Constraint satisfaction problem
Graph Coloring

Butnaru Gheorghita

1 Introduction

The present paper presents some statistics related to the CSP (Constraint Satisfaction Problem) - Graph Coloring. The optimization focuses on binary constraints imposed by the condition that adjacent nodes cannot have the same color.

2 Algorithms

The main algorithm used in this study is **Backtracking Search**, which uses **depth-first search** to select a variable for coloring. In addition, several heuristics were applied to optimize the algorithm:

1. MRV (Most Remaining Values) - It picks a variable (node) with the fewest "legal" values (smallest domain of colors). This strategy aims to reduce the branching factor of the search tree by prioritizing variables that have fewer options for assignment. By selecting the variable with the most limited choices first, the algorithm can potentially prune large portions of the search space and reach a solution more efficiently.

2. FW (Forward Checking) - Whenever a variable X is assigned, the forward-checking process establishes arc consistency for it: for each unassigned variable Y that is connected to X by a constraint, delete from Y 's domain any value that is inconsistent with the value chosen for X. The forward checking heuristic includes following advantages: early conflict detection, efficient search tree pruning, improved constraint propagation, and earlier detection of infeasible solutions.

3. Degree - Selects the variable that is most constrained, that is the variable with the highest number of adjacent vertices, which tends to reduce the branching factor and increase the likelihood of finding a feasible solution faster.

3 Statistics

Four randomly generated test instances were used for the statistical analysis.

3.1 Instance 1 - 50 nodes, 100 edges, 226 binary constraints

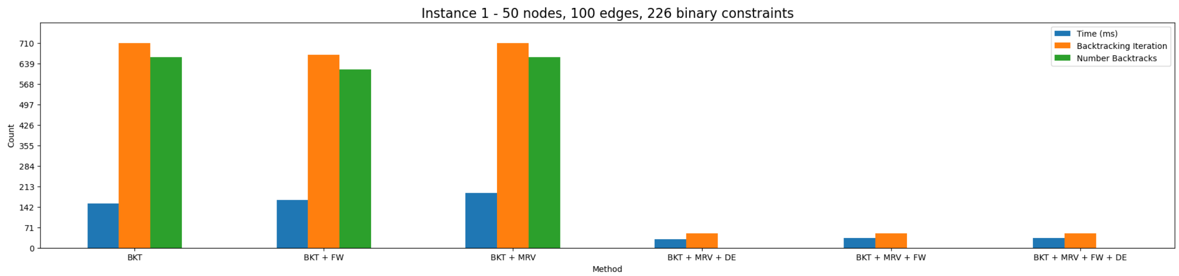

Figure 1: Instance 1.

Table 1: Description of Results for Instance 1

Method	Backtracking Iteration	Number Backtracks	Time (ms)
BKT	710	660	154
BKT + FW	669	619	165
BKT + MRV	710	660	191
BKT + MRV + DE	50	0	30
BKT + MRV + FW	50	0	33
BKT + MRV + FW + DE	50	0	34

We can observe that the **worst** method in terms of backtracking iteration, number of backtracks and time execution is BKT.

The **best** method is BKT + MRV + DE, followed by BKT + MRV + FW and BK + MRV + FW + DE.

