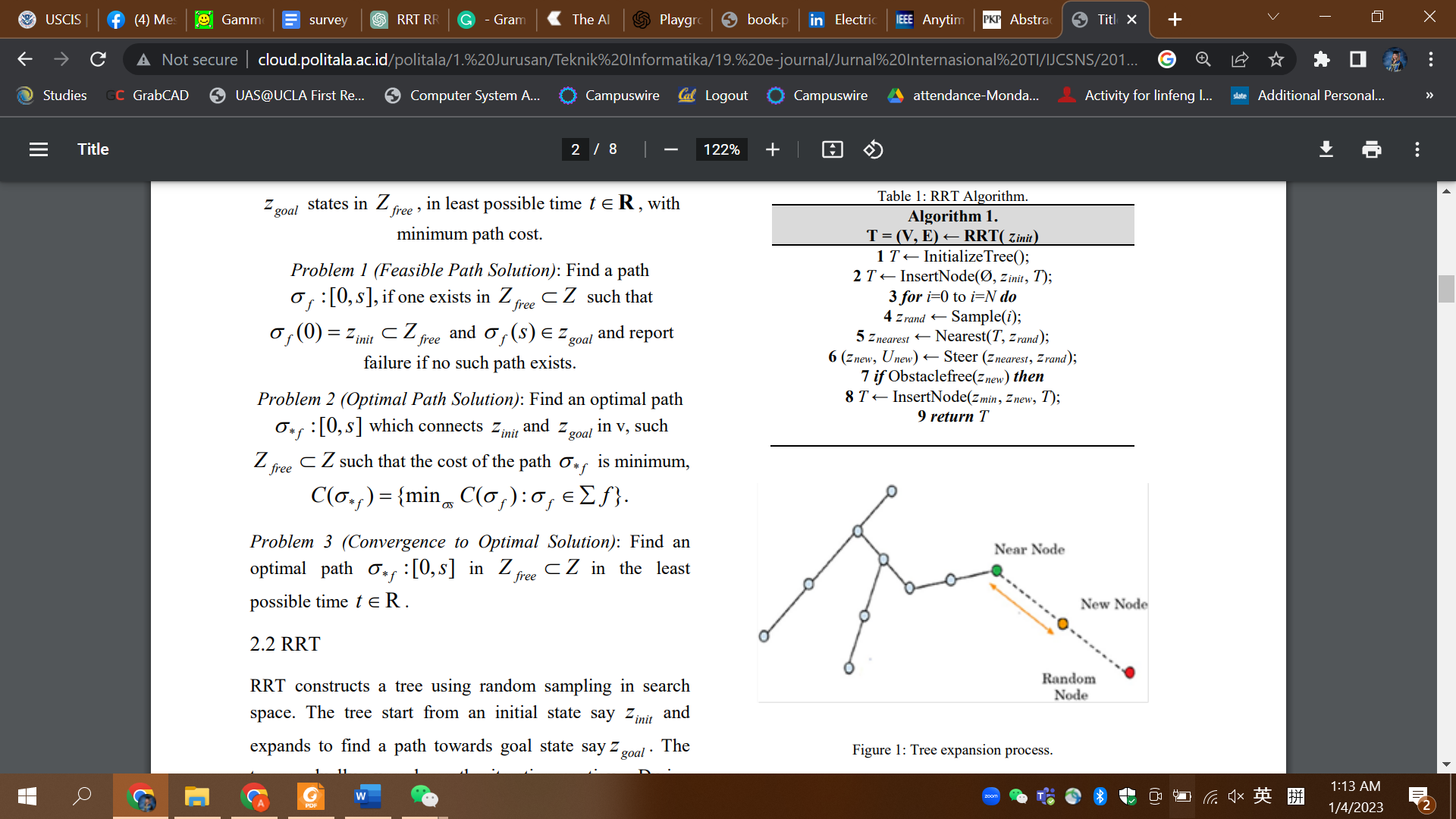
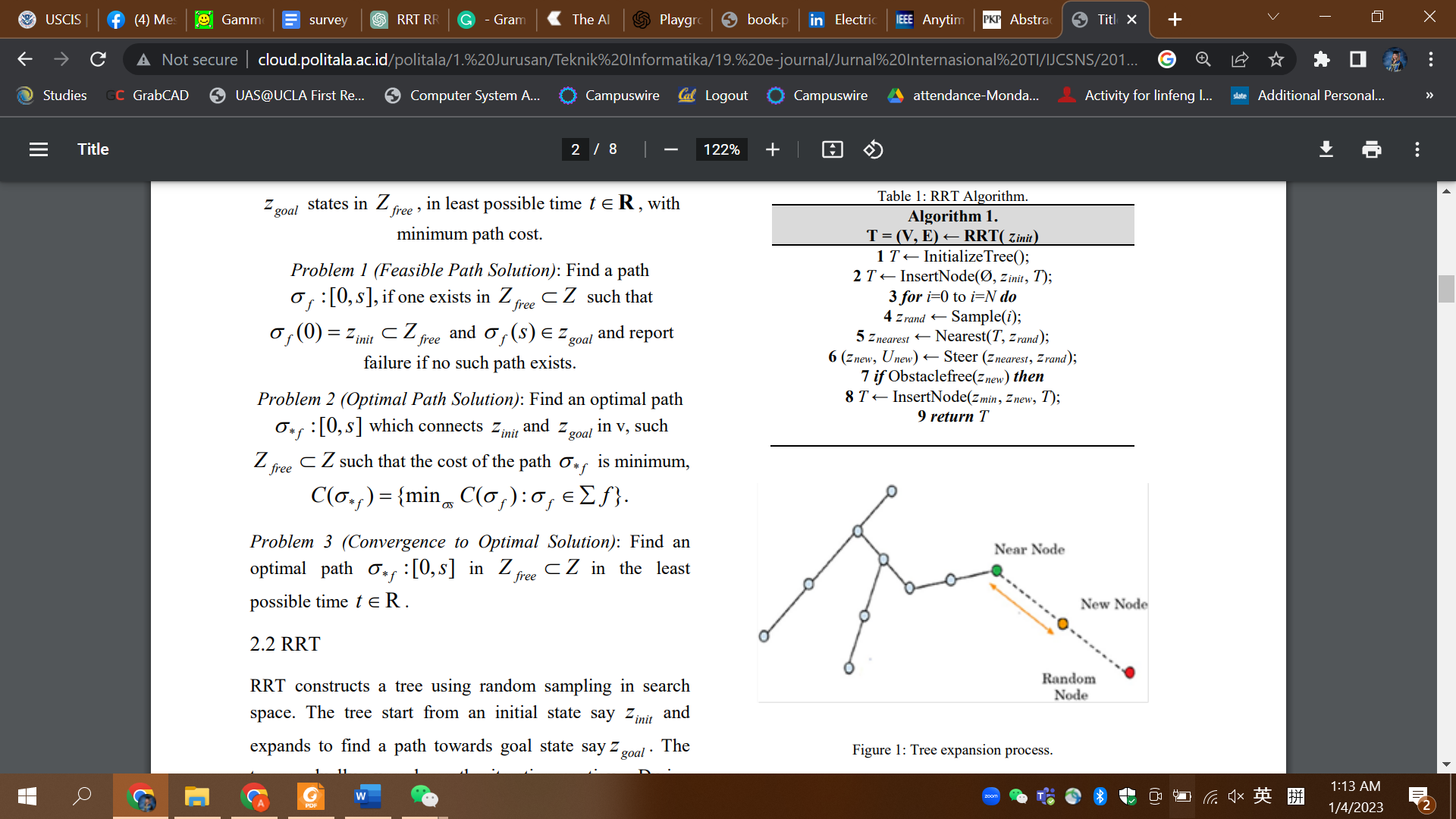
Extensions to the vanilla RRT algorithm

