Report - Empirical Algorithm Analysis

Bartosz Michalak

June 2025

1 Introduction

This report presents the implementation and experimental evaluation of a 2-approximation algorithm for the problem of scheduling on unrelated parallel machines with the makespan criterion. The algorithm is based on linear programming and follows the approach described in Algorithm 17.5 from the book *Approximation Algorithms* by Vijay V. Vazirani. The implementation was done in the Julia programming language using the JuMP optimization library and HiGHS solver.

The quality of the approximation algorithm was evaluated using benchmark instances from the RCmax dataset, which includes a wide range of scheduling instances previously used in the literature. The goal of this study is to assess the practical performance of the approximation algorithm in terms of solution quality and computational efficiency.

2 Plots

2.1 1a100

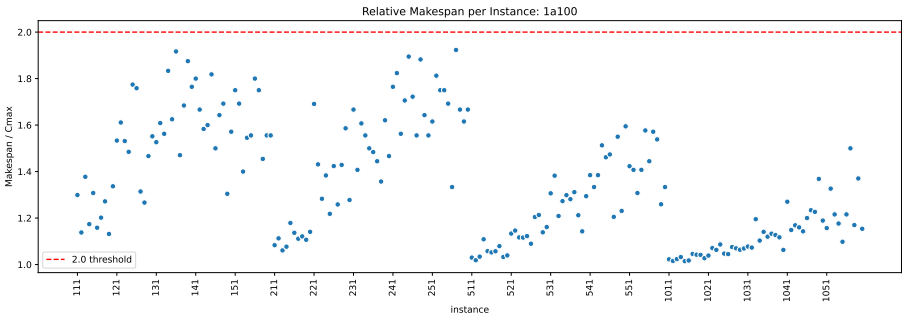

Figure 1: Approximation ratio

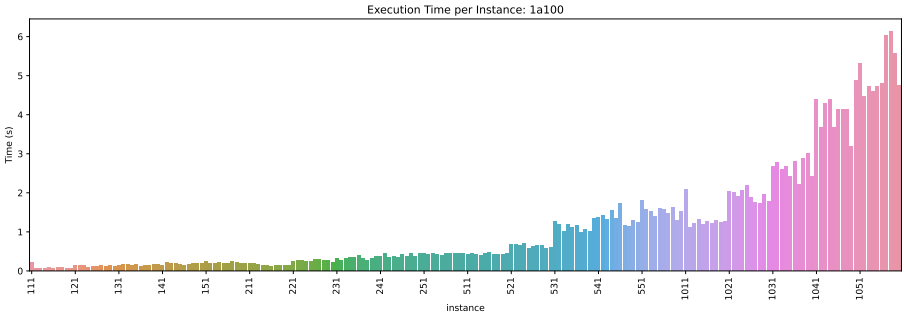

Figure 2: Execution Time

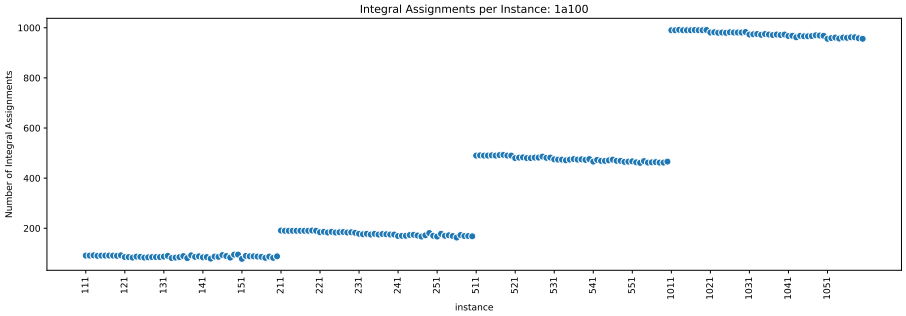

Figure 3: Integer Assignments

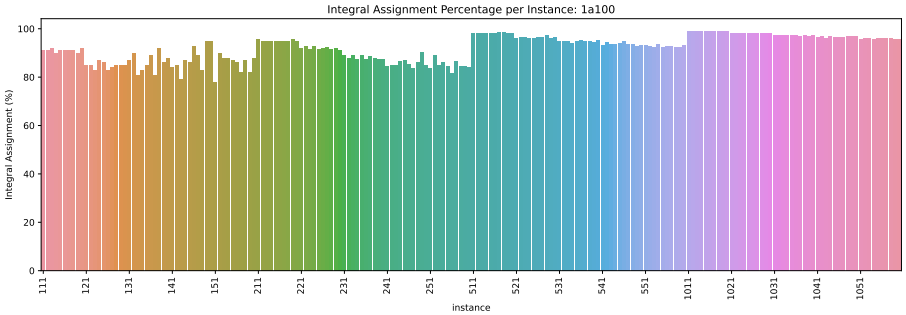

Figure 4: Integer Assignments %

2.2 100a120

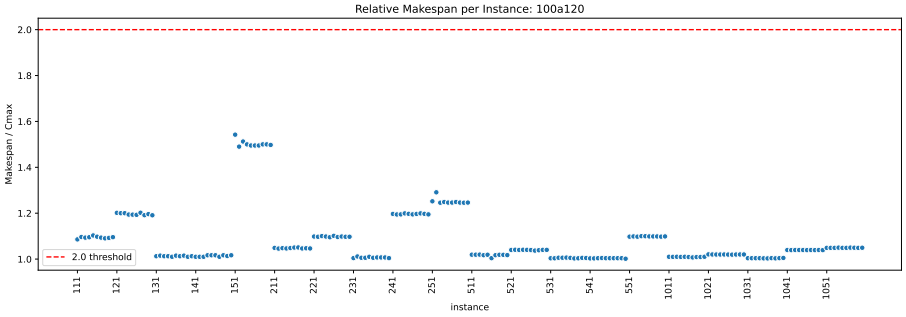

