ANN-CMCGS: Generalizing Continuous Monte Carlo Graph Search with Approximate Nearest Neighbors

Christoph Scherer and Wolfgang Hönig

***Abstract*— Continuous Monte Carlo Graph Search (CMCGS) enables state reuse in continuous domains but still relies on a layered, acyclic structure, limiting its effectiveness. We introduce ANN-CMCGS, a generalized, non-layered formulation to detect approximate transpositions in continuous spaces via Approximate Nearest Neighbor (ANN) search. By allowing arbitrary directed graphs and enabling incremental reuse across decision steps, ANN-CMCGS demonstrates improved exploration efficiency and success rates in challenging continuous domains.**

I. Introduction

Online decision making in continuous state spaces with sparse rewards and uncertainty is a challenge in robot motion planning. Monte Carlo Tree Search (MCTS) [1], [2] is a promising approach [3], [4], but suffers from high branching factors and redundant exploration in continuous domains, limiting its applicability [5], [6].

Continuous Monte Carlo Graph Search (CMCGS) [7] extends the idea of state reuse through transpositions of MCGS [8], [9] to continuous spaces using state abstractions and clustering. Although effective for certain problems, the layered directed acyclic graph (DAG) structure limits connectivity and reuse, which is restrictive in robot path planning where cycles and backtracking naturally occur [10].

We propose ANN-CMCGS, a non-layered Monte Carlo Graph Search formulation that detects approximate transpositions in continuous domains via ANN search, allowing arbitrary directed graphs with cycles and incremental reuse of the search graph. Our results show that ANN-CMCGS matches CMCGS in control-oriented settings and substantially outperforms it in higher-dimensional and non-holonomic motion planning problems.

II. Continuous Monte Carlo Graph Search with Approximate Nearest Neighbors

We formulate the decision-making problem as a Markov Decision Process (MDP) with the standard MCTS steps of *selection*, *expansion*, *rollout* , and *backpropagation* [2]. Selection uses the upper confidence bounds (UCT) node selection criterion with a stochastic criterion for progressive widening (epsilon-greedy) and rollouts of random actions [1], [2], [6], [7].

This work has been supported by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG).

The paper has been accepted as an extended abstract at AAMAS 2026.

All authors are with Technical University of Berlin, Berlin, Germany, {c.scherer, hoenig}@tu-berlin.de

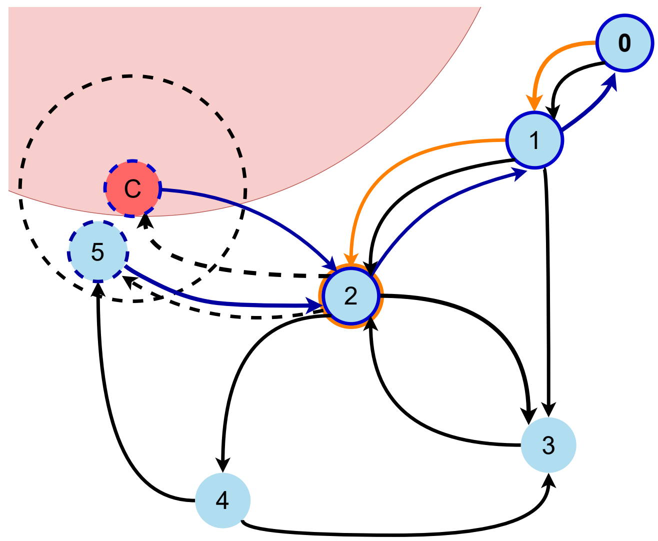

Fig. 1. Illustration of graph expansion in ANN-CMCGS. Black arrows indicate existing edges, while orange and blue arrows mark node selection and backpropagation. (1) Starting from the root node 0, node 2 is selected for expansion. (2) From 2, a candidate node C is sampled; red denotes truncation due to collision with an obstacle. (3) The ANN insertion finds node 5 as a reachable and non-truncated transposition within the radius around C and adds both 5 and C and the corresponding actions as new children (dashed). (4) During backpropagation, C, 5, its edges and all nodes and edges along the playout path are updated toward the root (blue).

The key departure from the existing CMCGS formulation is the replacement of layered, clustered transpositions with reachability-based approximate transpositions detected via ANN search over the entire search graph. This yields arbitrary directed graphs with cycles, removes the strict layer structure, and supports variable-length edges and incremental reuse across decision steps.

Contrary to the implicit strong assumptions about suitable state-space metrics and implicit controllability for state clustering approaches, we explicitly assume access to a controller as well as only a heuristic state-space metric, circumventing problems with complex, non-holonomic robot dynamics. The result is a highly connected and feasible roadmap-like graph that drastically improves node sampling efficiency.

A. Approximate transpositions

The step corresponding to the MCTS expansion is illustrated in Fig. 1. Possible approximate transpositions for any sampled candidate node are identified using ANN radius queries. Neighbors passing the controller reachability check are connected via edges, increasing graph connectivity without increasing node count. The candidate node is inserted into the graph and ANN index only if its state contains novel information, such as differing termination or truncation outcomes. This selective insertion prevents the loss of critical states while keeping the graph compact.

During backpropagation, all nodes and edges added during expansion are updated once, followed by standard backpropagation along the playout path, consistent with existing MCGS formulations.

B. Handling Cycles

Unlike layered approaches that enforce a directed acyclic graph, graph-wide transpositions introduce cycles. This poses potential problems such as infinite loops during node selection as well as backpropagation. To prevent this, we track the current playout path to select each node at most once and backpropagate only along this path. While tree/DAG-based methods ensure that statistics propagate monotonically toward the root, different playout paths in cyclic graphs may induce updates in opposing directions. This complicates the theoretical analysis of convergence and completeness, which we do not address in this work. Empirically, however, we did not observe behaviors such as oscillatory expansion or repeated switching between nodes. We partly attribute this to the stochastic progressive widening approach used.