Linfeng Li, Shubham Wani, Yang Liu

The Rapidly-exploring Random Trees (RRT) algorithm is a powerful tool that can be utilized in robotics and computer graphics to find paths between two points. It has applications in robotics for path planning in known environments and motion planning.

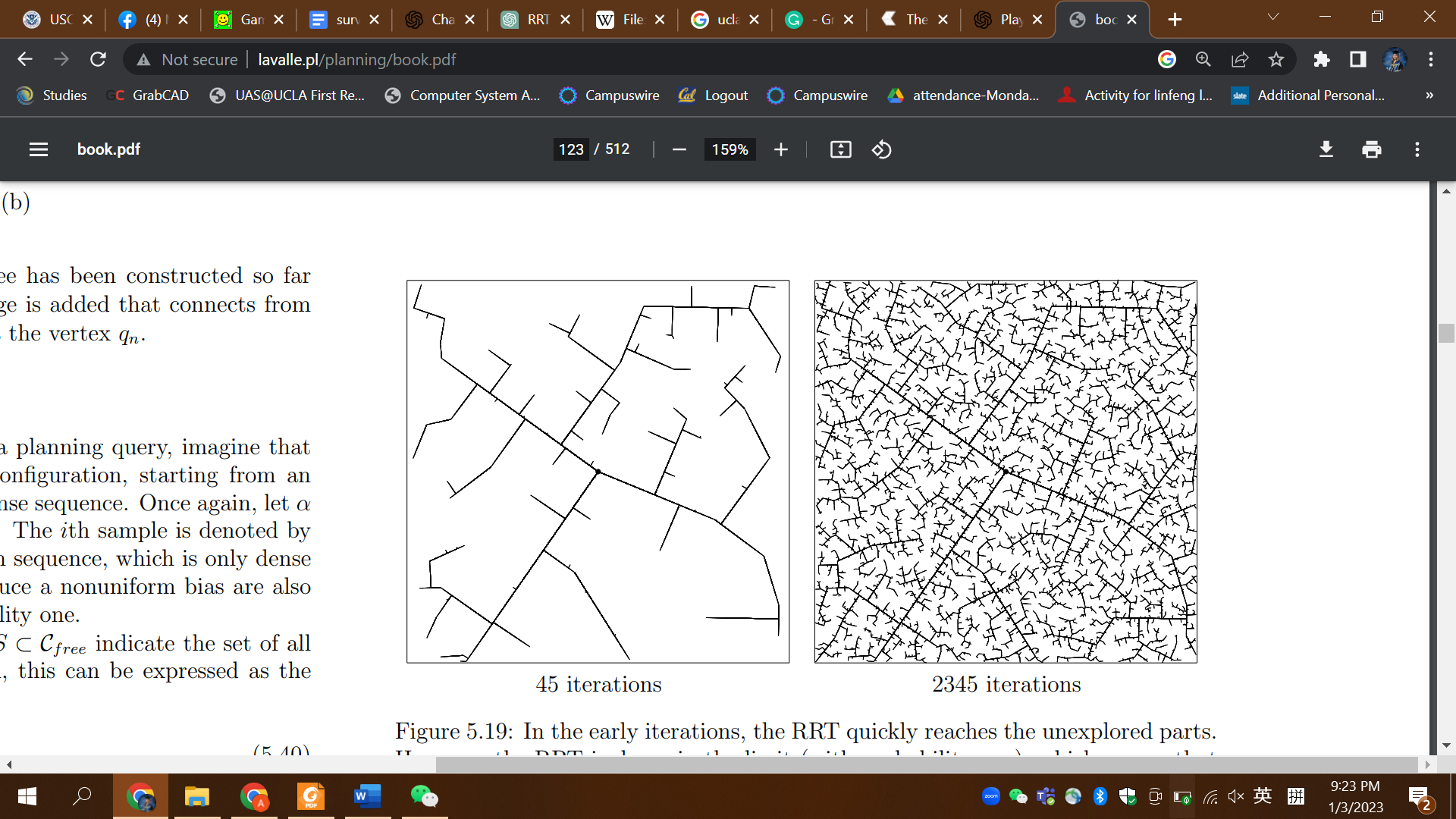
The RRT algorithm constructs a tree in the search space using random sampling. This tree begins at the starting point and grows towards the goal point, expanding as the algorithm runs through multiple iterations and adds nodes to the tree.

The algorithm selects a uniformly random point in the state space during each iteration. The nearest node is found in the tree to the random point by calculating the minimum euclidean distance to all nodes in the neighborhood. The tree is steered from the nearest node to the random node unless an obstacle is reached. If the random point is too far away, a new point, known as the "new node," is generated and added to the tree. This new node is then connected to the nearest node using straight lines or kinematically valid curves. These curves are then added as edges to the tree, causing the tree to grow. The algorithm continues execution until a path to the goal is discovered or a certain number of iterations have occurred. If the path is not found within the maximum iterations, either the path does not exist, or we need more time to explore the map. The process of adding new nodes to the tree is depicted in the following figure. [1]



*Fig. 1[1]*

The picture below shows how RRT explores a state space with a growing number of iterations:



*Fig. 2[2]*

The RRT algorithm is a good choice for finding paths in high-dimensional spaces or continuous spaces. Hence, it is a widely used method for tasks that involve planning paths. However, the basic RRT algorithm has some limitations. It does not necessarily find the shortest path between the starting point and the goal; it only guarantees to find a path if one exists. These limitations can be a problem if there is a need to find the optimal path for a particular application, like the shortest distance or the quickest time. We need to use a modified version of the algorithm in these cases.

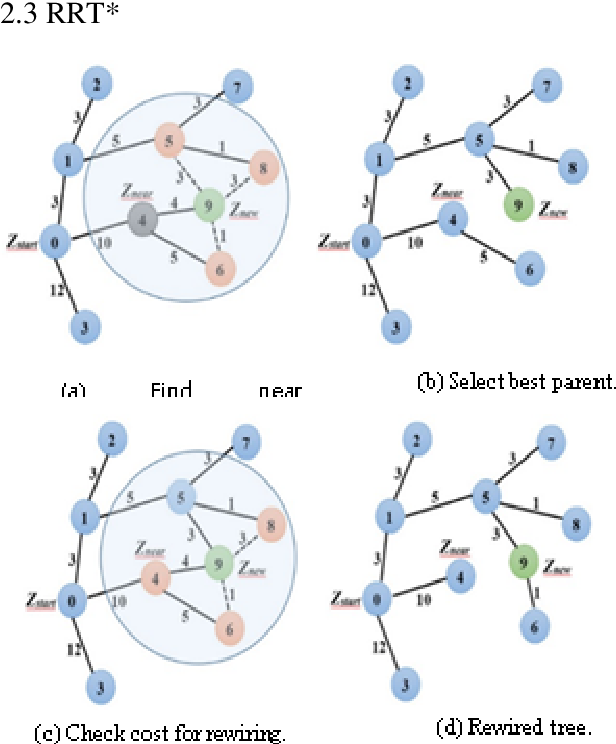
In this paper, we explored three versions of extension to basic RRT: RRT\*, informed RRT\*, and RRT\*-FN.

