DIRRT*-Connect: An Optimal Ellipsoidal Heuristic Sampling-based Path Planning Algorithm using Gradient Descent Optimization

Yidong Guo* and Zihao Liu[2]

Abstract*— Sampling-based planning algorithms have demonstrated strong performance in high-dimensional path planning problems. However, achieving rapid convergence toward high-quality solutions remains challenging. In this paper, we propose the deformable informed RRT*-Connect(DIRRT*-Connect) algorithm, which integrates a whole-tree gradient descent optimization scheme into a bidirectional RRT-based framework. The proposed method first employs an RRT-Connect framework to rapidly obtain an initial feasible solution through greedy bidirectional expansion. Informed ellipsoidal sampling is activated to restrict the sampling domain according to the current best solution cost. Meanwhile, a gradient-descent-based deformation strategy is applied to branch nodes, enabling global trees refinement and accelerating convergence toward the optimal solution. By coupling passive sampling concentration induced by informed heuristics with active structural concentration driven by gradient-based optimization, DIRRT*-Connect significantly improves convergence and solution quality. Experimental evaluations in both two-dimensional and six-dimensional spaces indicate that DIRRT*-Connect exhibits superior convergence performance and can rapidly obtain the first feasible solution. Compared with RBI-RRT*, the DIRRT*-Connect algorithm demonstrates superior performance in six-degree-of-freedom path planning with a path length reduction by 38% under the same planning time.

I. INTRODUCTION

Path planning is primarily applied in fields such as autonomous driving and intelligent transportation, industrial and logistics applications, as well as drones and specialized robots[1], [2], [3]. Path planning algorithms can generally be categorized into two major classes: map-based methods and sampling-based methods. Map-based planning algorithms typically rely on explicit environment representations, such as grid maps or graph structures, and are well suited for low-dimensional planning problems in two-dimensional or three-dimensional workspaces[4], [5], [6]. However, their computational complexity increases rapidly with the dimensionality of the state space, rendering them less practical for high-dimensional configuration spaces. In contrast, sampling-based algorithms explore the configuration space by probabilistic sampling without requiring an explicit discretization of the entire state space[7], [8], [9], [10]. By incrementally constructing a random sampled tree(RRT) over the feasible region, these methods effectively handle high-dimensional planning problems and are therefore more suitable for robotic manipulators and other systems with large configuration spaces. Thus RRT is widely used for path planning.

A series of variants of the RRT algorithm are used for optimal or efficient path planning. RRT-Connect[11] is an efficient path planning algorithm that grows two Rapidly-exploring Random Trees simultaneously from the start and goal configurations. It enhances search speed by employing a greedy bidirectional strategy where each tree alternately expands towards the newest node of the other, facilitating quicker connection in the state space. RRT*[12] guarantees asymptotic optimality which means the path cost converges towards the global optimum as the number of iterations increases through rewiring the tree structure by selecting a better parent node for a new node and then recursively optimizing the paths of nearby nodes to ensure minimal cost from the start. Informed-RRT*[13] focuses subsequent sampling within an ellipsoidal subset of the state space defined by the current best path cost after finding an initial solution improving the convergence rate towards an optimal solution compared to the RRT* algorithm.

Informed RRT*[13] accelerates convergence toward the optimal solution by restricting sampling to an ellipsoidal subset of the state space. However, this adaptive sampling strategy is activated only after a feasible solution with finite cost is obtained. When the solution cost remains infinite, Informed RRT* behaves identically to the standard RRT*, limiting the speed of discovering the first feasible solution. Therefore, a common strategy is to first use RRT-Connect to quickly obtain a feasible solution without emphasizing optimality, thereby triggering the informed sampling mechanism at an earlier stage. Both the RBI-RRT*[14] and Hybrid RRT[15] algorithms adopt this idea: they first rapidly obtain a feasible solution using RRT-Connect and then leverage Informed RRT* to find the optimal solution. Here we adopt this concept as well.

Although RBI-RRT*[14] demonstrates better near-optimal solution convergence, it relies solely on stochastic sampling and local rewiring for path refinement. Since node positions remain fixed after insertion, optimization is entirely driven by the quality of newly generated samples. In high-dimensional or topologically complex environments, this sampling-dependent mechanism may lead to slow convergence and premature stagnation within suboptimal homotopy classes. As indicated by the DRRTd algorithm[16], performing gradient descent calculations on nodes for space deformation can accelerate the optimization process and shorten the time required to find the feasible solution[16].