3.2 Instance 2 - 80 nodes, 160 edges, 407 binary constraints

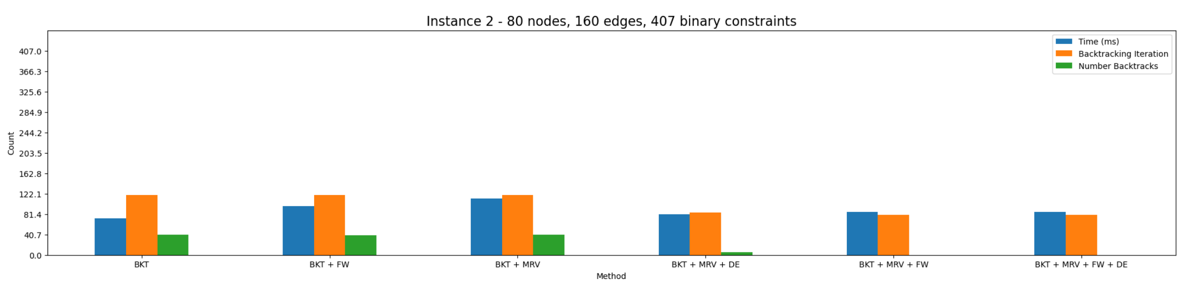

Figure 2: Instance 2

Table 2: Description of Results for Instance 2

Method	Backtracking Iteration	Number Backtracks	Time (ms)
BKT	120	40	73
BKT + FW	119	39	97
BKT + MRV	120	40	113
BKT + MRV + DE	85	5	81
BKT + MRV + FW	80	0	86
BKT + MRV + FW + DE	80	0	86

We can observe:

- The **worst** method is BKT.
- The **best** method is BKT + MRV + FW and BKT + MRV + FW + DE.

3.3 Instance 3 - 100 nodes, 200 edges, 509 binary constraints

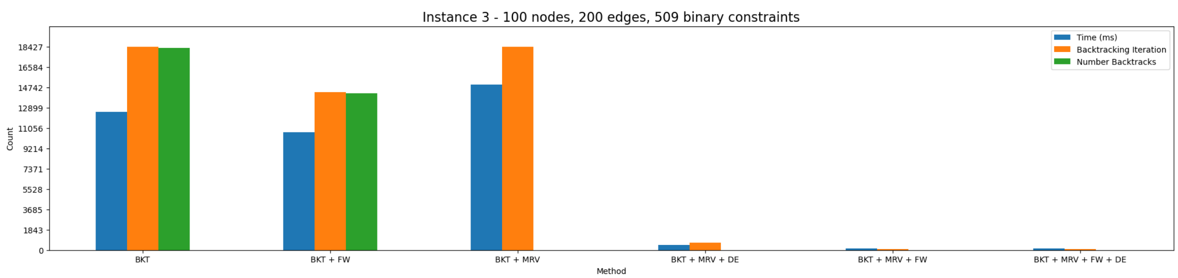

Figure 3: Instance 3

Table 3: Description of Results for Instance 3

Method	Backtracking Iteration	Number Backtracks	Time (s)
BKT	18427	18327	12.512
BKT + FW	14293	14193	10.657
BKT + MRV	18427	0	14.988
BKT + MRV + DE	684	0	0.453
BKT + MRV + FW	100	0	0.121
BKT + MRV + FW + DE	100	0	0.123

We can observe:

- The **worst** method is BKT.
- The **best** method is BKT + MRV + FW followed by BKT + MRV + FW + DE

3.4 Instance 4 - 100 nodes, 280 edges, 662 binary constraints

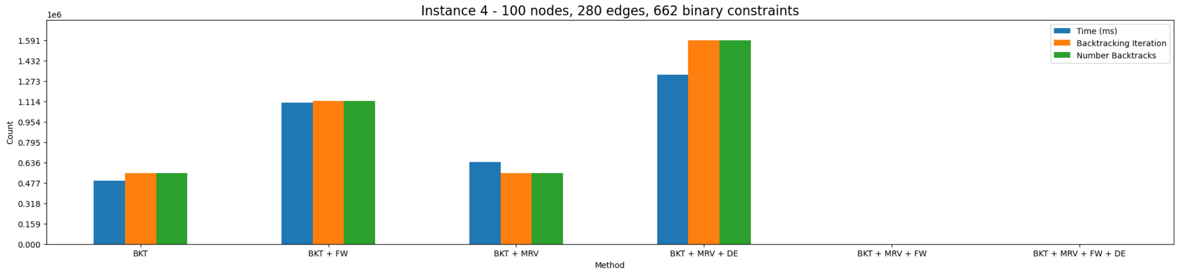

Figure 4: Instance 4

Table 4: Description of Results for Instance 4

Method	Backtracking Iteration	Number Backtracks	Time (m)
BKT	553753	553653	8.265
BKT + FW	1118034	1117934	18.40
BKT + MRV	553753	553653	10.67
BKT + MRV + DE	1590768	1590668	22.039
BKT + MRV + FW	105	5	0.00255 (153 ms)
BKT + MRV + FW + DE	116	16	0.00293333 (176 ms)