Figure 5: Approximation ratio

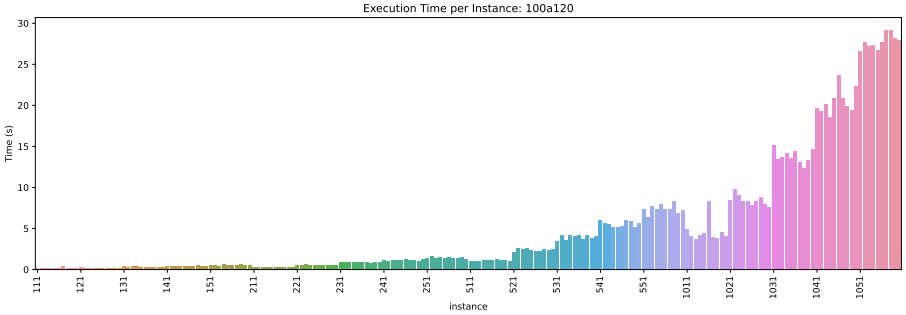

Figure 6: Execution Time

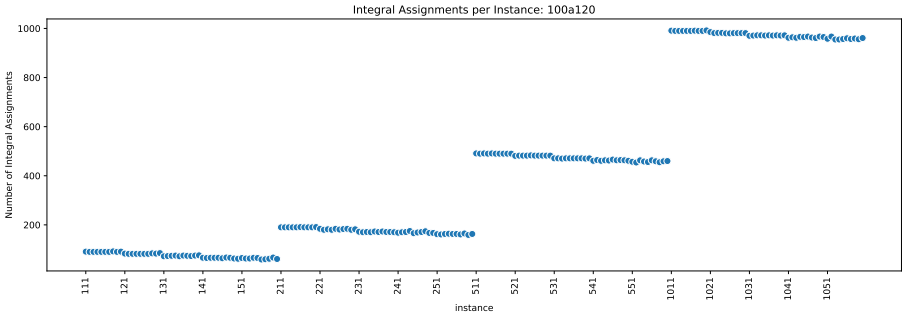

Figure 7: Integer Assignments

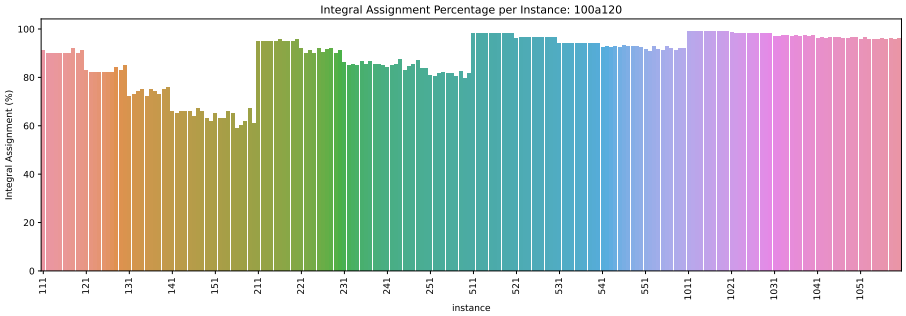

Figure 8: Integer Assignments %

2.3 100a200

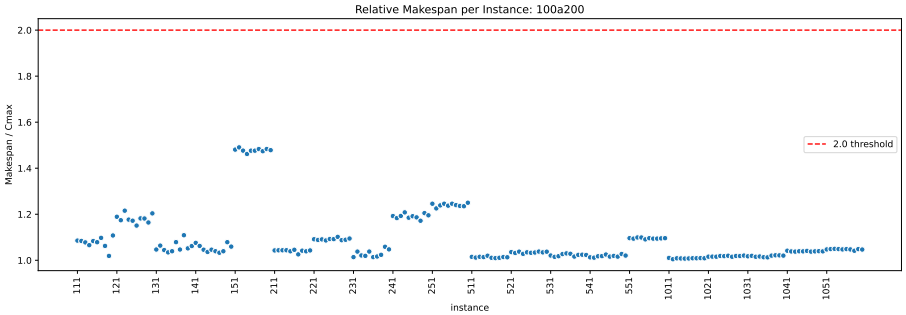

Figure 9: Approximation ratio

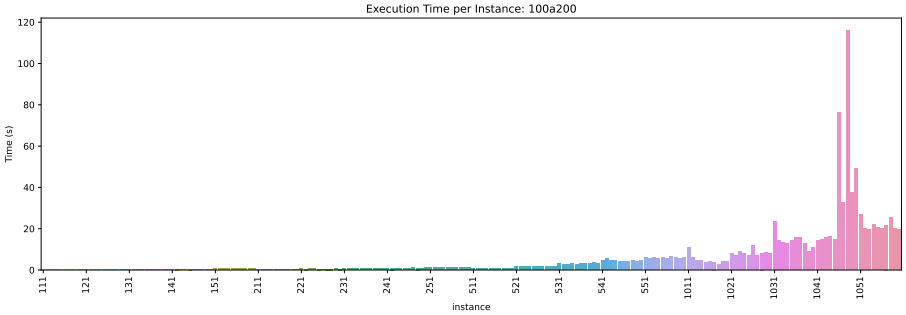

Figure 10: Execution Time

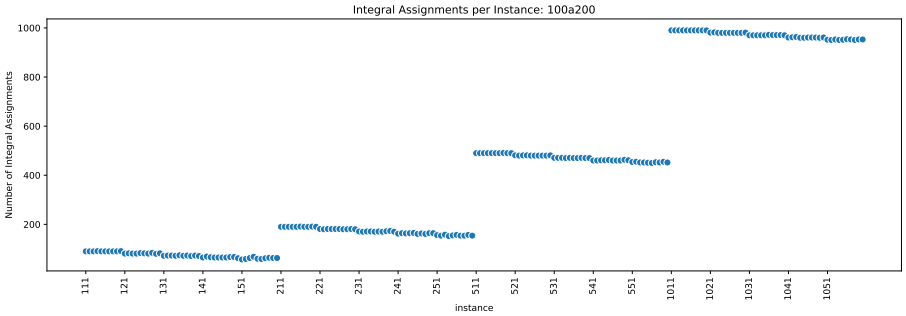

Figure 11: Integer Assignments

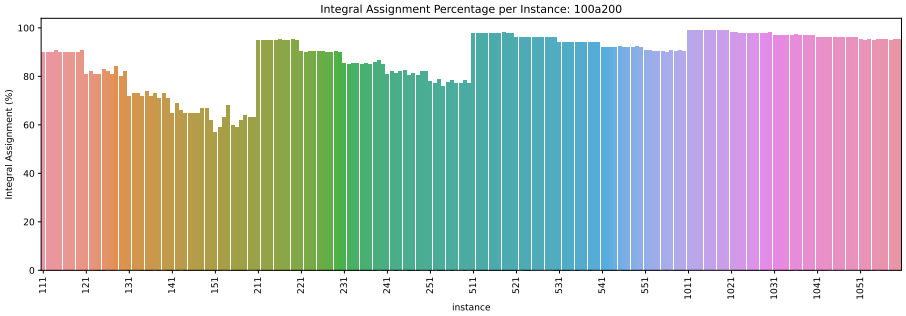

Figure 12: Integer Assignments %

2.4 1000a1100

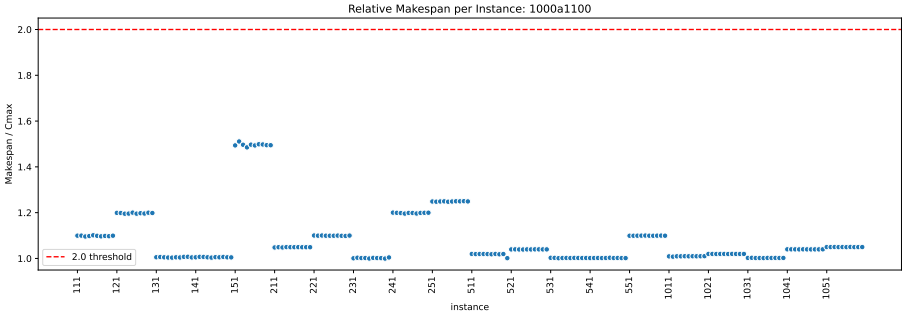

Figure 13: Approximation ratio