Inspired by DRRTd[16], we firstly introduce gradient-

*Yidong Guo is the first author and corresponding author. E-mail: zzgyd97@163.com

[2]Zihao Liu is with the Research Center for Intelligent Robotics, School of Astronautics, Northwestern Polytechnical University, and National Key Laboratory of Aerospace Flight Dynamics, Northwestern Polytechnical University, Xi'an 710072, China

Code: https://github.com/zgyd97/DIRRT-Connect based branch deformation to two expanding trees which enables active geometric adjustment of tree nodes and more robust convergence toward high-quality solutions. RRT-Connect is first used to find the quick feasible solution. Then the sampling space is constrained using an elliptical sampling region to limit the scope of the potential optimal solution. Gradient descent optimization promotes the aggregation of RRT nodes around promising topological regions. DIRRT*-Connect algorithm is proposed in this paper.

(1) Unlike previous gradient-based deformation methods designed for single-tree planners[16], [17], the proposed DIRRT*-Connect algorithm extends integer gradient descent to a bidirectional tree structure for the first time. After RRT-Connect obtains an initial feasible solution, an ellipsoidal informed sampling strategy is activated to restrict the sampling domain. The sampled nodes are subsequently refined through gradient-based deformation, encouraging the search tree to concentrate around promising regions. By coupling passive sampling restriction with active geometric refinement, DIRRT*-Connect achieves faster and more stable convergence toward high-quality solutions.

(2) DIRRT*-Connect follows the initial solution generation strategy of RBI-RRT* by employing the RRT-Connect algorithm with a greedy extension mechanism to rapidly obtain the first feasible solution. Owing to the greedy connection strategy, the time required to find the initial feasible solution is significantly shorter than that of the standard RRT algorithm. Compared with DRRTd, the proposed method achieves a 79.5% improvement in the time to first feasible solution in 2D map environments.

(3) Compared with RBI-RRT*, DIRRT*-Connect consistently generates smoother trajectories with lower final cost. The proposed method further demonstrates enhanced performance in high-dimensional path optimization scenarios. Extensive evaluations in a six-degree-of-freedom robotic manipulator configuration space indicate that DIRRT*-Connect achieves a 38% average reduction in final path cost relative to RBI-RRT* which validates the superior convergence efficiency and practical effectiveness of the proposed framework in robotic manipulator motion planning tasks.

II. PROBLEM STATEMENT

The optimal path planning problem is defined in this section. Let the configuration space be denoted as $X \in \mathbb{R}^n$, where n represents the dimensionality of the state space. The space is partitioned into the obstacle region X_{obs} and the collision-free region $X_{free} = X \setminus X_{obs}$. A feasible solution is defined as a continuous path, given an initial state $X_{start} \in X_{free}$ and a goal state $X_{goal} \in X_{free}$, the objective of the path planning problem is to find a continuous path σ and $\sigma(s) \in X_{free}$ for all $s \in [0, 1]$.

The optimal path planning problem can be formulated as searching for a collision-free feasible path σ that minimizes a given cost function [16]. Formally, it can be defined as

$$\sigma^* = \arg\min \left\{ C(\sigma) \;\middle|\; \begin{array}{l} \sigma(0) = x_{start},\ \sigma(1) = x_{goal}, \\ \forall s \in [0,1],\ \sigma(s) \in X_{free} \end{array} \right\}. \tag{1}$$

Algorithm 1 DIRRT*-Connect Algorithm

1: **Notation:** T_1 and T_2 denote the RRT trees rooted at the start and goal state respectively, and C_{best} denotes the minimum cost among all feasible solutions found so far.
2: **Initialize:** $C_{\text{best}} \leftarrow \infty$, $b_{\text{optimize}} \leftarrow$ false
3: **while** not TerminationCondition **do**
4: $\quad x_{\text{rand}} \leftarrow \text{InformedSample}(x_{\text{start}}, x_{\text{goal}}, C_{\text{best}})$
5: $\quad$ **if** Extend(T_1, x_{rand}, b_{optimize}) != TRAP **then**
6: $\quad\quad x_{\text{new1}} \leftarrow T_1$.lastNode
7: $\quad\quad$ **repeat**
8: $\quad\quad\quad$ status $\leftarrow$ Extend(T_2, x_{new1}, b_{optimize})
9: $\quad\quad$ **until** status != ADVANCE
10: $\quad\quad$ **if** status == REACH **then**
11: $\quad\quad\quad b_{\text{optimize}} \leftarrow$ true
12: $\quad\quad\quad x_{\text{new2}} \leftarrow T_2$.lastNode
13: $\quad\quad\quad X_{\text{soln}} \leftarrow X_{\text{soln}} \cup \{x_{\text{new1}}, x_{\text{new2}}\}$
14: $\quad\quad\quad C_{\text{best}} \leftarrow \min(\text{cost}(X_{\text{soln}}))$
15: $\quad\quad$ **end if**
16: $\quad\quad$ swap(T_1, T_2)
17: $\quad$ **end if**
18: **end while**
19: **return** T_1, T_2

As stated in DRRT[15], when the cost function is differentiable, gradient descent optimization can be applied. Thus C denotes the Euclidean distance here.

