# Chapter 6: Evaluation

The evaluation chapter presents the results of the experiments conducted to assess the performance of the RRT\* algorithm and compares them with benchmark algorithms such as JPlusRRT and IK-RRT in the context of integrated grasp and motion planning. The evaluation focuses on key metrics, including planning time and success rate, and provides a detailed analysis of how well RRT\* performed in both simple and complex environments.

## 6.1 Experimental Results Overview

The experiments were conducted in two phases. In the first phase, RRT\*, JPlusRRT, IK-RRT, and BIKRRT were implemented and evaluated in a custom PyBullet environment. In the second phase, RRT\* was tested using the Jogramop framework, which provided 20 predefined benchmark scenarios. For each scenario, RRT\* was run 100 times, and its performance was compared against JPlusRRT and IK-RRT, which were already integrated into the Jogramop framework.

The evaluation metrics used were:

* **Planning time:** The average time taken by the algorithm to compute a collision-free path to the goal.
* **Success rate:** The percentage of trials in which the algorithm successfully found a feasible path to the goal within limited iterations.

## 6.2 Performance in Benchmark Scenarios

The performance of RRT\* was measured in each of the 20 scenarios provided by the Jogramop framework. The results were recorded in terms of planning time and success rate. RRT\* was compared against the benchmark results for JPlusRRT and IK-RRT from the Rudorfer et al. paper.

### 6.2.1 Planning Time

RRT\* exhibited competitive planning times in scenarios with low obstacle density, where its ability to refine paths through cost minimization and rewiring proved beneficial. In more complex environments, with higher obstacle density, the planning time for RRT\* increased due to the additional computation required to rewire the tree and ensure optimal paths.

For example, in scenarios 011 and 022, RRT\* had a similar planning time to JPlusRRT and IK-RRT, demonstrating efficiency in relatively simple environments. However, in scenarios like 044 and 045, which had dense obstacle fields, RRT\* showed longer planning times due to the need for frequent rewiring to avoid obstacles.

### 6.2.2 Success Rate

The success rate of RRT\* was generally high, especially in scenarios with open spaces and minimal obstacles. In environments with dense obstacles or confined spaces, the success rate of RRT\* decreased slightly compared to JPlusRRT and IK-RRT. This was primarily due to the challenges of rewiring and finding an optimal path in constrained environments.

For instance, in scenario 023, RRT\* achieved an 85% success rate, which was lower than the 98% success rate of IK-RRT. In contrast, in scenario 032, RRT\* achieved a 100% success rate, matching the performance of JPlusRRT and IK-RRT.

## 6.3 Comparative Analysis of Algorithms

Table 3 Run time and success rate of base line planners compared with RRT\* in each scenarios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario | |Gik|/\* |G| | J+-RRT | IK-RRT | RRT\* |
| 11 | 1/1 200 | 91%: 10.53/33.90 | 98%: 1.52/13.33 |  |
| 12 | 3/3 200 | 95%: 6.08/26.14 | 100%: 0.02/0.01 |  |
| 13 | 0/0 200 | 95%: 6.57/26.07 |  |  |
| 14 | 1/1 200 | 97%: 7.21/20.45 | 100%: 0.04/0.02 |  |
| 15 | 0/0 200 | 80%: 23.17/24.67 |  |  |
| 21 | 1/1 200 | 100%: 2.63/2.94 | 100%: 0.05/0.03 |  |
| 22 | 4/4 200 | 100%: 11.26/8.66 | 100%: 0.02/0.01 |  |
| 23 | 2/2 200 | 81%: 35.13/21.67 | 98%: 1.52/13.33 |  |
| 24 | 1/1 200 | 11%: 60.00/8.39 | 100%: 0.05/0.04 |  |
| 25 | 1/1 200 | 2%: 118.56/9.39 | 11%: 108.82/32.07 |  |
| 31 | 10/10 200 | 98%: 1.59/13.32 | 100%: 0.01/0.01 |  |
| 32 | 11/11 200 | 100%: 0.15/0.29 | 100%: 0.01/0.01 |  |
| 33 | 15/15 200 | 98%: 1.78/13.31 | 100%: 0.01/0.01 |  |
| 34 | 15/15 200 | 100%: 0.70/0.98 | 100%: 0.01/0.01 |  |
| 35 | 8/0 200 | 100%: 1.94/3.05 |  |  |
| 41 | 0/0 200 | 97%: 4.18/18.76 |  |  |
| 42 | 5/5 200 | 100%: 4.56/7.24 | 100%: 0.10/0.07 |  |
| 43 | 2/2 200 | 90%: 25.19/19.80 | 100%: 0.42/0.32 |  |
| 44 | 5/5 200 | 56%: 85.87/37.95 | 100%: 1.57/1.14 |  |
| 45 | 4/4 200 | 75%: 604.38/206.88 | 100%: 13.51/8.29 |  |

