UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA

CARLOS WESTERMANN

Winning a Tournament

Porto Alegre
May 2025

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitora: Profª. Marcia Barbosa
Vice-Reitor: Prof. Pedro Costa
Pró-Reitora de Pós-Graduação: Profª. Claudia Wasserman
Diretor do Instituto de Informática: Prof. Luciano Paschoal Gaspary
:
Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

1 Introduction

The following works presents an implementation and performance analysis of a reduction from the problem "Can team 1 win the tournament" to a maximum flow problem. The objective is to evaluate the execution time of the algorithm and the impact of biases for team 1 in the tournament result.

2 Testing

2.1 Testing Environment

All test cases were run on a machine with the following specifications:

- Chip: Apple M3 Pro
- Total Number of Cores: 12 (6 performance and 6 efficiency)
- Memory: 18 GB

2.2 Relevant implementation aspects

For the implementation of the reduction, I have used the same previous implementation of the Ford-Fulkerson (fattest-path version) algorithm. The reduction follows the demonstration present in the reference text for the class.

2.3 Testing Methodology

1. **Time Complexity Analysis:** This phase examined how the execution time of the algorithm scales with the number of teams.
   Multiple test instances with team counts ranging from 16 to 512 (in powers of 2) were generated while keeping constant the number of rounds (100), the fraction of games completed ($\alpha = 0.5$) and the bias parameter ($\beta = 0.0$).
2. **Winning Strategy Probability Analysis:** This phase investigated how tournament parameters affect the probability that a team has a winning strategy remaining.
   - **Bias Sensitivity:** With fixed parameters (20 teams, 38 rounds, $\alpha = 0.5$), the

bias for team 1 winning (β) varied from -1.0 to 1.0 with a 0.1 step. For each value, 20 random instances were generated to compute the probability of team 1 having a winning strategy.

- **Game Completion Effect:** With fixed parameters (20 teams, 38 rounds, $\beta = 0.0$), the fraction of completed games (α) varied from 0.0 to 1.0 in a 0.05 step. For each value, 20 random instances were generated to compute the probability of team 1 having a winning strategy.
- **Combined Parameter Effects:** With fixed parameters (20 teams, 38 rounds), α values from 0.0 to 1.0 and β values from -1.0 to 1.0 were tested in a grid with 5 repetitions per configuration. With the calculated results, a heatmap of the winning probabilities can be created.

3 Experimentation results and analysis

3.1 Time for execution x Number of teams

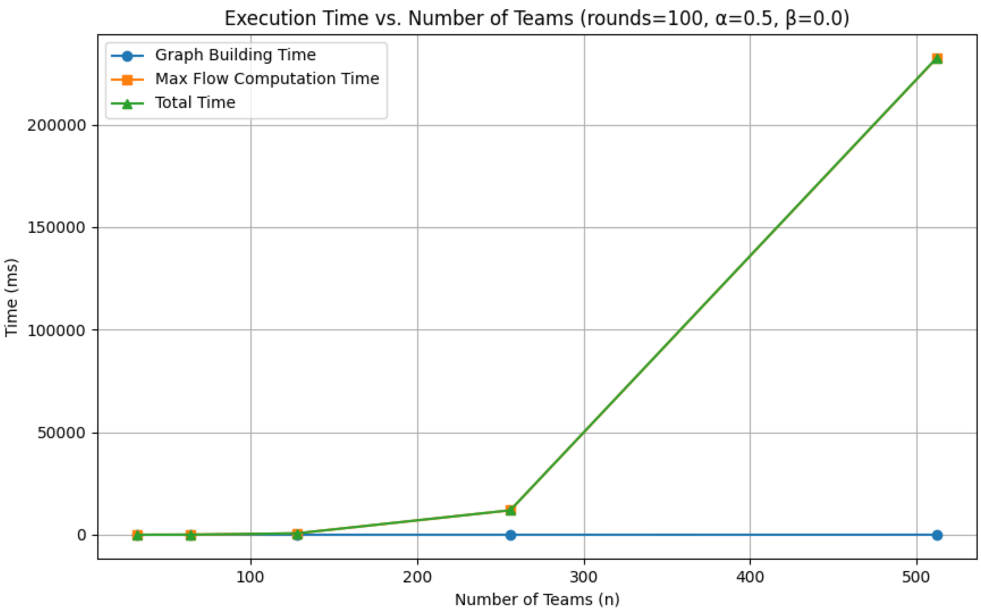

Figure 1 – Execution time of algorithm (ms) x Number of teams (with fixed number of rounds = 100, alpha = 0.5 and beta = 0.0).

In figure 1 we can see that the execution time of the graph building part of the algorithm stays almost constant, meaning that as the problem scales, the dominant bottleneck is the computation of the maximum flow. As for the expected complexity, the execution time behaves accordingly to what is expected, in a pattern slightly below the squared value of n. This was to be expected as the algorithm was already evaluated in previous works.

