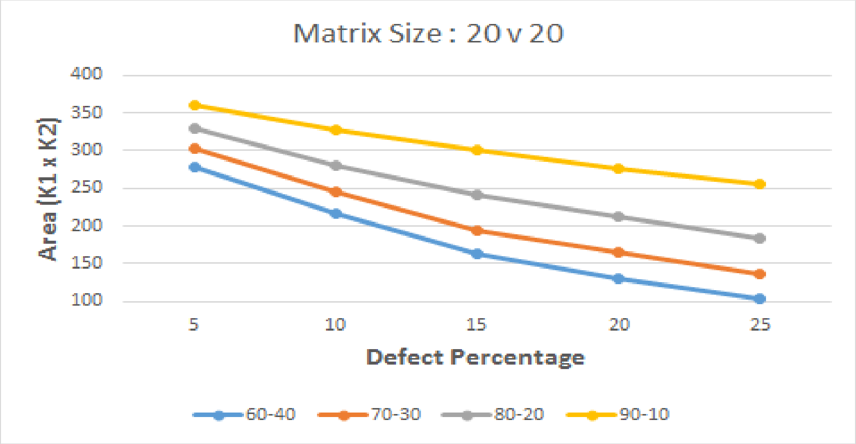
Example

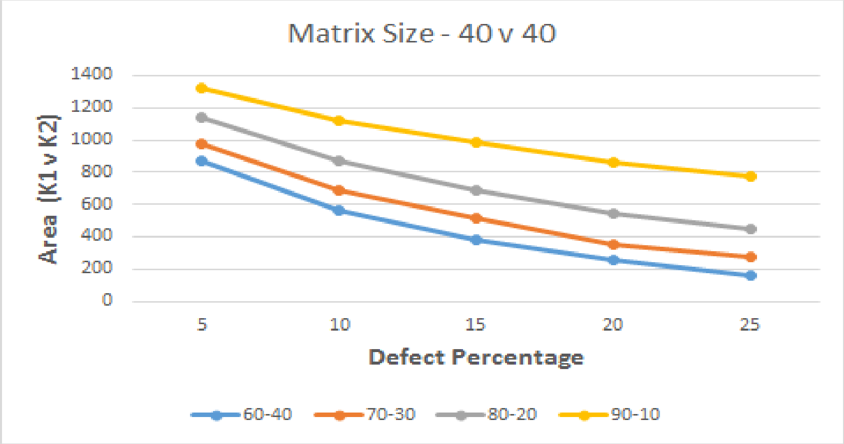

0 1 2 3 4
0 (0, 2)
1 (1, 0)
2 (0, 0)
3 (1, 0)
4 (0, 1)
(0, 0) (0, 2) (x, x) (2, 0) (0, 1)
0 1 2 3 4
0 (0, 2)
1 (0, 0)
2 (0, 0)
3 (0, 0)
4 (0, 1)
(0, 0) (0, 2) (x, x) (x, x) (0, 1)