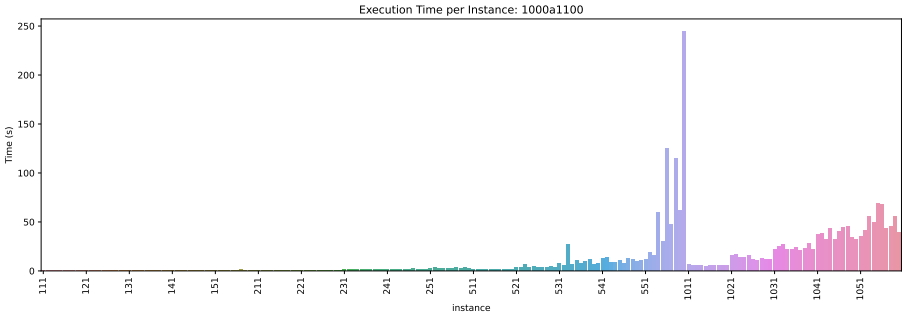

Figure 14: Execution Time

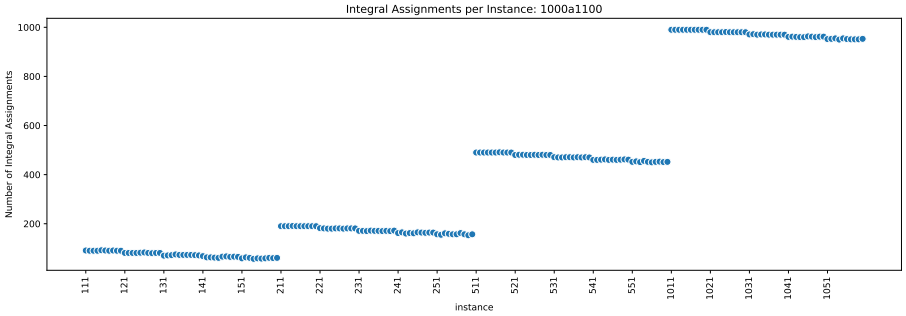

Figure 15: Integer Assignments

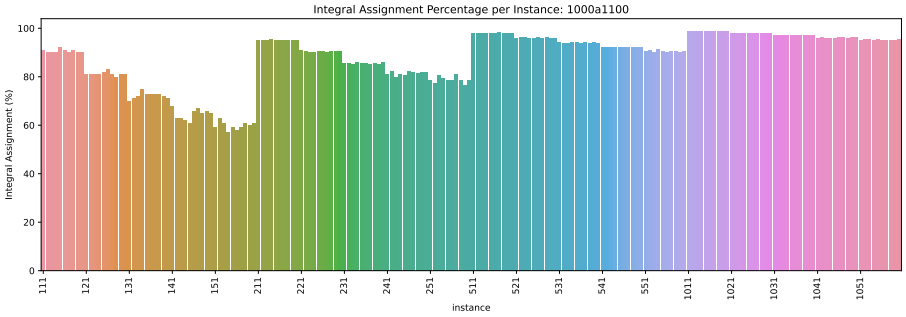

Figure 16: Integer Assignments %

2.5 de10a100

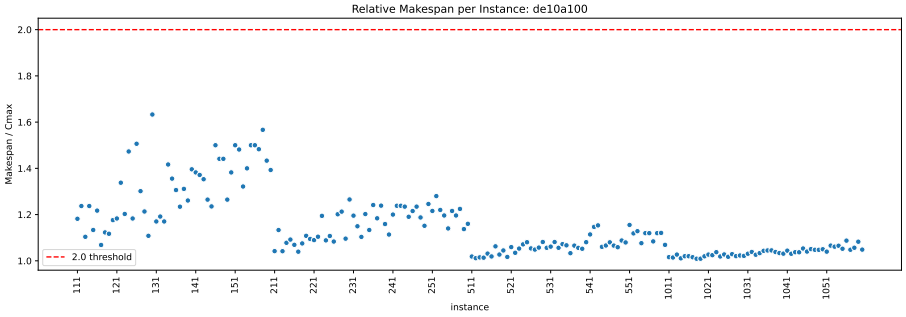

Figure 17: Approximation ratio

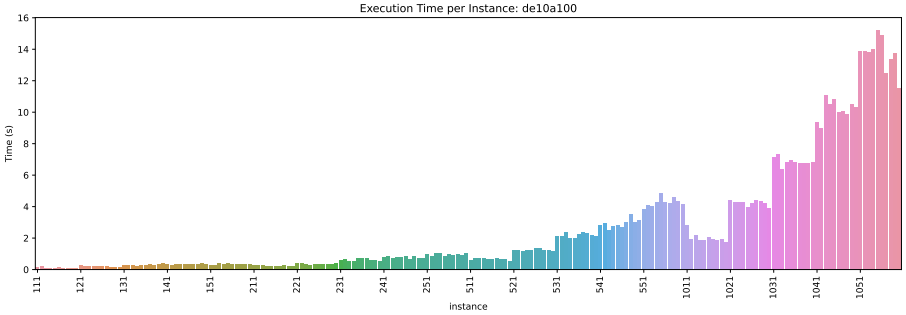

Figure 18: Execution Time

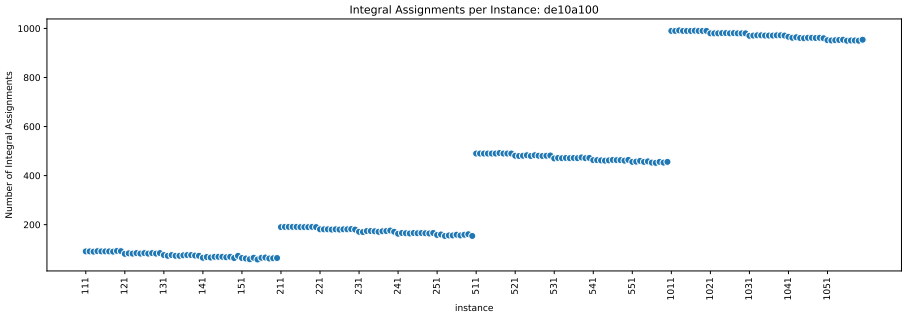

Figure 19: Integer Assignments

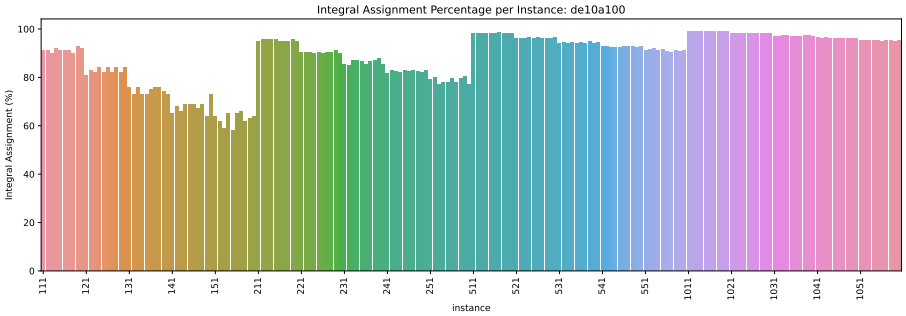

Figure 20: Integer Assignments %

2.6 JobsCorre

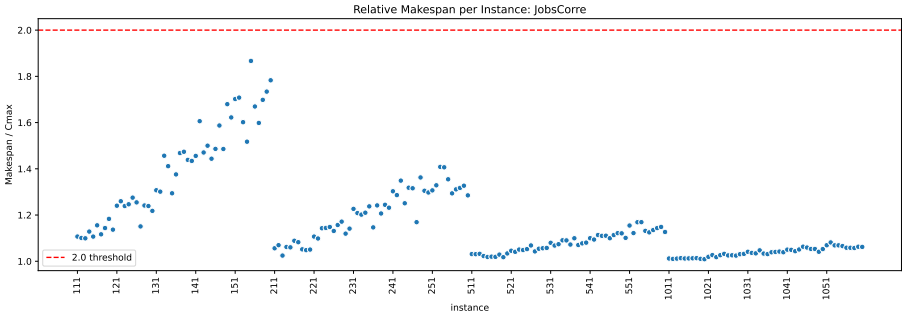

Figure 21: Approximation ratio