3.2 Probability of existing a winning strategy for team 1 x Fraction of completed games

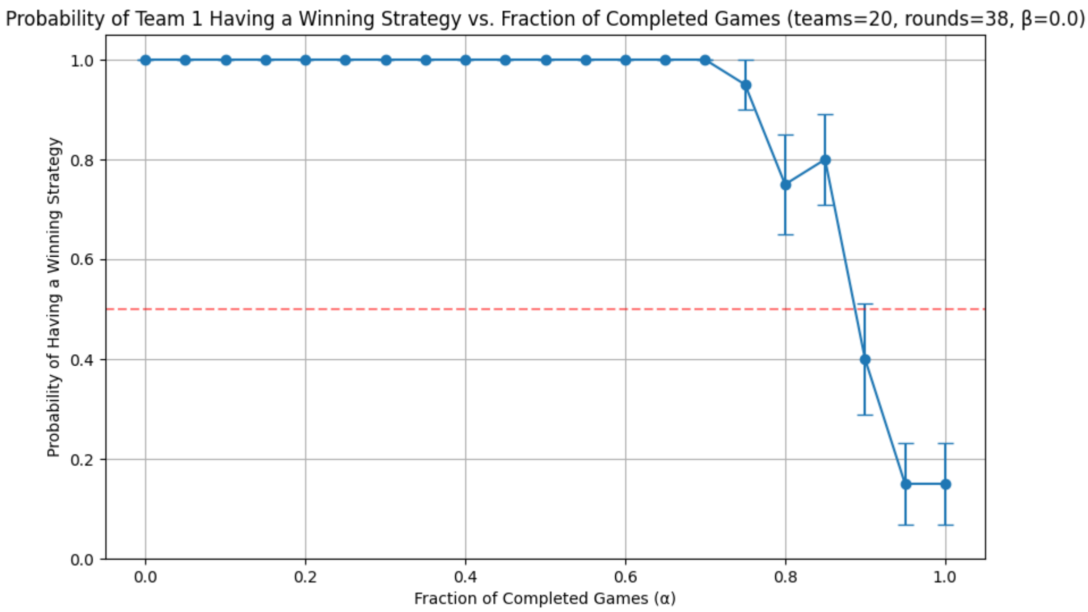

Figure 2 – robability of existing a winning strategy for team 1 x Fraction of completed games (with fixed teams = 20, rounds = 38 and beta = 0.0).

In figure 2 we can observe that for almost all scenarios where games were still available to be played and no bias towards team 1 was set, team 1 still had a winning strategy. This remains true for up to 3 quarters of the games already played, where team 1 started to not always have a winning strategy all the way to when all games were complete where it is never possible to win because all teams tie.

3.3 Probability of existing a winning strategy for team 1 x Bias for team 1 winning

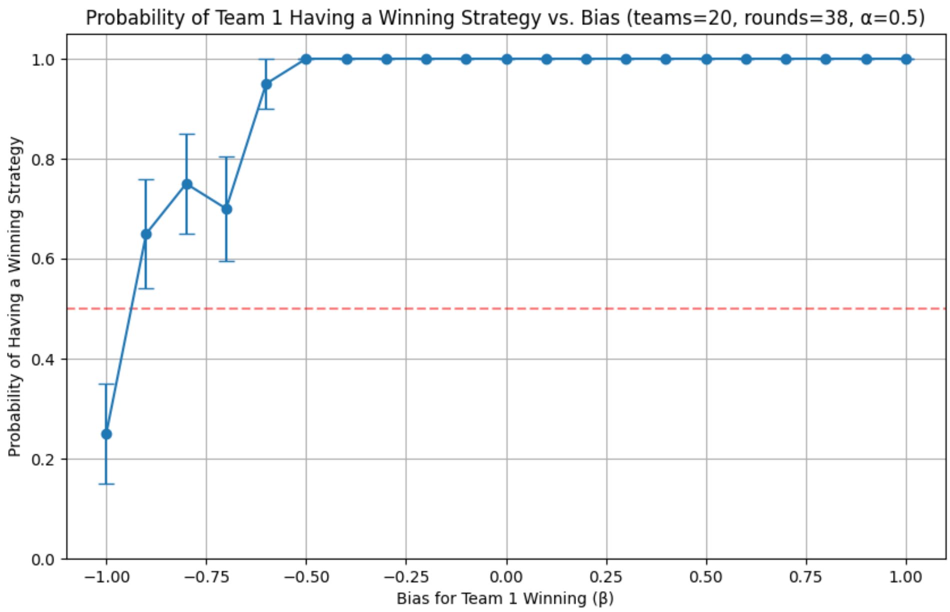

Figure 3 – Probability of existing a winning strategy for team 1 x Bias for team 1 winning games (with fixed teams = 20, rounds = 38 and alpha = 0.5).

In figure 3, we can observe that when half of the games were played, when the bias for team having already won some games does not play that big of a role for team 1 to still have a winning strategy. We see that after a bias of -0.5, team 1 can always find a winning strategy. This may be happening because there are still many games to be played and a winning strategy in this case would have to be winnning almost all following games, which although may be possible is unlikely in a real scenario.