1. RRT\*

The basic RRT algorithm may take longer to converge in finding a path if the space is too large. Additionally, while it can find a path between a start and end point, it does not necessarily find the optimal path. To overcome these limitations, researchers introduced a modified version of the algorithm called RRT\* (RRT-Star).

RRT\* utilizes the technique of "guided sampling" to quickly find shorter paths. This method directs the search towards areas of the space that are more likely to contain shorter paths. As a result, RRT\* is more effective at finding the optimal solution as compared to the basic RRT algorithm. It can find shorter paths more quickly and is usually more efficient overall.

According to the paper “A Comparison of RRT, RRT\* and RRT\*-Smart Path Planning Algorithms,” RRT\* is a much more efficient way of sampling with a faster convergence rate.



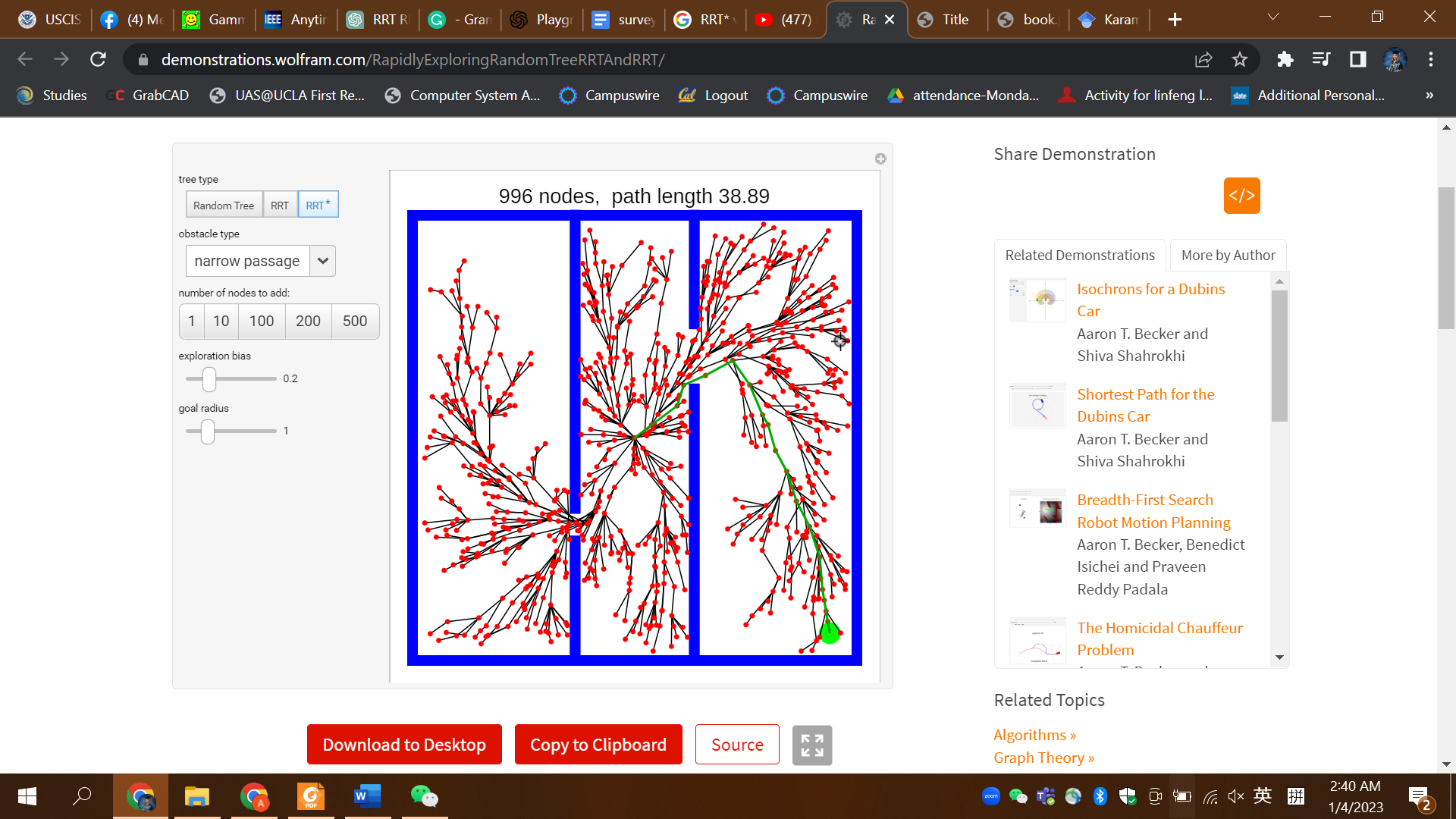
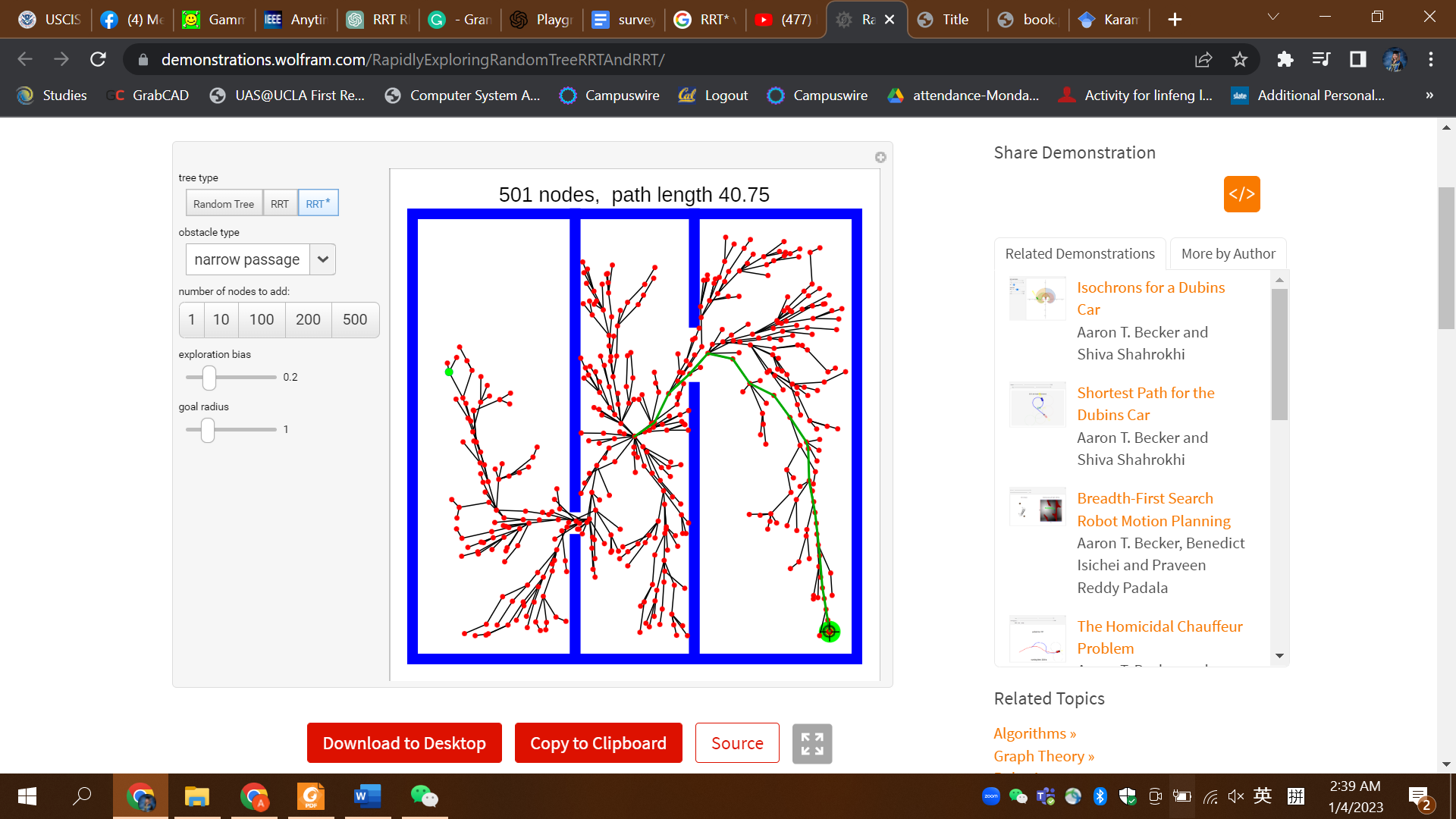
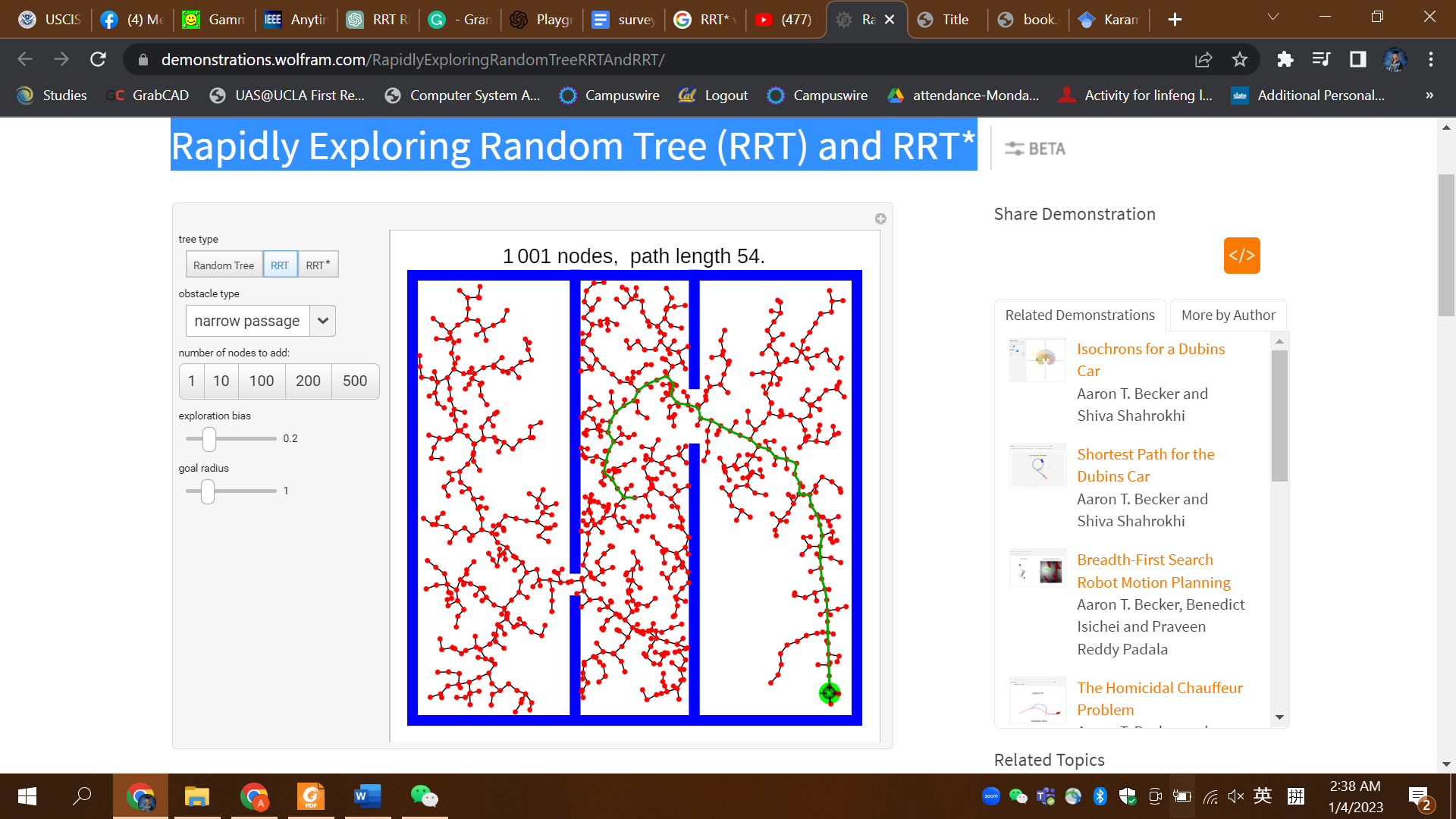
*Fig. 3[1]*