We can observe:

- The **worst** method is BKT + MRV + DE.
- The **best** method is BKT + MRV + FW followed by BKT + MRV + FW + DE

3.5 Instance 5 - 500 nodes, 1000 edges, 2706 binary constraints

The BKT method encountered a significant challenge with 2706 constraints. Despite running for 6 hours and 30 minutes and reaching iteration 1,000,000, it was unable to complete and had to be stopped. At that point, it had already accumulated 999,626 backtracks.

Based on the given information, it is recommended to focus on two methods: BKT + MRV + FW and BKT + MRV + FW + DE. These methods have shown the best results thus far and should be given priority over other methods.

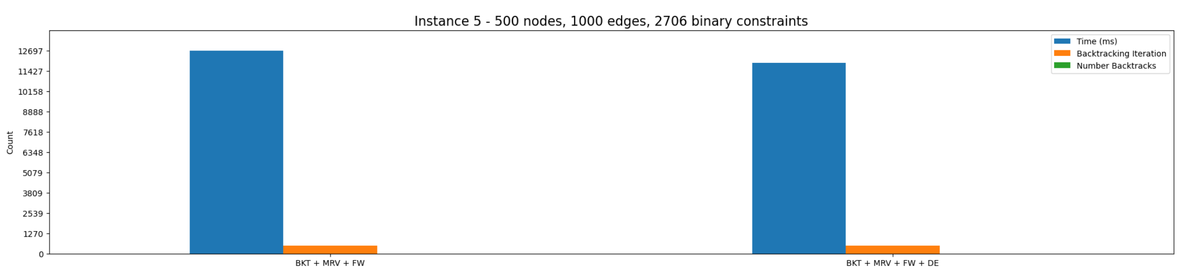

Figure 5: Instance 5

Table 5: Description of Results for Instance 5

Method	Backtracking Iteration	Number Backtracks	Time (s)
BKT + MRV + FW	500	0	12.70
BKT + MRV + FW + DE	500	0	11.93

Considering these two methods, it is evident that both BKT + MRV + FW and BKT + MRV + FW + DE achieved similar performance. The only discernible distinction is that BKT + MRV + FW + DE completed the task slightly faster, with a time difference of nearly 1 second compared to the BKT + MRV + FW method.

4 Conclusion

In conclusion, this study analyzed the performance of various algorithms and heuristics for solving the Constraint Satisfaction Problem (CSP) - Graph Coloring. The main algorithm used was Backtracking Search, complemented by heuristics such as Most Remaining Values (MRV), Forward Checking (FW), and Degree (DE). The analysis was conducted on five test instances with varying sizes and numbers of constraints.

The results demonstrated that the choice of algorithm and heuristics significantly impacted the performance in terms of backtracking iterations, number of backtracks, and execution time. The BKT method alone showed the worst performance, while the methods incorporating MRV and FW consistently outperformed others.

Among the tested methods, BKT + MRV + FW and BKT + MRV + FW + DE emerged as the most effective approaches for solving graph coloring problems. These methods consistently achieved lower backtracking iterations and execution times across different instances.

Furthermore, the study revealed that as the size and complexity of the problem increased, the performance gap between the best and worst methods became more pronounced. In the largest instance, the BKT method struggled to complete the task even after running for several hours, highlighting the limitations of a basic backtracking approach for complex problems.

Based on these findings, it is recommended to prioritize the use of BKT + MRV + FW and BKT + MRV + FW + DE methods for graph coloring problems. These approaches offer a balance between effectiveness and efficiency, delivering satisfactory results for a wide range of instances.