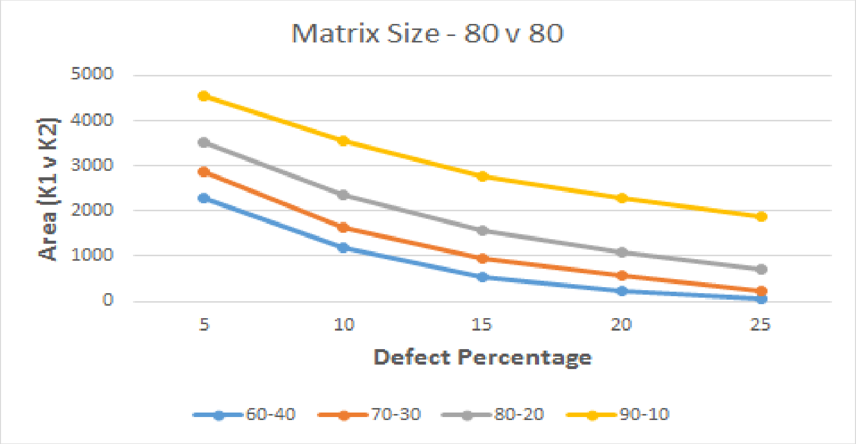
Comparison

1. Matrix-wise





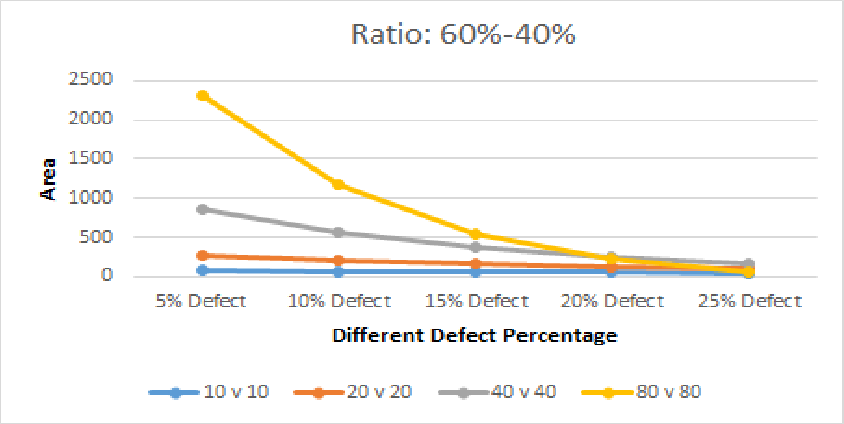


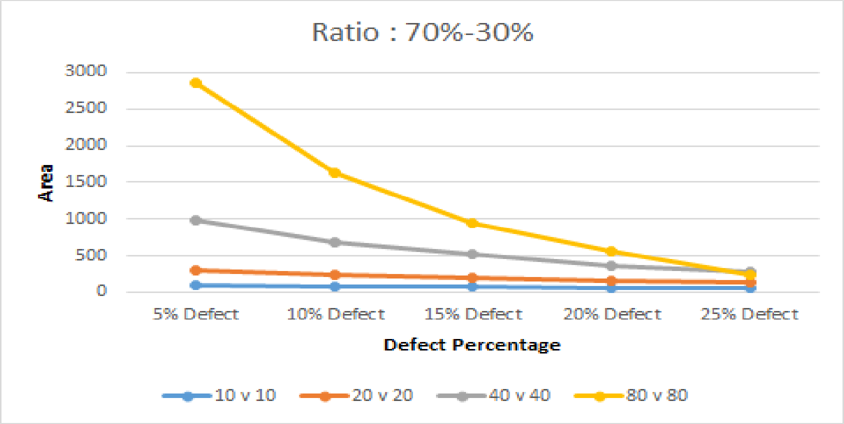


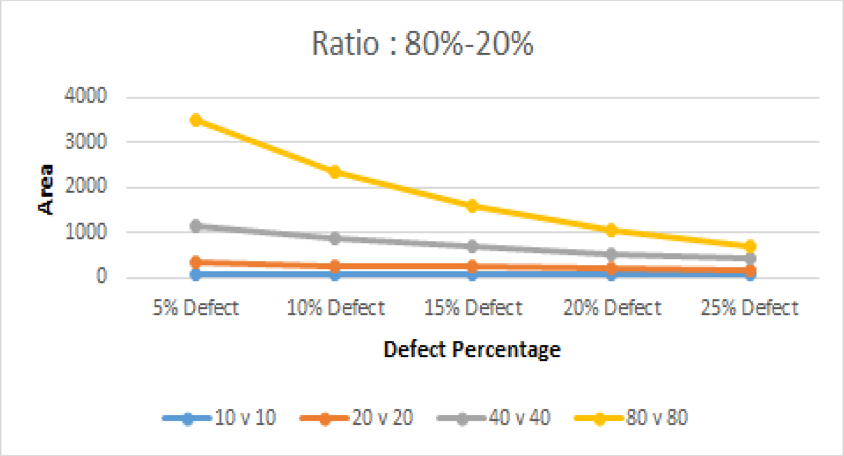
Area with only Stuck Open Defect Vs. Defect Percentage Graph for Different Open : Closed Ratio