III. THE DIRRT*-CONNECT ALGORITHM

In this section, a new path planning algorithm, DIRRT*-Connect, is proposed. Inspired by the core part of DRRTd[15] and following the framework of the RBI-RRT* algorithm[14], our method first aims to rapidly obtain a feasible solution. Subsequently, gradient descent optimization is applied to accelerate the convergence toward the optimal solution.

A. Deformable informed RRT-connect*

The pseudocode of the proposed DIRRT*-Connect algorithm is presented in Algorithm 1. Inspired by RBI-RRT*[13], our method is structured into two distinct phases. Rapid initial solution acquisition phase with b_optimze setting false, where a feasible path is quickly obtained. Global refined optimization phase (Algorithm 1, Lines 10–14), where the b_optimize turns true and solution is further improved toward optimality through systematic refinement.

During rapid feasible solution acquisition phase, the proposed method first adopts an RRT-Connect framework with a greedy extension strategy to rapidly obtain an initial feasible path without any optimization. Two trees are grown simultaneously from the start and goal states, respectively, in a bidirectional manner. Once a feasible connection is established, the two trees are reconstructed. In addition, an informed heuristic sampling strategy is incorporated to restrict the sampling domain after a feasible solution is found, thereby improving convergence efficiency toward high-quality solutions. Unlike conventional incremental rewiring approaches, the two entire trees are subsequently treated as global optimization objects[16]. A gradient descent procedure is then applied to refine the cost function of the two RRT trees. The key difference between the proposed algorithm and RBI-RRT* lies in the process of searching for the optimal solution, specifically the Extend function which will be further explained in Algorithm 2.

Algorithm 2 Extend(T, x_{target}, b_{optimize})

1: **Notation:** T denotes RRT tree to expand, x_{target} denotes target node to reach
2: $x_{\text{nearest}} \leftarrow \text{FindNearestNode}(T, x_{\text{target}})$
3: $x_{\text{new}} \leftarrow \text{Steer}(x_{\text{nearest}}, x_{\text{target}})$
4: $x_{\text{near}} \leftarrow \text{FindNearNodes}(T, x_{\text{new}})$
5: **if** not CollisionFree(x_{nearest}) **then**
6:     **return** TRAP
7: **end if**
8: **if** b_{optimize} **then**
9:     $x_{\text{parent}} \leftarrow \text{ChooseParent}(x_{\text{near}}, x_{\text{new}})$
10:     $T \leftarrow T \cup \{(x_{\text{parent}}, x_{\text{new}})\}$
11:     $b \leftarrow \text{Branch}(x_{\text{new}})$
12:     DeformInOrder(b)
13:     Rewire(T, x_{new})
14: **else**
15:     $T \leftarrow T \cup \{(x_{\text{nearest}}, x_{\text{new}})\}$
16: **end if**
17: **if** isReachable(x_{nearest}, x_{target}) **then**
18:     **return** REACH
19: **else**
20:     **return** ADVANCE
21: **end if**

The DIRRT*-Connect algorithm is illustrated in Fig. 1. As shown in Fig. 1(a), after RRT-Connect rapidly finds a feasible solution, b_optimize turns true, and further expansion and optimization are performed to ultimately obtain an optimal solution that satisfies the required criteria, as shown in Fig. 1(c). After the new node is determined, the branch identifies all deformable nodes except which are the leaf nodes and the start node, and deforms the corresponding RRT branch, as illustrated in Fig. 1(d). Once a new node is determined, all nodes within the branch are deformed along the negative gradient direction of the cost function over the entire RRT tree. "Branch" is defined as the RRT subtree rooted at the parent of the new node, excluding all leaf nodes. This branch structure has been demonstrated to achieve an optimal trade-off between computational overhead and optimization effectiveness[17].

Finally, the algorithm returns RRT tree T1 and T2 that contain multiple feasible solutions connecting the start node and the goal node, as well as the optimal solution obtained through gradient-based optimization.

In contrast to the RBI-RRT* algorithm, the proposed method differs in the shaded portion of Algorithm 2. Once an initial solution is obtained, the variable b_optimize is set to true, and a branch deformation strategy is introduced to guide