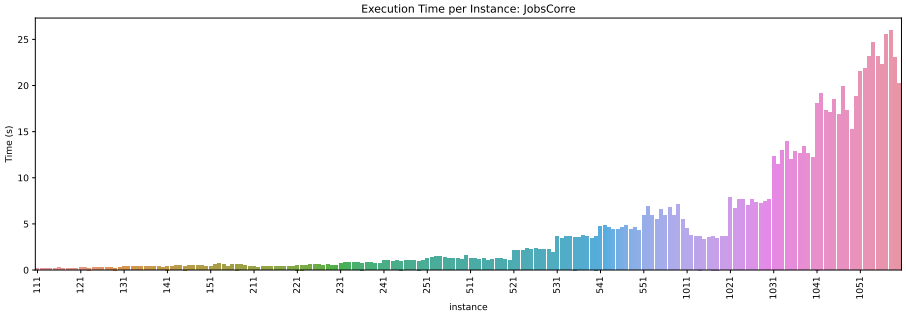

Figure 22: Execution Time

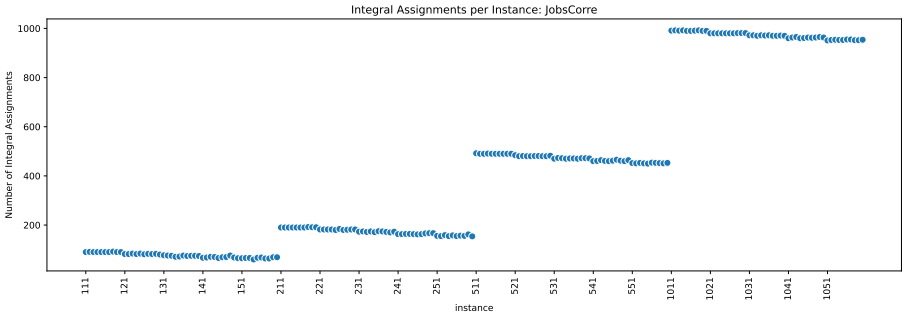

Figure 23: Integer Assignments

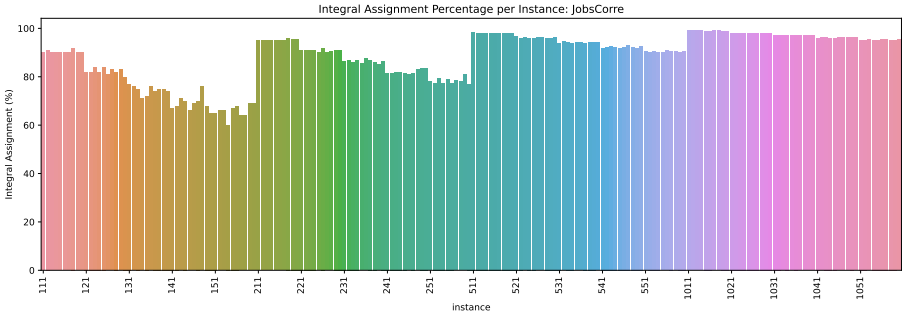

Figure 24: Integer Assignments %

2.7 MaqCorre

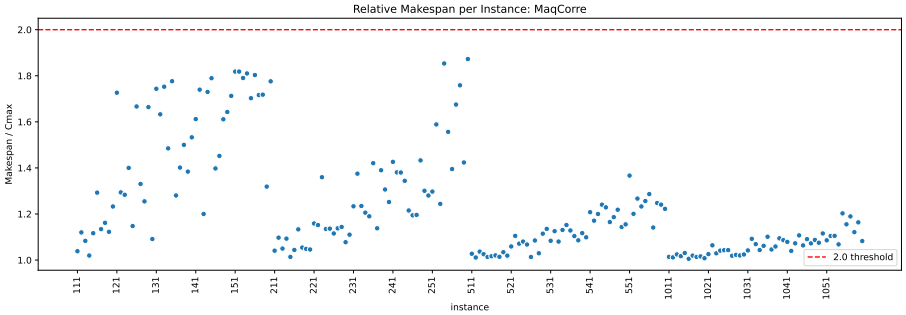

Figure 25: Approximation ratio

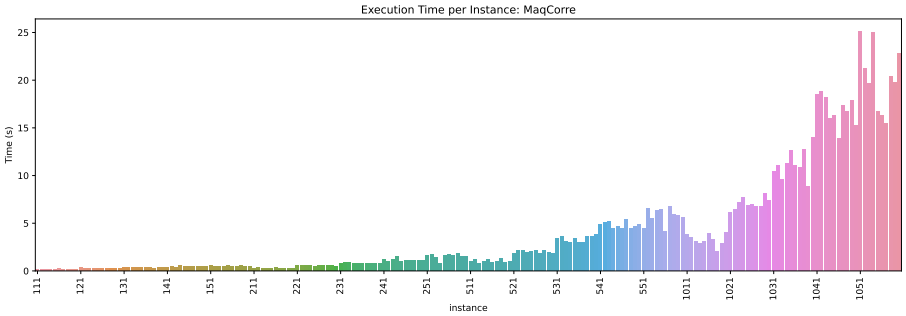

Figure 26: Execution Time

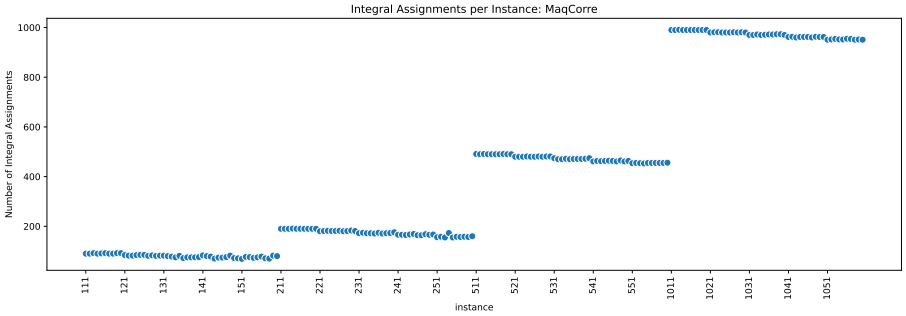

Figure 27: Integer Assignments

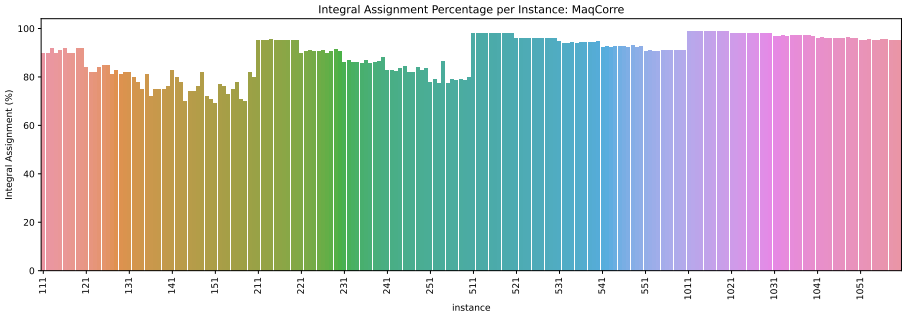

Figure 28: Integer Assignments %

3 Tables

Table 1: Average metrics across datasets

Dataset	Makespan / Cmax	Time (s)	Integral %