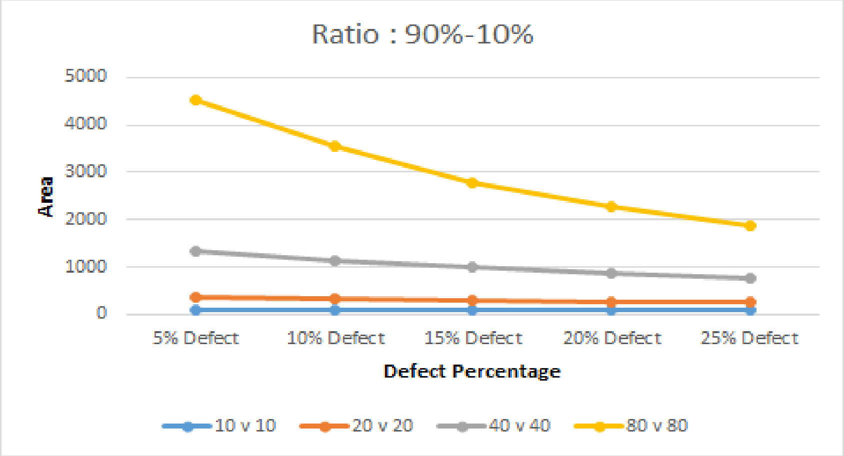
Comparison

2. Defect Ratio-wise









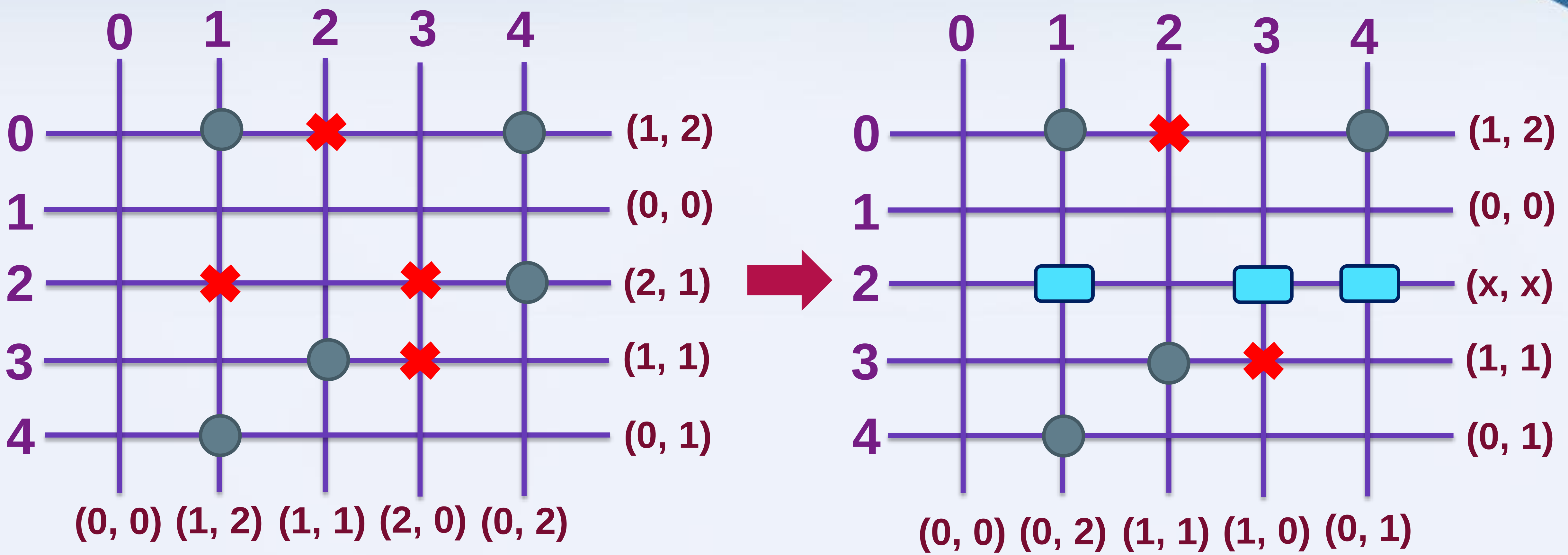
Area with only Stuck Open Defect Vs. Different Defect Percentage Graph for Different Matrix Size

Algorithm – K x K

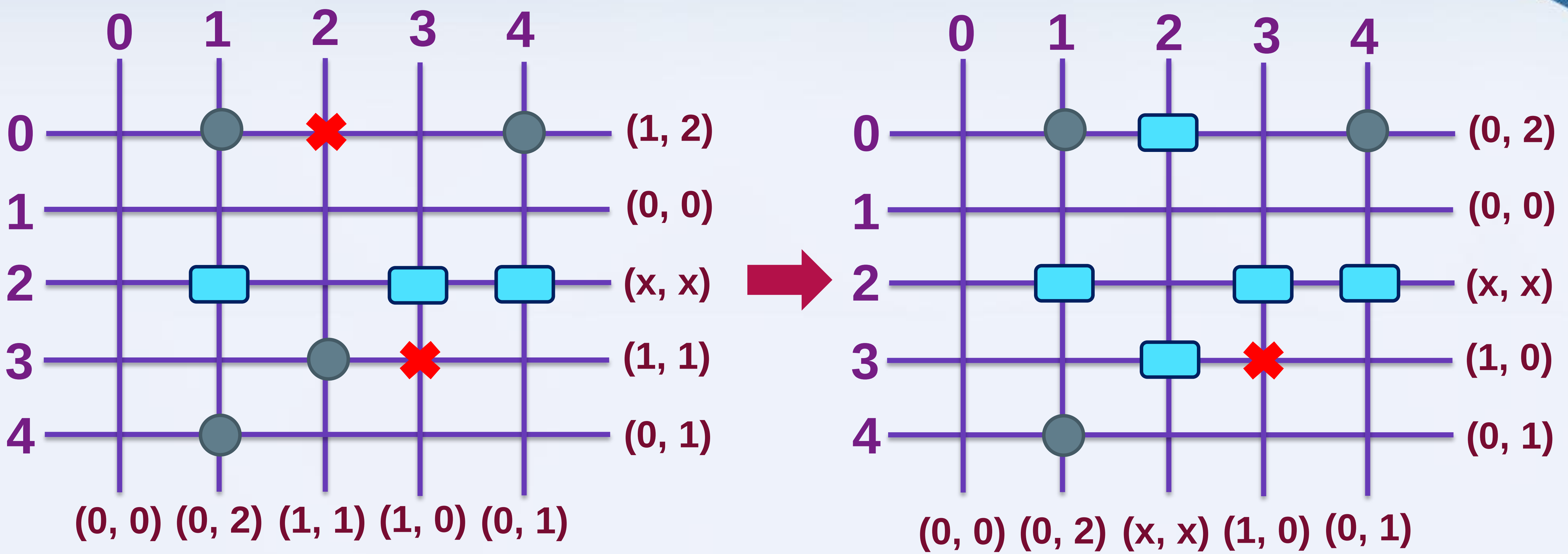
Objective :