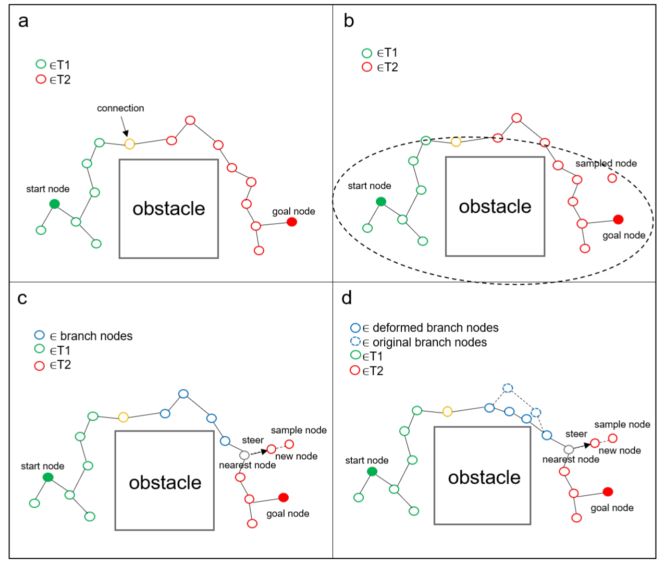


Fig. 1: Illustration of the whole procedure of DIRRT*-Connect (a) Random trees during RRT-Connect. The Green dots represent RRT started from start node and the red dots represent RRT started from goal node. (b) Sampling new node with Ellipsoidal Heuristic (c) Deform branch Nodes using gradient descent (d) The RRT after gradient descent opptimization

Algorithm 3 DeformInorder(b)

Notation: b denotes the branch nodes to be deformed
while not TerminationCondition **do**
    for $b_i \in b$ **do**
        $t = 1$
        while not TerminationCondition **do**
            $t = \beta \cdot t$
        end while
        temp $= x_{b_i} - t\left(\frac{\partial J_V}{\partial x_{b_i}}\right)$
        if !isFeasible(temp) **then** continue
        end if
        if $\text{cost}(P_{b_i}) + \text{distance}(P_{b_i}, \text{temp}) \leq C_{\text{best}}$ **then**
            $x_{b_i} = \text{temp}$
            updateCosts(C_{b_i})
        end if
    end for
end while

the subsequent optimization process. Specifically, before performing the rewiring operation for a newly generated node, a gradient-descent-based optimization is applied to the branch nodes of the tree, which deforms and optimizes the branch structure associated with the new node[7].

In Algorithm 2, Line 9, the ChooseParent function selects the parent of the new node from the k nearest neighbors by choosing the node that minimizes the path cost from the start node to the new node. In Algorithm 2, Line 13, the Rewire function checks the same set of k nearest neighbors and updates the parent of any neighbor whose path cost can