RRT\* (RRT-Star) is an extended version of the RRT algorithm with additional capabilities. As stated in the previous page, RRT generates nodes randomly in the state space and finds the node among the existing nodes closest to the random node, creating a connection between them. RRT\* also has two abilities: "near neighbor search" and "rewiring tree operations" [1].

Near-neighbor search involves drawing a circle with the new node at the center and a certain radius. The algorithm searches within this circle for other nodes. If it finds a node with a lower cost within the circle, it creates a connection between the new node and the node with the lower cost, resulting in a "rewiring" of the tree to reduce the overall cost according to the equation *cost\_child = cost\_parent + distance*.

RRT\* improves at finding paths with a lower cost as the algorithm continues running through more iterations. It improves gradually over time and eventually finds the optimal path. In contrast, the basic RRT algorithm does not improve as much since it cannot optimize the edges of the tree.

We used Wolfram Demonstration Project “Rapidly Exploring Random Tree (RRT) and RRT\*” to do a simulation. It illustrated the differences between RRT and RRT\*. The results have been showcased below:



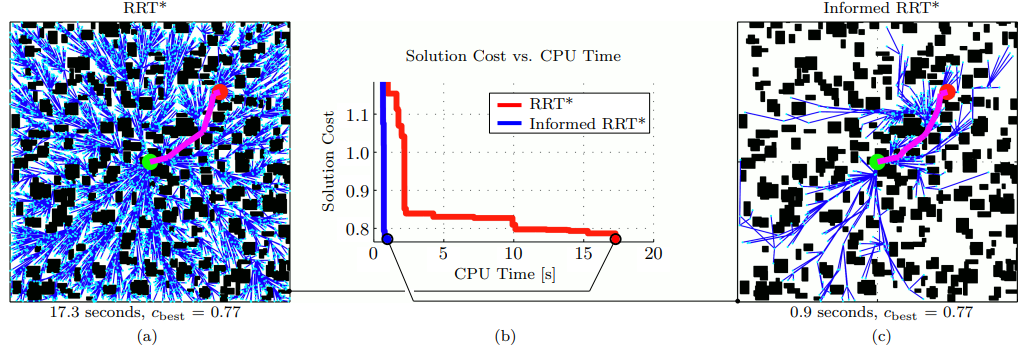
Result using RRT Result using RRT\* Result using RRT\*

*Fig. 4[3]*

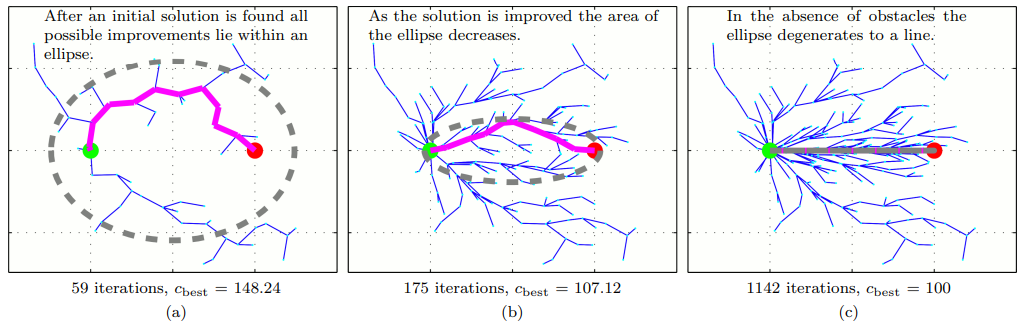
The results of using the RRT algorithm show that 1000 nodes were generated to find a path from the starting point to the goal, with a cost of 54. In contrast, using the RRT\* algorithm resulted in faster convergence and a solution with a significantly lower cost. These results demonstrate the superiority of RRT\* in terms of efficiency and performance compared to the basic RRT algorithm.[3]

1. Informed RRT\*

The Informed RRT\* algorithm has gained attention in the field of robot motion planning for its improvement on existing approaches. Through simulation, it performs comparably to basic RRT\*, but it demonstrated a significant improvement in more complex scenarios (Fig. 5)[4]. The key feature of Informed RRT\* is its ability to focus the search on a specific area of the planning problem, reducing its dependence on the size and complexity of the domain. This allows for more efficient and accurate identification of topologically distinct paths. In the absence of obstacles, Informed RRT\* is capable of degenerating the ellipse to a line(Fig. 6)[4].

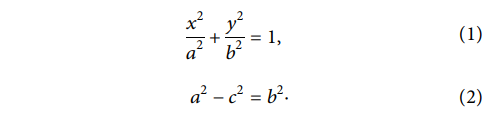


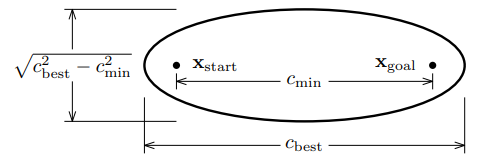
*Fig. 5[4]*



*Fig. 6[4]*

To generate an ellipse as the new sampling domain, the following steps are necessary. Figure 7 has shown that the sampling space is generated based on certain parameters when using Informed RRT\*. The start and end points are represented by and , respectively, while is the distance from one node to the other, and represents the the path length found after the first iteration of informed RRT\* is finished. To obtain the initial path, the algorithm calculates the distance between the two vertices of the ellipse (）in equations (1) and (2). , on the other hand, represents the distance between the two focal points of the sampling domain. By using these two parameters, the Informed-RRT\* algorithm can generate a sampling space in the form of an ellipse. It is also possible to obtain the ellipse's eccentricity, /, and the conjugate diameter[4]. As the algorithm continues to optimize the path, decreases, and the sampling space becomes smaller, leading to faster convergence[5].