1. Find the maximum size of rectangle including Open Defect
2. Eliminate Closed Defect
3. Better than K1 x K2 approach

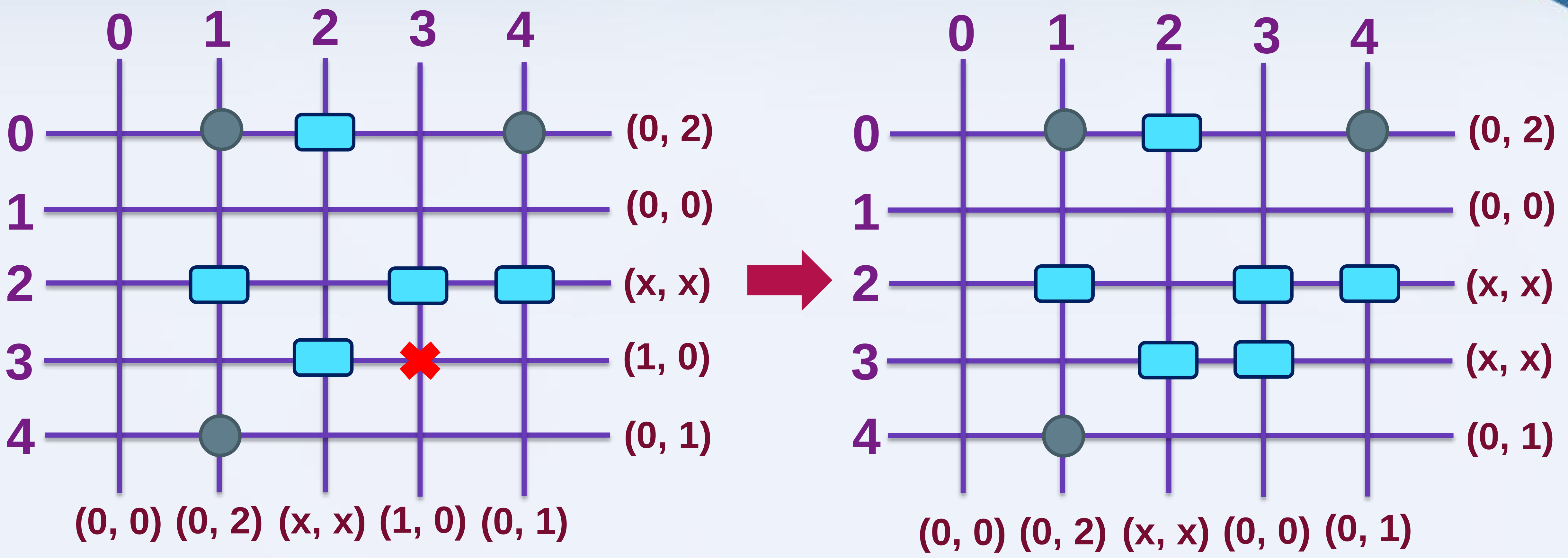
Example


0 1 2 3 4
0 1 2 3 4
(1, 2)
(0, 0)
(2, 1)
(1, 1)
(0, 1)
(0, 0) (1, 2) (1, 1) (2, 0) (0, 2)
0 1 2 3 4
0 1 2 3 4
(1, 2)
(0, 0)
(x, x)
(1, 1)
(0, 1)
(0, 0) (0, 2) (1, 1) (1, 0) (0, 1)