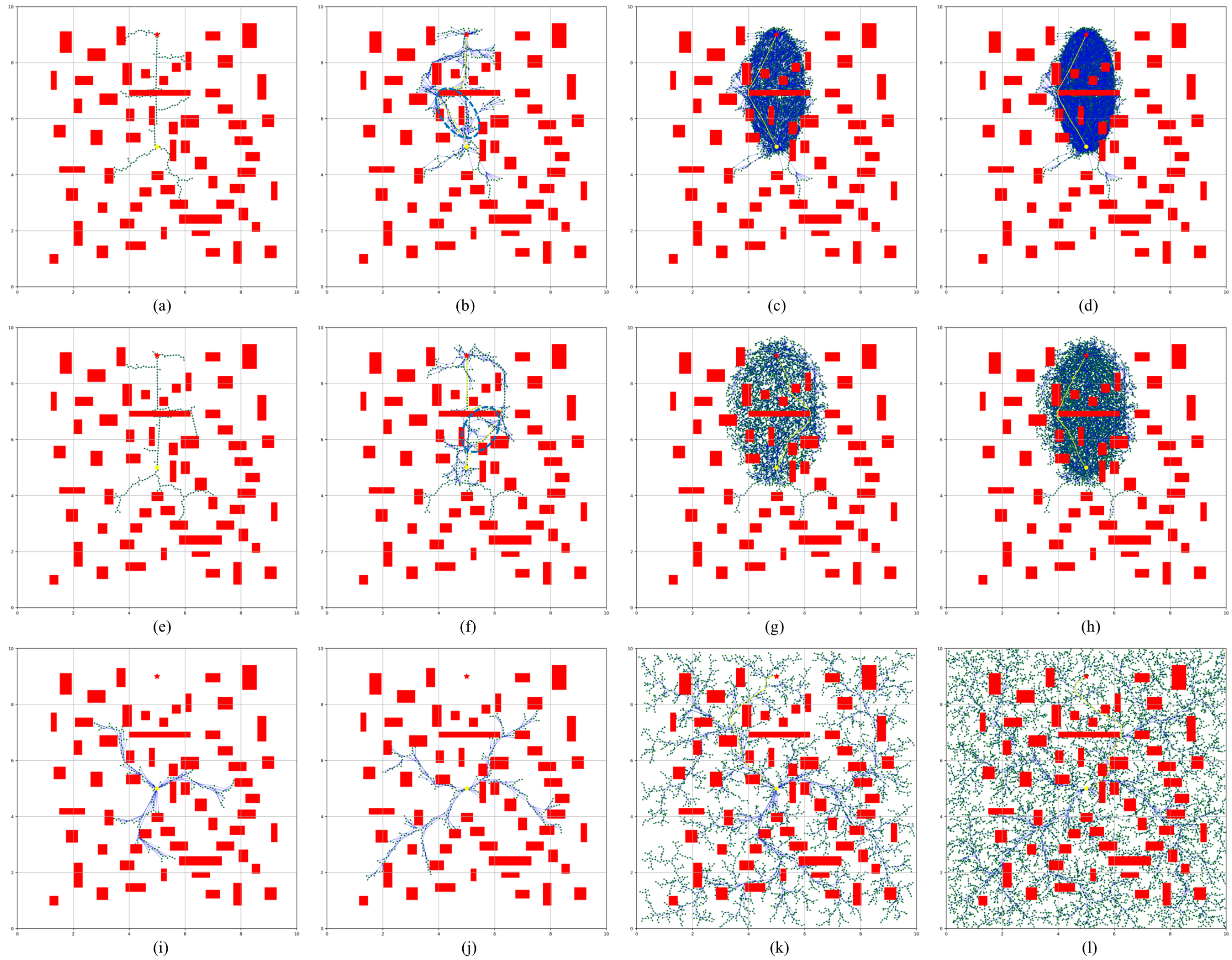


Fig. 2: The evolution of the tree computed by DIRRT*-Connect, RBI-RRT*, DRRTd algorithm is shown in (a)-(d), (e)-(h), (i)-(l) respectively. The configuration of the trees (a) is at 250 iterations, (b) is at 500 iterations, (c) is at 5000 iterations, and (d) is at 10000 iterations.

be reduced through the new node.

B. Tree deform optimization

During the refinement phase, the RRT trees will be deformed using gradient descent optimization[16]. Specifically, informed sampling is first employed to generate a new node. After the new node is validated, a gradient-descent-based deformation is applied to the branch nodes, followed by the rewiring operation to further optimize the tree structure. The pseudocode of Gradient-descent optimization is shown in Algorithm 3.

As shown in Algorithm 3, according to the Armijo condition, each deformable node is updated along the negative gradient direction. This step is the core part of the proposed algorithm, where node positions are adjusted to ensure a decrease in the loss function of the whole RRT. J_v is defined as the summation of the cost functions over all nodes in the RRT tree.

$$J_v = \sum_{i \in T} c_i \tag{2}$$

For a non-root node in the tree T, its accumulated cost satisfies:

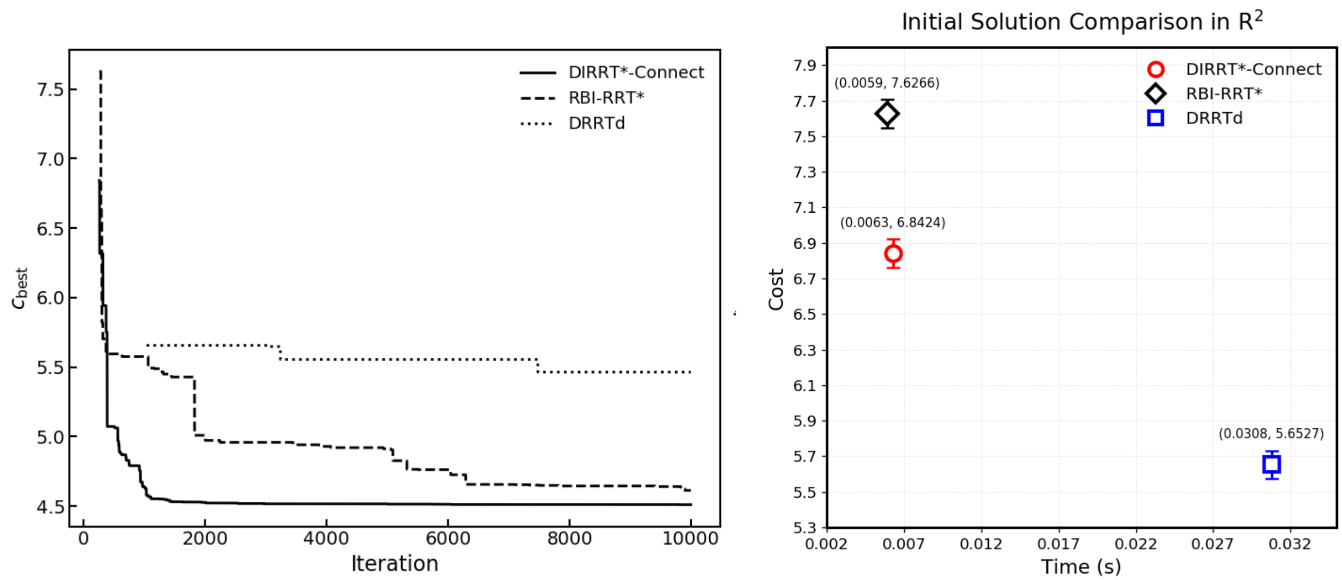
$$c_i = c(x_{pi}, x_i) + c_{pi} \tag{3}$$

where x_{pi} denotes the parent of node x_i. Therefore, the objective term J_v can be equivalently expressed as:

$$J_v = \sum_{i \in T} \sum_{j \in T} \text{num\_edge} \cdot c(x_i, x_j) \tag{4}$$

where num_edge represents the number of times the edge cost $c(x_i, x_j)$ contributes to the overall cost function, i.e., the number of descendants of node x_j in RRT tree plus one.