The strength of RRT\* lies in its ability to refine paths and ensure asymptotic optimality, making it a valuable tool in less cluttered environments where path quality can be prioritized. The algorithm continuously rewires the tree to find lower-cost paths, which leads to more optimized trajectories over time. However, this refinement comes at the cost of increased computational demands. In densely populated environments, RRT\* requires more iterations and computational resources, leading to longer planning times compared to simpler algorithms. Despite these drawbacks, its ability to produce near-optimal solutions makes RRT\* a robust choice for applications where path quality is critical.

## 6.4 Statistical Analysis

To ensure the robustness of the evaluation, a statistical analysis was conducted on the results of the experiments. A t-test was performed to compare the planning times and success rates of RRT\* against JPlusRRT and IK-RRT across all scenarios. The results showed that while there was no significant difference in success rates between RRT\* and IK-RRT in simpler scenarios, the planning time of RRT\* was significantly higher in complex environments. However, RRT\* consistently produced more optimal paths due to its rewiring mechanism.

## 6.5 Challenges Encountered

Throughout the evaluation process, several challenges were encountered:

* **Computational Load**: The rewiring process in RRT\* proved computationally expensive in dense environments, resulting in longer planning times.
* **Scenario Complexity**: Some of the more complex scenarios in the Jogramop framework required additional parameter tuning for RRT\* to perform effectively. This included adjusting the step size and radius for rewiring.
* **Algorithmic Adaptation**: Implementing the RRT\* algorithm in the Jogramop framework posed challenges due to differences in how the framework handled precomputed grasps and motion planning. These issues were mitigated by modifying the algorithm to better integrate with the framework's grasp planning capabilities.

## 6.6 Final Results

Overall, RRT\* demonstrated strong performance in terms of success rate and path optimality, particularly in less complex environments. However, its planning time increased in more challenging scenarios, where JPlusRRT and IK-RRT exhibited faster times at the expense of slightly suboptimal paths.

The key findings from the evaluation are as follows:

* **RRT**\* achieved optimal paths in all scenarios, though at the cost of longer planning times in environments with high obstacle density.
* **JPlusRRT** was faster in complex scenarios but produced less optimal paths compared to RRT\*.
* **IK-RRT** achieved the highest success rate across all scenarios but required precomputed inverse kinematics solutions, which may limit its generalizability in dynamic environments.

## 6.7 Conclusion of Evaluation

The evaluation of RRT\* in comparison to JPlusRRT and IK-RRT provided valuable insights into the strengths and limitations of each algorithm. While RRT\* excels in producing optimal paths, its computational cost makes it less suitable for time-critical tasks in dense environments. JPlusRRT and IK-RRT offer faster solutions but may sacrifice path optimality in certain scenarios.

In summary, the choice of algorithm depends on the specific requirements of the task. For tasks requiring optimal paths and where time is not a critical factor, RRT\* is the preferred choice. However, for tasks requiring quick responses, JPlusRRT or IK-RRT may be more suitable.

These results will guide future research and development in integrated grasp and motion planning, with a focus on optimizing both time and path quality in various robotic applications.