Example


0 1 2 3 4
0 1 2 3 4
(1, 2)
(0, 0)
(x, x)
(1, 1)
(0, 1)
(0, 0) (0, 2) (1, 1) (1, 0) (0, 1)
0 1 2 3 4
0 1 2 3 4
(0, 2)
(0, 0)
(x, x)
(1, 0)
(0, 1)
(0, 0) (0, 2) (x, x) (1, 0) (0, 1)

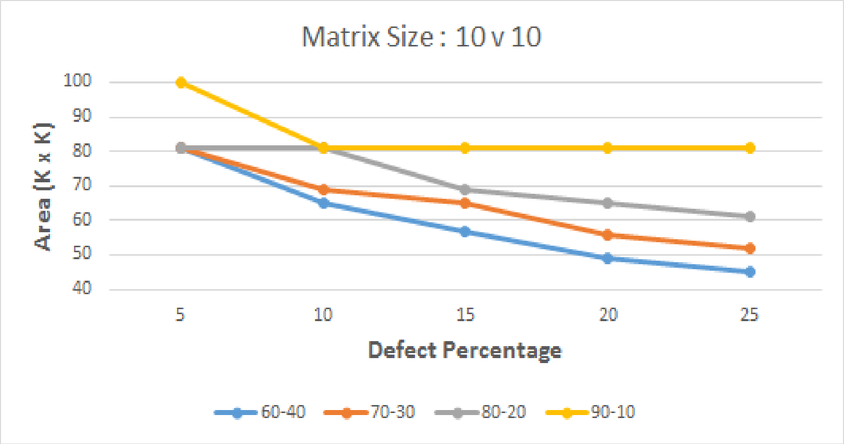
Example

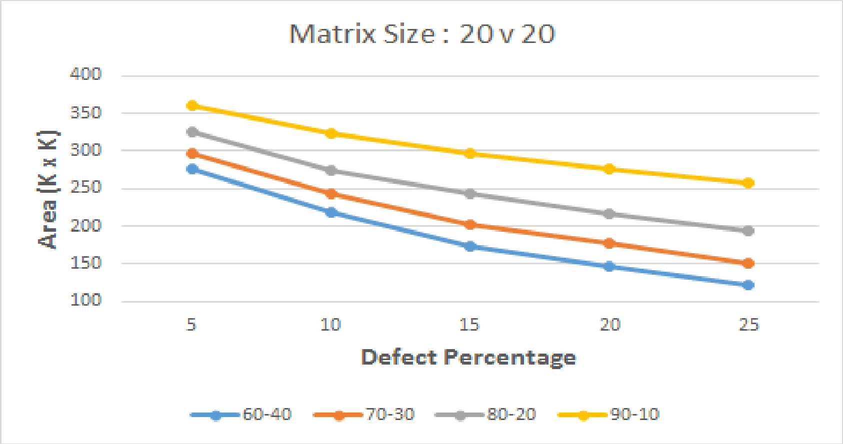


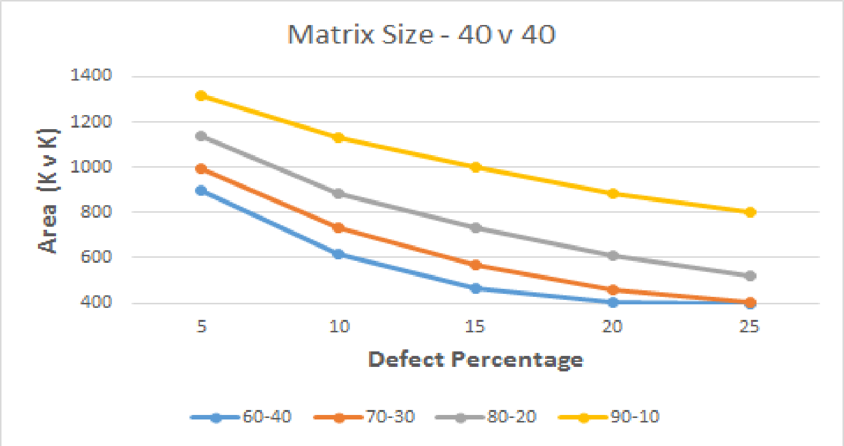
0
1
2
3
4
(0, 2)
(0, 0)
(x, x)
(1, 0)
(0, 1)
(0, 0) (0, 2) (x, x) (1, 0) (0, 1)
0
1
2
3
4
(0, 2)
(0, 0)
(x, x)
(x, x)
(0, 1)
(0, 0) (0, 2) (x, x) (0, 0) (0, 1)