$$\text{num\_edge} = \text{num\_desc}(j) + 1 \tag{5}$$



(a)

(b)

Fig. 3: A set of simulations result in $\mathbb{R}^2$ state space (a) Convergence of best Cost vs. iterations (b) Comparison of mean computation time to get the initial solution and initial path cost

For different nodes, gradient descent is performed with respect to their spatial positions. Given the selected cost function, we compute partial derivatives along each dimension of the n-dimensional configuration space. Consequently, in J_v, only the terms containing the spatial coordinates of the currently deformed node n matter. Thus, J_v depends solely on the edges incident to the node under optimization, and can be simplified as

$$J_v = d_n \cdot c(x_{p_n}, x_n) + \sum_{i \in C_n} d_i \cdot c(x_n, x_i) + C \tag{6}$$

where p_n and C_n denote the parent and child of node n, respectively. C is the constant. d_n equals $1 + \mathrm{num\_des}(n)$ with $\mathrm{num\_des}(n)$ the number of descendants of node n. Similarly d_i equals $1 + \mathrm{num\_des}(i)$ with $\mathrm{num\_des}(i)$ the number of descendants of node i. If the updated deformation point remains collision-free and maintains connectivity with both its parent and child nodes, and if the resulting cost c_{temp} is smaller than the original cost c_n, the spatial position of the current node is updated accordingly. Subsequently, the costs of all its descendant nodes are recursively updated (Algorithm 3, Lines 12–13).

IV. EXPERIMENTS AND SIMULATIONS

In this section, we conduct both simulations in 2 degree-of-freedom and 6 degree-of-freedom space. In Part A, DIRRT*-Connect, RBI-RRT* and DRRTd algorithm are evaluated and compared under two dimensional map. The evaluation metrics include the optimal cost function value and the number of iterations required for convergence to the optimal solution. In Part B, both DIRRT*-Connect and RBI-RRT* are applied to simulations of six-degree-of-freedom robotic manipulator for further validation.

A. Simulations in 2D spaces

To evaluate the effectiveness of the proposed DIRRT*-Connect algorithm, simulation experiments were conducted in a two-dimensional environment. All experiments were performed on a laptop running Ubuntu 20.04, and the algorithm was executed for a total of 10000 iterations. We conducted comparative path planning experiments under an identical map configuration using RBI-RRT*, and DRRTd. The intermediate results of the algorithm at different stages are illustrated in Fig. 2.

The path planning task aims to compute the shortest collision-free path connecting the start and goal configurations. The size of the simulated map is 10*10, with obstacles randomly distributed throughout the environment following the work of Oktay Arslan[18]. The start position is indicated by a yellow circle at coordinates (5, 5), while the goal position is marked by a red star at (5, 9). As shown in Fig. 2, the planned path gradually converges toward a near-optimal solution as the number of iterations increase.

The RRT trees are visualized in blue, with tree nodes depicted in green. Comparing the circled region of Fig. 2(b) and (f), as the iterations progress, the nodes after the deform operation tend to concentrate around the trunk of the original RRT tree and the edge of the obstacles. This distribution differs from the passive sampling restriction introduced by traditional informed sampling methods; Instead, it actively guides the exploration toward high-quality sampling regions.

Compared with the RBI-RRT* and DRRTd algorithm, under the same number of iterations, the proposed DIRRT*-

Connect algorithm produces paths with lower cost and significantly improved smoothness. After 10000 iterations, the final average best path cost C_best obtained by RBI-RRT* and DRRTd is 4.64 and 5.65 respectively, whereas the proposed method achieves a lower cost of 4.49. As a result, DIRRT*-Connect achieves a noticeably lower final path cost, highlighting its superior refinement ability and improved asymptotic convergence performance under the same computational budget.

