Motion-Planning Lab 2

Nir Manor 305229627, Ortal Cohen 308524875

17/03/2024

Part 2 – RRT* Planner

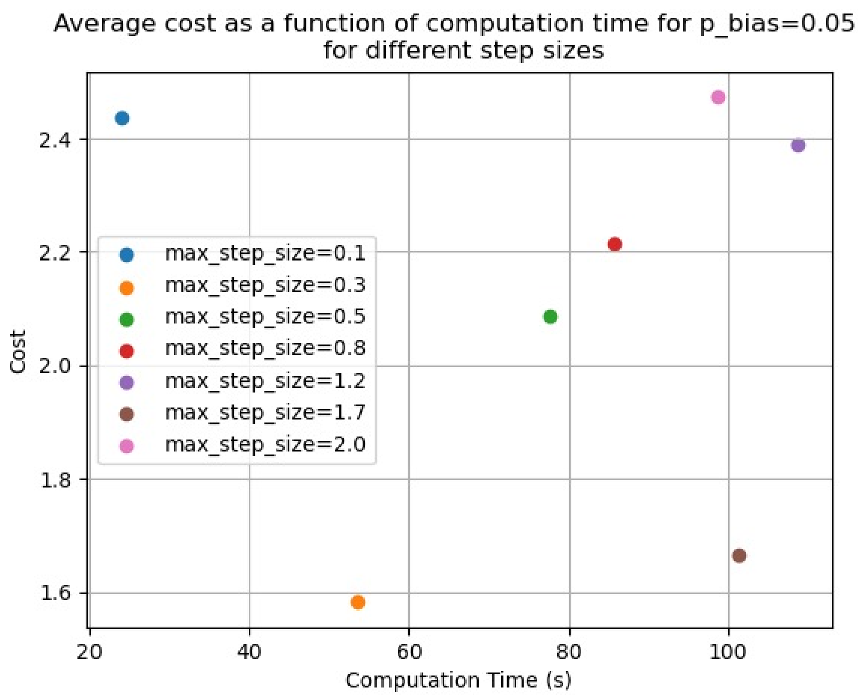

Figure 1: Path cost comparison for p-bias = 0.05

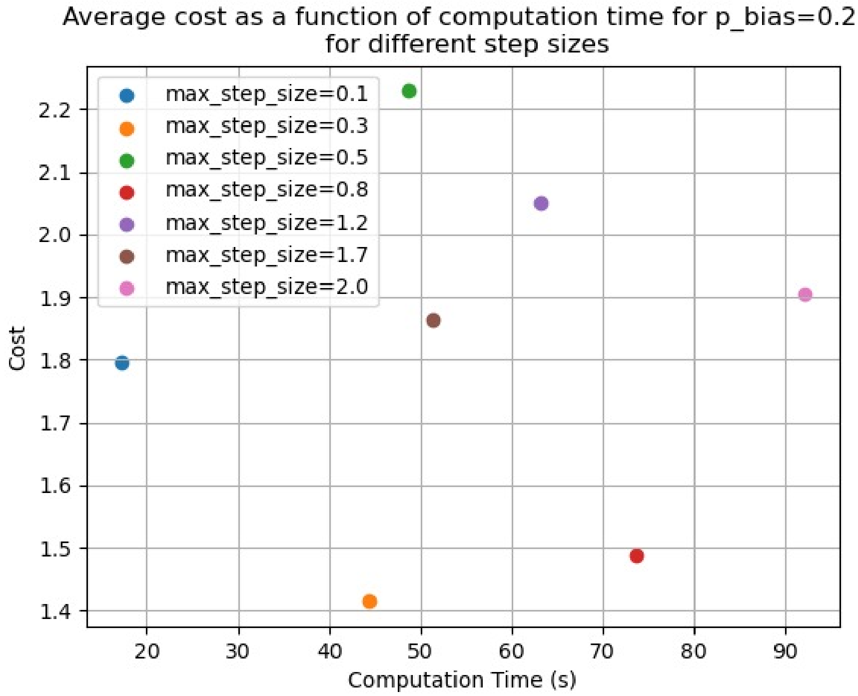

Figure 2: Path cost comparison for p-bias = 0.2

P-bias

P-bias in RRT* dictates the likelihood of selecting a random configuration from the goal region during tree expansion. Before testing, we anticipated that:

- Higher values will reduce planning time on the cost of producing suboptimal solutions because it will guide the exploration towards the goal.

- Lower values will increase the planning time possibly leading to discover better paths because it will increase the chance of discovering unique paths.

In our experiments, we found that setting p-bias to 0.05 (with a maximum step size of 0.8) resulted in the lowest path cost of 1.174, while setting p-bias to 0.2 (with a maximum step size of 1.2), the lowest cost was 1.211. However, we noticed that on average, paths with p-bias = 0.05 had slightly higher costs compared to p-bias = 0.2. This indicates that while higher p-bias values guide exploration toward promising areas, they may lead to higher average costs. Conversely, lower p-bias values allow for more thorough exploration, potentially revealing paths with lower costs. Although this exploration might take longer, it can be beneficial as it discovers better paths.

In conclusion, the effect of p-bias on the performance:

- Cost: Higher p-bias values may reduce the cost of planning, while lower p-bias values can lead to paths with lower costs by allowing for more thorough exploration.

- Time: Higher p-bias values may reduce planning time, while lower p-bias values may result in longer planning times due to more thorough exploration.

Max Step Size

The maximum step size parameter in RRT* determines how far the algorithm moves in each step as it explores the space. Larger steps mean the algorithm covers more ground quickly, but it might miss important details. On the other hand, smaller steps take more time but allow the algorithm to thoroughly examine the space, potentially finding better paths.

P-bias 0.05:

- Cost: For p-bias = 0.05, different step sizes resulted in different costs. Paths with step sizes around 0.3 tended to be the least expensive, while those with step sizes around 0.1 were the most costly. Interestingly, a step size of 1.7 had a cost similar to 0.3, but it took twice as long to find.
- Time: Increasing the step size led to longer times to find a solution.

P-bias 0.2:

- Cost: Similarly, for p-bias = 0.2, paths with a step size of 0.3 were usually the least expensive. However, we observed some inconsistency in behavior across different step sizes.
- Time: While the step size of 2.0 took the longest time, the relationship between step size and solution time did not follow a consistent pattern for other step sizes. This suggests that the impact of step size on solution time may be influenced by other factors as well.

In conclusion, the impact of maximum step size on performance:

- Cost: In general, smaller step sizes tended to lead to less expensive paths, but sometimes larger steps were just as good, even if they took longer.
- Time: Surprisingly, larger steps usually meant longer solution times. This might be because larger steps sometimes wasted time exploring unimportant areas. Further study is needed to determine the best step size for different tasks.

Comparison with OMPL

In our experiments, we observed a substantial discrepancy in planning time between our implementation and the Open Motion Planning Library (OMPL). While our implementation required around 43-44 seconds to find a solution, OMPL completed the same task in approximately 3 seconds. Although the quality (cost) of the solutions was not directly compared, it's reasonable to assume that OMPL's solution is superior due to its efficiency. The significant difference in planning time could be attributed to several factors, including the optimized collision detection mechanisms and algorithmic efficiencies employed by OMPL. Additionally, OMPL's implementation in C++ allows for low-level optimizations that may enhance performance.

Conclusion

Choosing appropriate parameters, such as p-bias and max step size, is crucial in optimizing the performance of RRT* motion planning. These parameters dictate the balance between exploration and exploitation, impacting both planning time and solution quality. Further optimization efforts, inspired by efficient libraries like OMPL, can enhance the performance of our RRT* implementation.