SAT solver performance on Monochromatic triangle

Frieda Musa
Ramezani Yousef

1 Introduction

The problem of a Monochromatic triangle is defined as follows:

Monochromatic triangle problem is an algorithmic problem on graphs, the goal is to partition the edges of a given graph into two triangle-free sub-graphs. It is NP-complete but has a fixed-parameter tractable on the size of the largest vertex set in an undirected graph[Wikb].
S $\subset$ V such that there is no triangle in S and V_i
there is no triangle in a graph $\exists V_i, V_j, V_k$
if there is a triangle such that there is an edge E $(V_i, V_j) \wedge E(V_j, V_k) \wedge E(V_k, V_i)$

Instance: Monochromatic triangle takes an instance of n-node undirected graph (V,E) where V is the Vertex set and E is the Edge set.
Acceptance Condition:
It accepts TRUE Boolean value if there is a set of edge E of G can be partitioned into two disjoint sets E1 and E2 such that both sub-graphs G1(V, E1) and G2 (V,E2) $\langle G_1(V, E_1), G_2(V, E_2)\rangle$ are triangle free graphs (as shown in the diagram below) and rejects if otherwise.

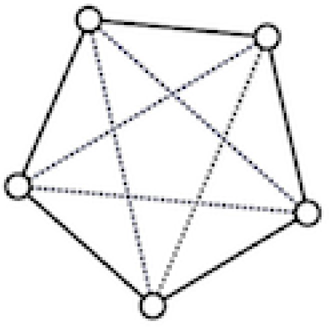

Figure 1: Monochromatic Triangle

The importance of a Monochromatic Triangle is to ensure that in a given graph G, there will be no triangle formed This decision problem is in NP-complete and is in connection

with Ramsey's Theory [Wikc] By Ramsey's theorem, for any finite number k of colors, there exists a number n such that complete graphs of n or more vertices do not have triangle-free edge colorings with k colors. If k = 2, the corresponding value of n is 6. I.e., the answer to the monochromatic triangle problem on the complete graph K6 is no.
This experiment was conducted by the following steps: generating a random graph, reduction of the problem, running instances and values of n on the solvers and finally analyzing the results found.

2 Reducing Monochromatic Triangle to SAT

We reduced Monochromatic Triangle to SAT as follows:

Variables: $x_i, ..., x_n$ is the variable such that i is the rightmost side of the partition
x_i is true $\Leftrightarrow$vertex $V_i \in$ S
Differentiate between vertices, a variables that tells the vertex is in a set. If there is a partition and there is no such assignment than there is no partition. The assignment tells us where the assignment S is.
The output from the satisfying assignment $x_1, \ldots, x_n$ take S to be $\{v_i \,|x_i\}$ and set it to true by this assignment.
For each triangle (V_i, V_j, V_k) we want a condition that there is at least one variable in S and at least one variable not in S no matter the triangle you pick from.

Constraints: Constraint of type 1
For all vertices there should be at least one vertex in S and one Vertices not in S because we want both side to be in the Triangle. No matter the vertices (3 Vertices) picked in the triangle. For each triangle in the rightmost side it takes $O(n^3)$ still in polynomial to the size of the input to check the given vertices in a graph.
And at least one of them is true in S $(x_i \vee x_j \vee x_k)$
And at least one of them is false not in S $(\neg x_i \vee \neg x_j \vee \neg x_k)$
Therefore the reduction is $\forall V_i, V_j, V_k$ such that $(x_i \vee x_j \vee x_k)$ $\wedge(\neg x_i \vee \neg x_j \vee \neg x_k)$

3 Generating random instances of Monochromatic Triangle

A random Graph will be used to generated using Erdos-Renyi[Wika] Model where each pair of vertices is connected to a probability P, as the existing link connecting vertices.
The predicted number of edges is n*(n-1)/2
The size of the graph n is the number of vertices
If P(u) is close to 1 the generator produce yes
If P(u) is close to 0 the generator produce no
A random graph is more likely to give a "no" and check if the graph can be partitioned

to form two different disjoint sets. To ensure that the instance of our problem is ”yes” a preferential attachment is introduced

Random graph was generated using an adjacency matrix where there are three for loops to create an edge between the vertices in the graph. With the algorithm below.

```
1. Set counter = 0
2. Given p as the probability represented by edge connecting nodes
3. Given n as the number of vertices
4. For i = 1 to n
5. For j = 1 to n
6. For k = 1 to n
7. Check if $E(i,j) \wedge E(j,k) \wedge E(i,k)$
8. Increment counter by 1
9. Save L = $(x_i \vee x_j \vee x_k) \wedge(\neg x_i \vee \neg x_j \vee \neg x_k)$
10.Print counter and n
11.For each line in L
12.Generate $G(p,n)$
```

4 Experimental results

System Specifications
Operating system: Ubuntu 16.04 LTS
Memory: 16 GB
Processor: Intel Core^{TM} i5-3470 CPU @ 3.20Hz x 4
Graphics: Intel Ivybridge Desktop
OS type: 64-bit
Hard Disk: 68.6 GB
Solver: Minisat Version 2.0 and Glucose syrup 4.1

For 20 values of n, we generated 10 instances each using the monochromatic triangle GENERATOR and CNF converter program[FM],with n having the range of 50 starting from 0 to 500. It is shown in the graphs above that as P increases the execution time increase and the chance of getting satisfiable variables for each value of n. As p tends to get to 1 it finds the existence of triangles in the set of n values. The curve shows that the problem runs in exponential time to the number of vertex.