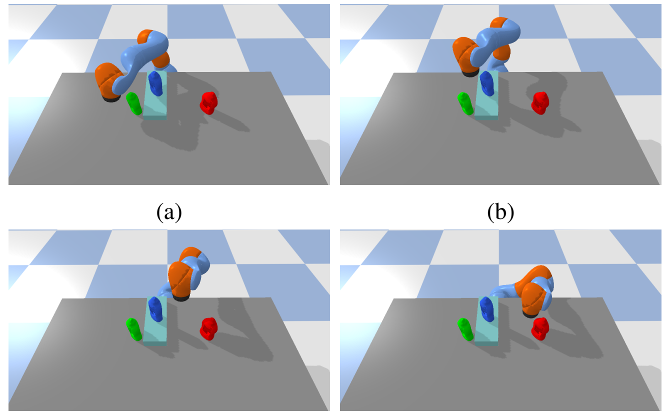
For each algorithm, the evolution of the best path cost C_best with respect to iteration, as well as the relationship between the initial solution cost and time, were analyzed. Fig.3(a) presents the evolution of the best path cost C_best over 10000 iterations for DIRRT*-Connect, RBI-RRT* and DRRTd under identical settings. As shown in the figure, DIRRT*-Connect converges rapidly toward the optimal solution. The path cost reaches its minimum value and stabilizes after approximately 1500 iterations. DIRRT*-Connect demonstrates a consistently stronger optimization capability throughout the entire process. In particular, DIRRT*-Connect achieves a rapid reduction in path cost during the early iterations and quickly converges to a lower-cost solution, while RBI-RRT* demonstrates a slower and more gradual decrease in path cost over the optimization process. In contrast, DRRTd operates in a relatively large sampling space, making it more prone to being trapped within a local topological configuration during exploration. As a result, its path refinement performance is less effective compared to the proposed DIRRT*-Connect algorithm. The value of C_best for DRRTd decreases nonlinearly with respect to the number of iterations due to changes in the topological structure of the path. In particular, transitions across different homotopy can lead to a significant reduction in C_best, thereby substantially accelerating the path cost improvement process.

As shown in Fig. 3(b), both DIRRT*-Connect and RBI-RRT* employ the RRT-Connect strategy to obtain the initial feasible solution. Owing to the greedy bidirectional expansion mechanism of RRT-Connect, the time required to find the first feasible solution is significantly shorter than that of DRRTd. DIRRT*-Connect finds the first feasible solution in 0.0063s, which is 79.5% faster than DRRTd, while maintaining competitive path quality.

B. Simulations for 6 DOF manipulators

In this section, to evaluate the performance of DIRRT*-Connect in high-dimensional configuration spaces, we conduct simulations in PyBullet using a six-degree-of-freedom KUKA LBR iiwa manipulator. Shown in Fig. 4, the green bottle denotes the start position, the red bottle indicates the goal position, and all other objects serve as obstacles. We vary both the initial configurations and obstacle placements and perform multiple independent simulation trials. Fig. 4(a)–(d) depict four representative snapshots of the manipulator during its motion execution process.

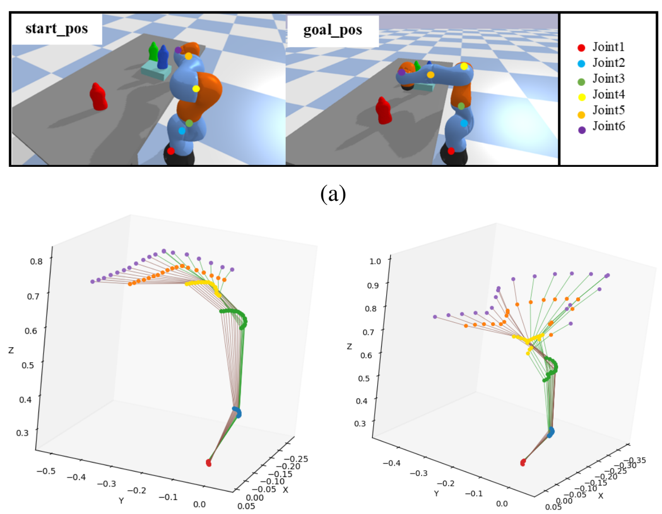
In high-dimensional configuration spaces, even when adopting an informed sampling strategy, the planner can still become trapped in local optima, remaining confined



(a) (b) (c) (d)

Fig. 4: The simulation of 6 DOFs manipulator path planning scenario.(a)-(d) denote different steps for manipulator in one run

within a suboptimal topological region. Moderately enlarging the k-nearest neighborhood radius enhances the rewiring quality to boost transition among topologies. The quality of Rewire mechanism significantly influences the branch deform quality. It aims to facilitate cross-topology exploration toward the global optimum and prevent convergence to locally suboptimal homotopy classes. To demonstrate the effectiveness of DIRRT*-Connect in high-dimensional solution spaces, comparative experiments are conducted with the RBI-RRT* algorithm.