*Fig. 7[4]*

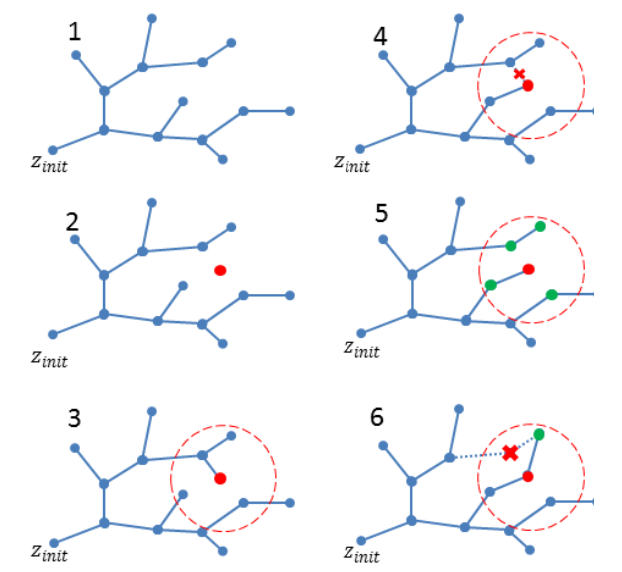
Overall, Informed RRT\* is an extended algorithm developed from RRT\*, which uses heuristics to narrow the focus of the search to specific subsets of the original problem domain. This approach allows the algorithm to be more efficient by only considering relevant parts of the problem. The effectiveness of this approach is limited by the current solution cost, as the algorithm is not able to focus the search when the associated search space is larger than the entire planning problem. Similarly, the algorithm can only shrink the search space down to the minimum size defined by the optimal solution. In other words, the algorithm depends on the current cost and cannot search beyond the optimal solution.

1. RRT\*FN

Researchers have established the mathematical guarantees of the optimality(distance) of the solution from the RRT\* given that we have no limitations on memory consumption and the time taken to execute the planning[6]. However, no such computing device exists in real life. We need a close-to-optimal solution within a limited time frame with limited memory, as commonly found in embedded devices/ controllers.

RRT\*Fixed Nodes (RRT\*FN) is an extension of the RRT\* paradigm of incrementally rewiring nodes to generate tree-like structures when a new node is created but limits the maximum number of nodes (and hence memory) created by removing nodes.

The algorithm starts with the generation of a uniformly random point in the state space of the system, conveniently denoted by . It finds all the nodes within the neighborhood of the and *steer* the tree from the to , considering the kinematic constraints with motion primitives unless the path intersects the . It chooses a parent node in the neighborhood based on the lowest cost and *not* the shortest distance and then connects them. Here, it checks if the total nodes exceed the fixed number M; if they do, it removes a node with no child nodes. This ensures that the number of nodes stays fixed at M, hence the name RRT\* Fixed Nodes.



*Fig. 8: Visualization of the algorithm for a 2D case showing insertion, rewiring, and removal, as seen in the original paper[7]*

The algorithm uses less memory but has limitations in the rate of convergence. The algorithm was benchmarked for 10 maps with a fixed starting point. We saw that it had a slower convergence as compared to RRT\* but converged to near-optimal with significantly lesser memory. The two algorithms behave very similarly to each other until the maximum number of nodes is reached, and then the algorithm tree is similar to a sparse RRT\* algorithm tree.

We can extend these methods with dynamic rewiring techniques to enable faster convergence and planning in dynamic environments.

Works Cited

[1] Noreen, I. (2016). *A Comparison of RRT, RRT\* and RRT\*-Smart Path Planning Algorithms*, *VOL.16*(NO.10), 1–8.

[2] LaValle, S. M. (n.d.). *PLANNING ALGORITHMS*. Retrieved January 6, 2023, from http://www.lavalle.pl/planning/book.pdf

[3] Aaron, T. B. (2018, January). *Wolfram Demonstrations Project*. Rapidly Exploring Random Tree (RRT) and RRT\*. Retrieved January 5, 2023, from https://demonstrations.wolfram.com/RapidlyExploringRandomTreeRRTAndRRT/

[4] J. D. Gammell, S. S. Srinivasa and T. D. Barfoot, "Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 2997-3004, doi: 10.1109/IROS.2014.6942976.

[5] Jun Dai, Dongfang Li, Junwei Zhao, Yanqin Li, "Autonomous Navigation of Robots Based on the Improved Informed-RRT Algorithm and DWA", Journal of Robotics, vol. 2022, Article ID 3477265, 9 pages, 2022.

[6] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," International Journal of Robotics Research, vol. 30, pp. 846-894, Jun 2011.

[7] Adiyatov, O., & Varol, H. A. (2013). Rapidly-exploring random tree-based memory efficient motion planning. 2013 IEEE International Conference on Mechatronics and Automation. doi:10.1109/icma.2013.6617944

.