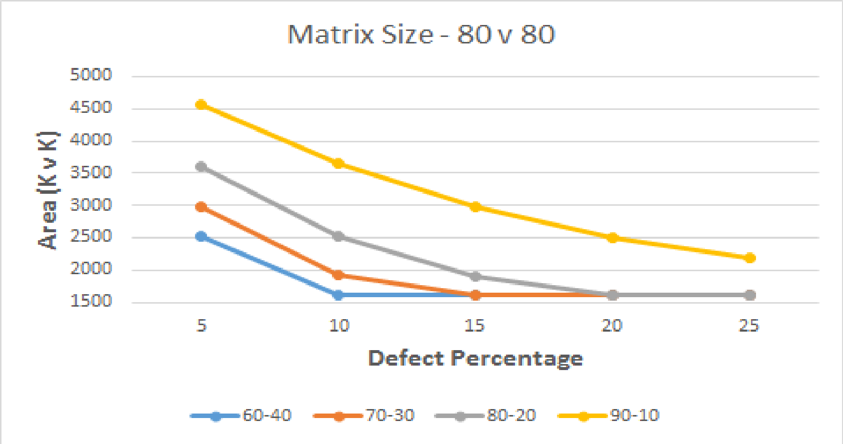
Comparison

1. Matrix-wise









Area with only Stuck Open Defect Vs. Defect Percentage Graph for Different Open : Closed Ratio

Comparison

2. Defect Ratio-wise

Area with only Stuck Open Defect Vs. Different Defect Percentage Graph for Different Matrix Size

K1xK2 vs KxK

1. **Unequal Rows and Columns**

2. **Balancing of Input and Output**

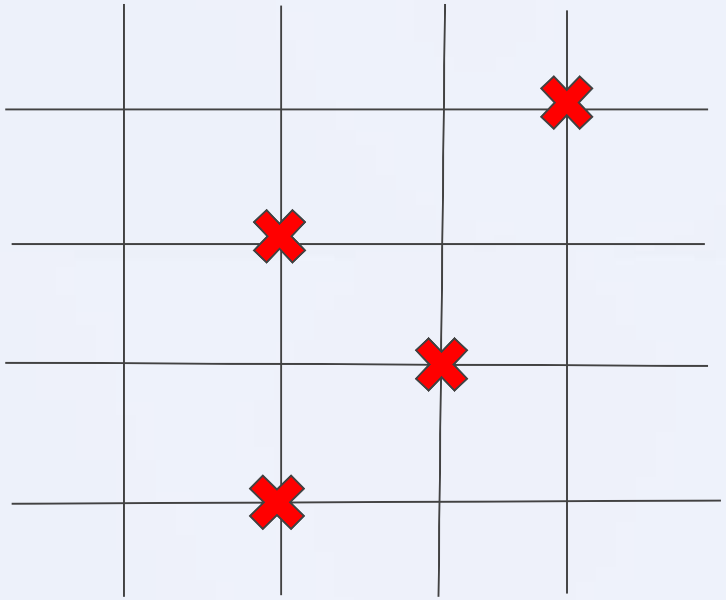
Maximum Empty Rectangle Problem

Objective:

1. Find a rectangle which can tolerate the defect

2. Finding a defect-free solution

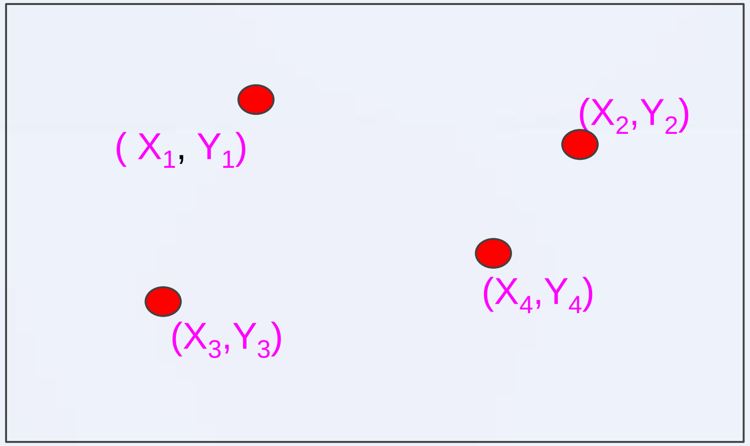
Example



This is a given nxn crossbar circuit with some defect points with red cross indications.

Example

We are considering the above circuit as:



Here we consider those defect points as those red points with a coordinate

Method in a nutshell

1. In every iteration we find the maximum empty rectangle avoiding the defect points.

2. Then we exclude it from the given crossbar circuit or the rectangle

3. Goto step 1 until we get a small rectangle (as for example 2x2 or 5x5)

Analysis

1. According to [6],time complexity to find a maximum empty rectangle is $O(n^2)$.

2. Now according to our method the above iteration goes from n to a very small number

Hence, the time complexity is $O(n^3)$.

Why this method is used?