(a) (b) (c)

Fig. 5: The simulations setup and comparison of trajectories between DIRRT*-Connect algorithm and RBI-RRT*. (a) the simulation setup (b)the trajectory of DIRRT*-Connect (c) the trajectory of RBI-RRT*

The initial and goal configurations of the robotic manipulator are shown in Fig. 5. The highlighted markers indicate the six joint nodes of the manipulator. Path planning from the initial configuration to the goal configuration is conducted using the DIRRT*-Connect and RBI-RRT* algorithms. The node trajectories from the start configuration to the goal configuration generated by DIRRT*-Connect and RBI-RRT* are illustrated in Fig. 5(b) and Fig. 5(c), respectively. Under the same number of iterations, DIRRT*-Connect achieves a final path cost of 1.63, while RBI-RRT* results in a final path cost of 3.75. Compared with RBI-RRT*, DIRRT*-Connect shortens the path length by 56%. A total of 100 simulation trials in 20s are conducted for both algorithms, and the results are summarized in Table I.

As reported in Table I, both DIRRT*-Connect and RBI-RRT* achieve a 100% success rate over 100 trials with a planning time limit of 20s. The average path length generated by DIRRT*-Connect is 1.67, compared to 2.71 for RBI-RRT*. This corresponds to a 38% reduction in average path length, highlighting the effectiveness of the proposed DIRRT*-Connect in improving path quality.

TABLE I: Comparison of DIRRT*-Connect and RBI-RRT* in 6-DOF Manipulator Planning

	Planning in 20s	
Method	DIRRT*-Connect	RBI-RRT*
Success rate	100%	100%
Path cost	1.67	2.71

The experimental results demonstrate that DIRRT*-Connect significantly outperforms RBI-RRT* in terms of both convergence speed and final solution quality. By introducing a gradient descent optimization scheme over the global tree cost function, the convergence of the path planning process is substantially accelerated. Specifically, global gradient descent optimization is applied to both trees during the bidirectional expansion process, which effectively improves the overall convergence performance of the planner. The proposed method provides a promising approach for path planning of six-degree-of-freedom robotic manipulators.

V. CONCLUSIONS

We propose a novel algorithm, DIRRT*-Connect, whose core idea is to first rapidly obtain a feasible solution using the RRT-Connect strategy. During the optimal solution refinement phase, DIRRT*-Connect integrates informed sampling with gradient-based optimization on whole trees to accelerate convergence toward the optimal path. Compared with RBI-RRT*, the proposed method demonstrates superior performance in high-dimensional planning problems, particularly in six-degree-of-freedom robotic manipulator path planning tasks.

The simulations of 6 DOFs manipulator show DIRRT*-Connect algorithm has a 38% reduction in average path length compared with RBI-RRT*. Overall, the proposed framework provides a practical and scalable approach for high-dimensional optimal motion planning, offering improved convergence behavior and enhanced performance in robotic manipulation tasks.

ACKNOWLEDGMENT

The authors would like to thank Hongkai for fruitful discussions.

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[2] M. Elbanhawi and M. Simic, “Sampling-based robot motion planning: A review,” *IEEE Access*, vol. 2, pp. 56–77, 2014.

[3] Y. K. Hwang and N. Ahuja, “A potential field approach to path planning,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 23–32, Feb. 1992.

[4] P. E. Hart, N. J. Nilsson and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[5] S. Koenig and M. Likhachev, “D* Lite,” in *Proc. AAAI Conference on Artificial Intelligence*, Edmonton, Canada, 2002, pp. 476–483.

[6] A. Stentz, “The Focussed D* Algorithm for Real-Time Replanning,” in *Proc. International Joint Conference on Artificial Intelligence*, 1995.

[7] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” Technical Report, Computer Science Dept., Iowa State Univ., 1999.

[8] J. D. Gammell, S. S. Srinivasa and T. D. Barfoot, “Batch Informed Trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 2015, pp. 3067–3074.

[9] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[10] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI, USA, 1999, pp. 473–479, vol.1.

[11] J. J. Kuffner and S. M. LaValle, “RRT-connect: An efficient approach to single-query path planning,” in *Proceedings 2000 IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, USA, 2000, pp. 995–1001, vol.2.

[12] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[13] 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,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, IL, USA, 2014, pp. 2997–3004.

[14] F. Chen, Y. Zheng, Z. Wang, W. Chi and S. Liu, “RBI-RRT*: Efficient Sampling-based Path Planning for High-dimensional State Space,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, 2024, pp. 8721–8727.

[15] R. Mashayekhi, M. Y. I. Idris, M. H. Anisi and I. Ahmedy, “Hybrid RRT: A Semi-Dual-Tree RRT-Based Motion Planner,” *IEEE Access*, vol. 8, pp. 18658–18668, 2020.

[16] F. Hauer and P. Tsiotras, “Deformable rapidly-exploring random trees,” in *Proc. Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, 2017.

[17] H. Ye, C. Xu and F. Gao, “STD-Trees: Spatio-temporal Deformable Trees for Multirotors Kinodynamic Planning,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, 2023, pp. 1200–1206.

[18] O. Arslan and P. Tsiotras, “Use of relaxation methods in sampling-based algorithms for optimal motion planning,” in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013, pp. 2421–2428.