C. Online Planning Update

To support incremental planning across decision steps, the robot's current state is matched to the existing graph via a $k = 1$ ANN query. The state is either merged with its nearest neighbor or inserted as a new node, after which subsequent expansions naturally reconnect it to the existing graph through the transposition mechanism.

III. Evaluation

We evaluate ANN-CMCGS on sparse-reward navigation tasks with obstacles under multiple 2D dynamics models, including single integrator, double integrator, and unicycle dynamics, extending the prior 1D single integrator, control-focused benchmark [7]. The evaluation focuses on two aspects: (i) the ability to efficiently explore continuous state spaces with cycles, and (ii) robustness across higher-dimensional and non-holonomic dynamics, where layered and clustering based approaches are expected to fail.

ANN queries use the *HNSWlib*[1] implementation of Hierarchical Navigable Small-World graphs [11] for online index updates, while reachability checks employ CasADi [12] to remove the effects of controller quality.

[1]https://github.com/nmslib/hnswlib

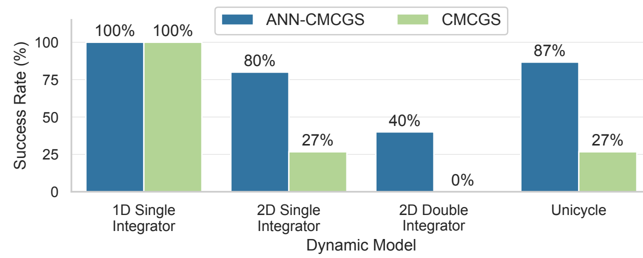

Fig. 2. Success rates at fixed node sampling budgets for ANN-CMCGS (ours) and CMCGS.

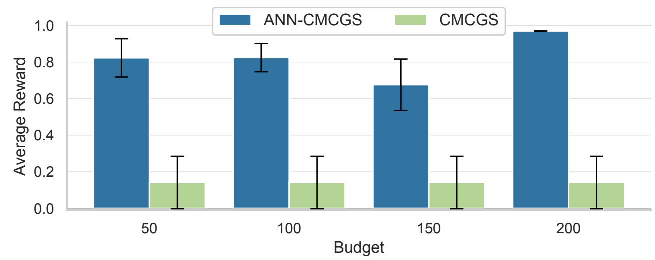

Fig. 3. Performance of ANN-CMCGS (ours) and CMCGS with unicycle dynamics across increasing node expansion budgets. Results averaged over 20 episodes; error bars show standard error of mean.

Results (Fig. 2, 3) show ANN-CMCGS matches CMCGS in 1D control tasks and substantially outperforms it in higher-dimensional and non-holonomic scenarios, efficiently exploiting cycles, state reuse and incremental graph building.

IV. Conclusion

We present ANN-CMCGS, a non-layered extension of Continuous Monte Carlo Graph Search that leverages ANN queries to detect approximate transpositions in continuous domains. By supporting arbitrary directed graphs with cycles and incremental graph updates, ANN-CMCGS improves state reuse, exploration efficiency, and planning success compared to the baseline CMCGS, particularly in higher-dimensional and non-holonomic motion planning tasks.

Although theoretical evaluation on convergence and optimality remain future work, our empirical evaluation suggests that ANN-CMCGS provides a practical and effective framework for online motion planning in continuous, sparse-reward environments.

REFERENCES

[1] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *Proceedings of the European Conference on Machine Learning (ECML)*. Berlin, Heidelberg: Springer, 2006, pp. 282–293.

[2] R. Coulom, "Efficient selectivity and backup operators in monte-carlo tree search," in *Proceedings of International Conference on Computers and Games (ICCG)*. Berlin, Heidelberg: Springer, 2006, pp. 72–83.

[3] S. Bone, L. Bartolomei, F. Kennel-Maushart, and M. Chli, "Decentralised multi-robot exploration using monte carlo tree search," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 7354–7361.

[4] B. Rivière, J. Lathrop, and S.-J. Chung, "Monte carlo tree search with spectral expansion for planning with dynamical systems," *Science Robotics*, vol. 9, no. 97, 2024.

[5] C. Mansley, A. Weinstein, and M. Littman, "Sample-based planning for continuous action markov decision processes," in *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, vol. 21, 2011, pp. 335–338.

[6] A. Couëtoux, J.-B. Hoock, N. Sokolovska, O. Teytaud, and N. Bonnard, "Continuous upper confidence trees," in *Learning and Intelligent Optimization (LION)*, C. A. C. Coello, Ed. Berlin, Heidelberg: Springer, 2011, pp. 433–445.

[7] K. Kujanpää, A. Babadi, Y. Zhao, J. Kannala, A. Ilin, and J. Pajarinen, "Continuous monte carlo graph search," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2024, pp. 1047–1056.

[8] A. Saffidine, T. Cazenave, and J. Méhat, "UCD : Upper confidence bound for rooted directed acyclic graphs," *Knowledge-Based Systems*, vol. 34, pp. 26–33, 2012, a Special Issue on Artificial Intelligence in Computer Games (AICG).

[9] J. Czech, P. Korus, and K. Kersting, "Improving alphazero using monte-carlo graph search," in *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, vol. 31, no. 1, 2021, pp. 103–111.

[10] L. E. Kavraki, P. Svestka, J. L. 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, 1996.

[11] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 42, no. 04, pp. 824–836, 2020.

[12] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.