1. In many methods, to find a defect-free region every defect point is taken into account.

2. In this case, a region is considered rather than considering each point.

3. Gives more reliable result

Disadvantage

1. It does not differentiate stuck-open and stuck-close faults.

2. It has higher time complexity.

Graham Scan Approach

Objective:

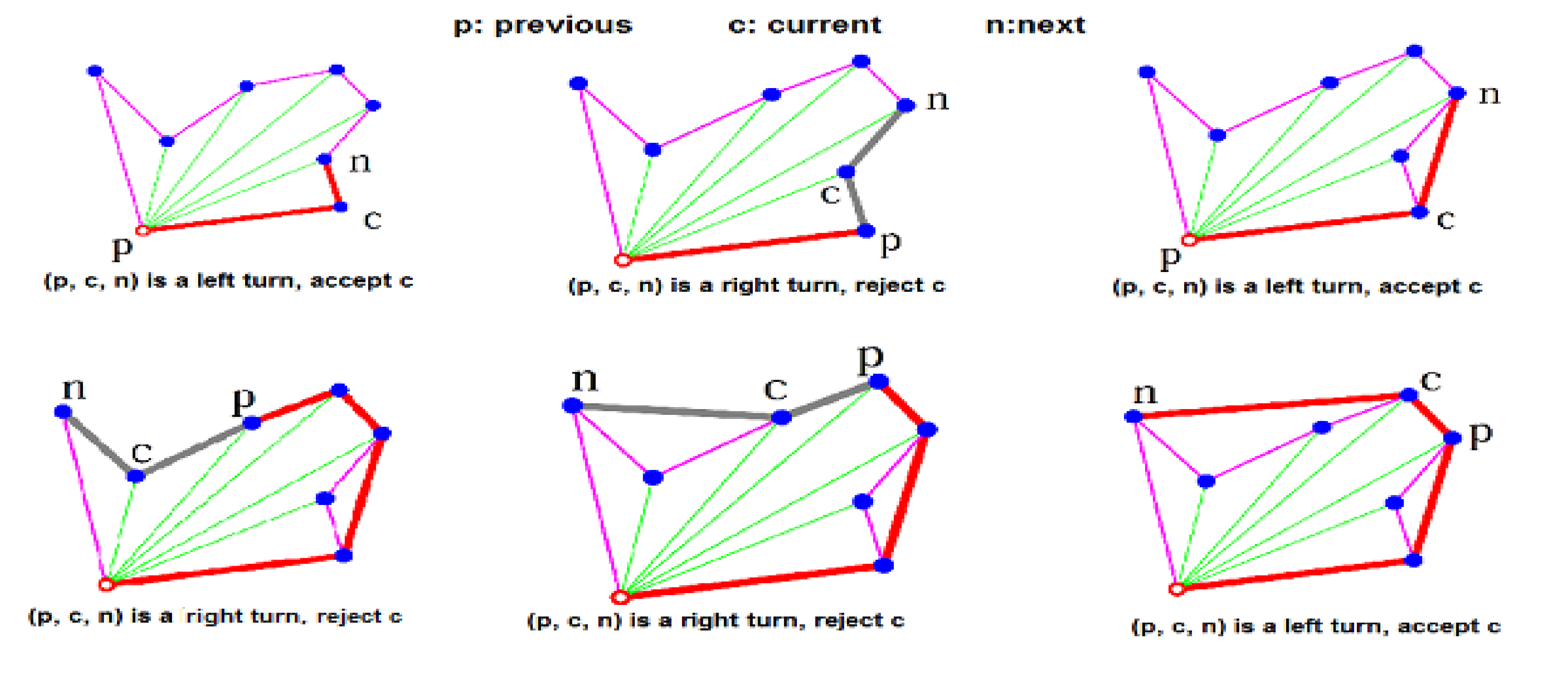
1. Find a smallest convex polygon which contains all the defect points.

2. Complement the convex hull of set to get the defect-free region.

Algorithm

1. Find the point(p0) with lowest ordinate(or lowest abscissa in case of a tie).
2. Sort them in ascending polar angles w.r.t to p0.
3. Choose first three points pushed them into the stack.
4. Choose remaining points one by one and check if it is a right turn.
5. If it is a right turn keep removing points from the stack until it is a left turn.
6. Push the point into the stack and goto step 4.