The values of n less than 15 were satisfiable but as n increased above that, the solver output unsatisfiable because it was more difficult not to form a triangle. Satisifiable variables run faster than unsatisfiable variables since they are found in small values of vertices.

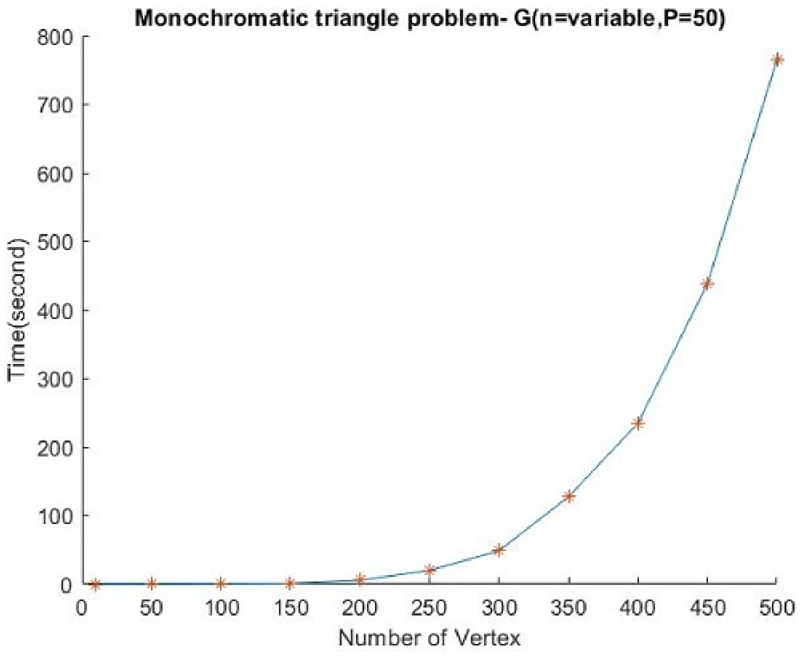

Figure 2: Monochromatic Triangle problem with time by n = 0 to 500

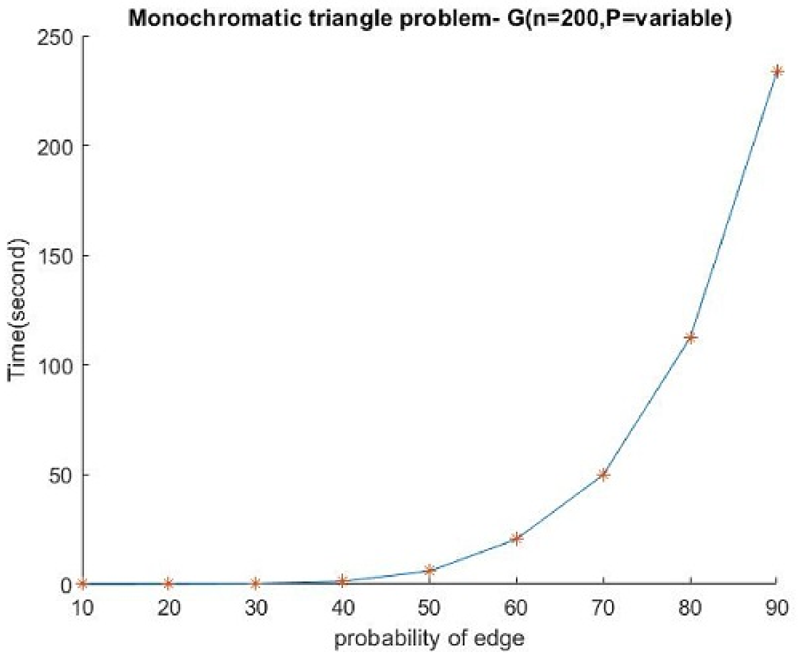

Figure 3: Monochromatic Triangle problem with time by 10 instance of probability

5 Additional experiments

Preferential Attachment is used to further analyze the behaviour of the vertices with high degree where the richer vertex keeps getting richer[Cou]. From the beginning where two vertices are connected by an edge. Every time a new vertex is added with an edge connecting to existing vertex. The node is connected with the vertex that has the highest probability proportional to each vertex's degree. the probability of connecting to a vertex u of degree k_u is k_u / Σ_j k_u. Unfortunately, we were unable to generate graphs to further analysis this model.

6 Conclusion

With parameters staring from 0 to 500 for values of n and 10 instances of Probability. Minisat solver and Glucose solver ran .cnf files effectively on the cluster computer but took longer execution time on my personal computer.

References

[Cou] Cousera.

[FM] Yousef R Frieda M. Graphgenerator.

[Wika] Wikipedia. Erds-rnyi.

[Wikb] Wikipedia. Monochromatic triangle.

[Wikc] Wikipedia. Ramsey theorem.