MaqCorre	1.236000	4.092000	89.688000
JobsCorre	1.181000	4.413000	88.593000
1a100	1.353000	1.126000	92.211000
100a120	1.085000	5.089000	88.622000
100a200	1.088000	5.469000	87.898000
de10a100	1.146000	2.630000	88.509000
1000a1100	1.084000	12.059000	87.790000

Table 2: Minimum metrics across datasets

Dataset	Makespan / Cmax	Time (s)	Integral %
MaqCorre	1.006000	0.128000	69.000000
JobsCorre	1.009000	0.126000	60.000000
1a100	1.015000	0.063000	78.000000
100a120	1.002000	0.146000	59.000000
100a200	1.005000	0.097000	57.000000
de10a100	1.009000	0.090000	58.000000
1000a1100	0.999000	0.169000	57.000000

Table 3: Maximum metrics across datasets

Dataset	Makespan / Cmax	Time (s)	Integral %
MaqCorre	1.873000	25.164000	99.100000
JobsCorre	1.867000	26.014000	99.200000
1a100	1.923000	6.145000	99.200000
100a120	1.542000	29.230000	99.200000
100a200	1.491000	116.195000	99.000000
de10a100	1.633000	15.250000	99.200000
1000a1100	1.511000	245.010000	99.000000

Table 4: Approximation ratios across datasets

Dataset	Min	Avg	Max
MaqCorre	1.006000	1.236000	1.873000
JobsCorre	1.009000	1.181000	1.867000
1a100	1.015000	1.353000	1.923000
100a120	1.002000	1.085000	1.542000
100a200	1.005000	1.088000	1.491000
de10a100	1.009000	1.146000	1.633000
1000a1100	0.999000	1.084000	1.511000

4 Conclusions

The algorithm's approximation coefficient in an average case is closer to 1 than to the worst case scenario of 2. What we may see is that in smaller problem instances, the algorithm gives on average worse approximations, and also here it reaches its overall worst case of 1.92.

Almost all problem instances were solved in less than 4 minutes, but on average they were computed in just a couple of seconds. On average, approximately 90% of the tasks were assigned according to the solver, while the rest were set using the perfect matching technique.

Algorithm works quickly and gives good approximation ratios.