Explanation



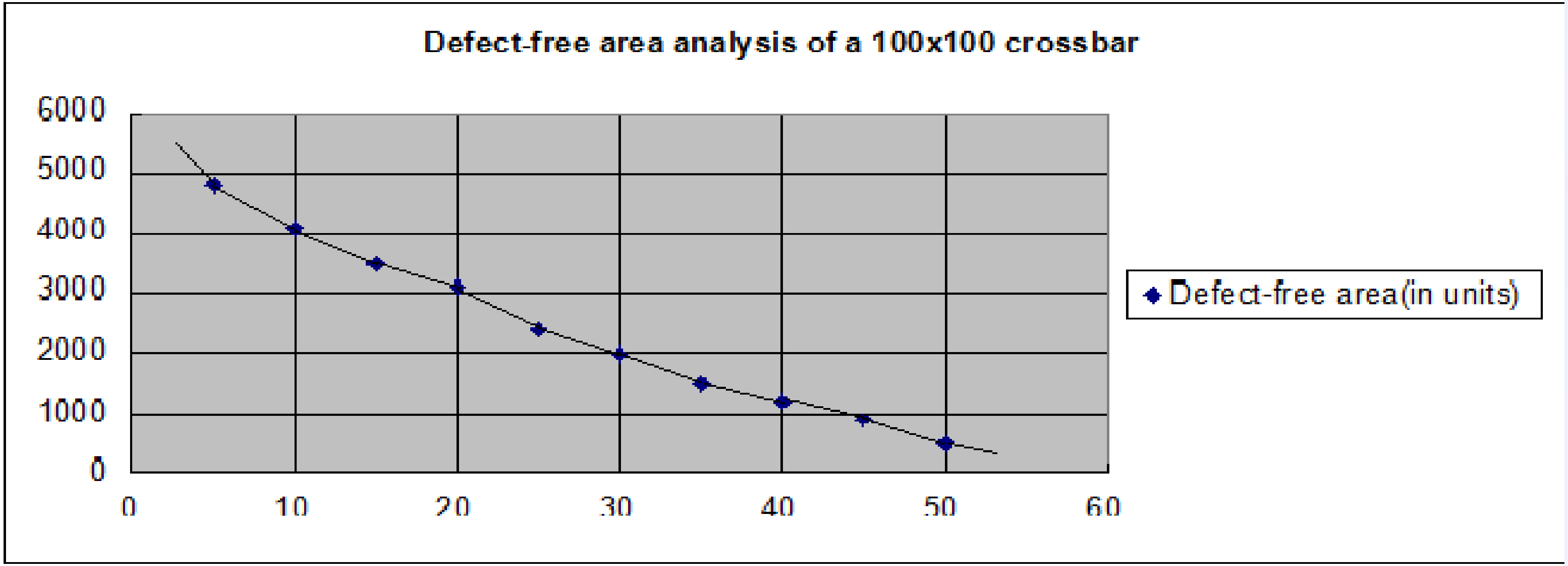
Analysis

1. Finding the bottom-most point takes $O(n)$ time.
2. Sorting takes $O(n \log n)$ time
3. For rest of the part it takes $O(n)$.

Hence, the overall complexity is $O(n)+O(n \log n)+O(n)$

$$= O(n \log n)$$

Analysis(Cont..)



X-axis: Defect Rate(in percentage) *Y-axis: Defect-free area (in units)*

Advantages

1. The runtime of this algorithm is far better than previous algorithms.

2. The Defect-free area we get by this algorithm is larger than the previous algorithms.

Disadvantages

1. In this algorithm we assumed that the defects are always in clusters. So, in case of scattered defects this algorithm fails. But, if we find different regions of clusters we can be able to apply the algorithm in different regions and enclosed all the defect points in different convex sub-hulls.

2. There are some functional parts enclosed in the polygon.

Conclusion

- Providing the measure of goodness of a crossbar with defect is very important.
- We proposed our algorithms on the basis of getting better solution than previous algorithms.
- We manage to overcome a solution with good time complexity but without considering extreme cases.

References

1. J. Huang, M. B. Tahoori, and F. Lombardi, “***On the defect tolerance of nanoscale two-dimensional crossbars***,” in *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Syst.*, 2004

2. M. M. Ziegler and M. R. Stan, “***Design and analysis of crossbar circuits for molecular nanoelectronics***,” in Proc. IEEE Int. Conf. Nanotechnology, 2002

3. Ahmad A. Al-Yamani, Sundarkumar Ramsundar and Dhiraj K. Pradhan, “***A Defect Tolerance Scheme for Nanotechnology Circuits***”, in Proc. IEEE Int. Conf. 2007

4. Bo Yuan and Bin Li, University of Science and Technology of China, “***A fast extraction Algorithm for Defect-Free Subcrossbar in Nanoelectronic Crossbar***”, ACM Journal Emerging Technology Computer System, April 2014

5. Malay Kule, Hafizur Rahaman, Bhargab B. Bhattacharya, “***On Finding a Defect-free Component in Nanoscale Crossbar Circuits***”, 4th International Conference on Eco-friendly Computing and Communication Systems (ICECCS 2015)

6. A. NAAMAD, D.T. LEE, “***On the Maximum Empty Rectangle Problem***”, (1984)

THANK YOU