3.4 Probability of existing a winning strategy for team 1 x Bias for team 1 winning and fraction of completed games

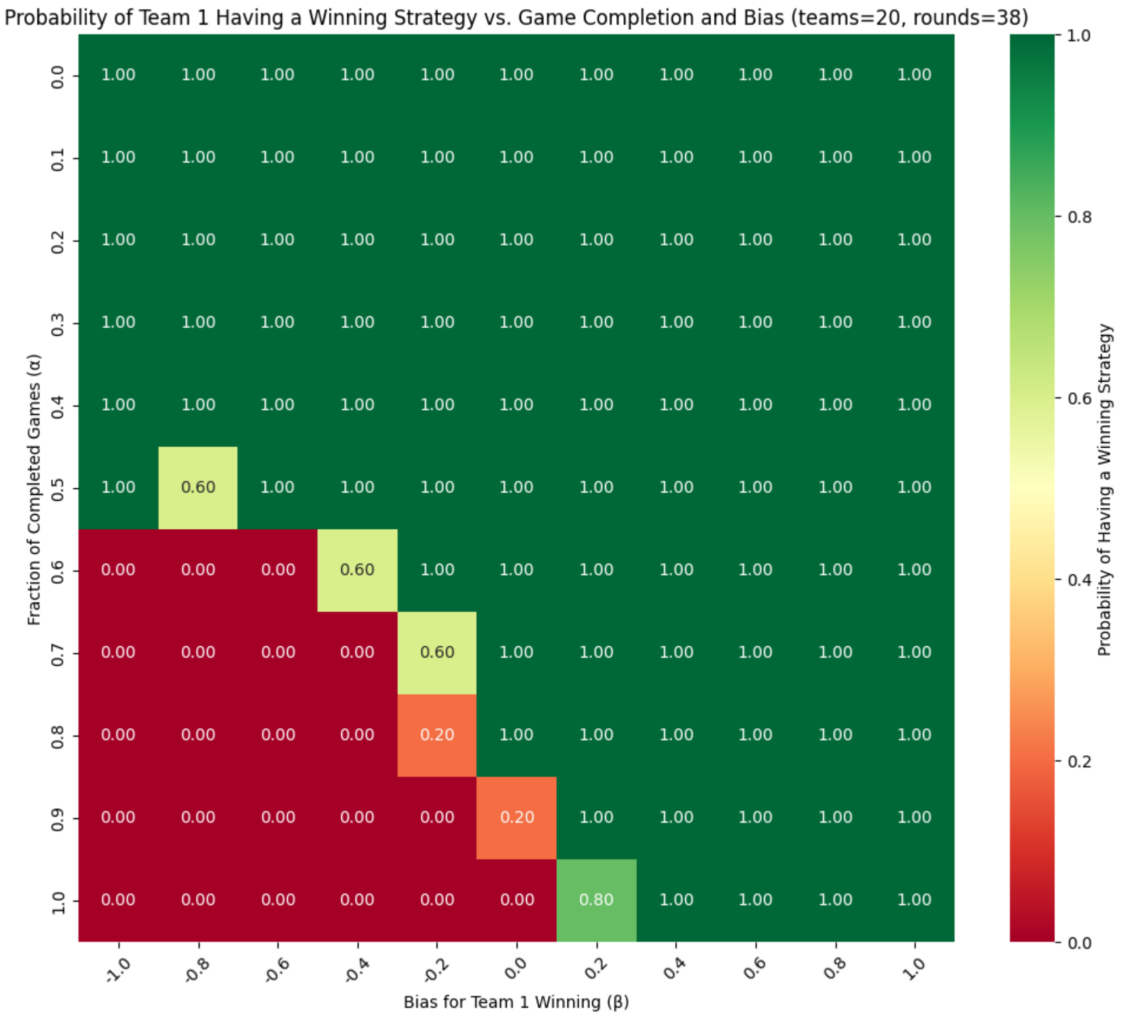

Figure 4 – Heatmap of probabilites of existing a winning strategy for team 1 for alpha x beta (with fixed teams = 20 and rounds = 38).

In figure 4 we observe a heatmap that combines the behavior for both previous figures, showing that only in scenarios in the bottom left corner, where the bias for team 1 to win is low and most games were already played is when there exists no chance for team 1 to still win. Other than that, there is always a winning combination for team 1 to win.

4 Conclusion

The experimentation and analysis presented in this work provide an evaluation of the reduction of the "Can team 1 win the tournament" problem to a maximum flow problem. The evaluation shows results that are consistent with the expectation, where the construction of the reduction shows minor impact on the overall execution time, and the vast majority of the complexity stems from the flow algorithm. As for the Probability of having a winning strategy for team 1, we observed that for most cases where at least half of the games were still waiting to be played, team 1 stood a chance of winning the tournament.