Distributed Constraint-Based Search using Multi-Hop Communication

Hannah Lee[1,2], James Motes[1], Zachary Serlin[2], Marco Morales[1], and Nancy M. Amato[1]

I. INTRODUCTION

Decentralized multi-agent pathfinding (MAPF) is a crucial area of study within artificial intelligence and robotics, focusing on enabling multiple autonomous agents to navigate a shared environment without conflicts. This approach is valuable for applications such as warehouse automation [1] and vehicle routing [2], [3]. Decentralized MAPF distributes the decision-making process among individual agents, enhancing scalability, robustness, and flexibility. It is particularly important for managing large-scale dynamic environments where centralized control is impractical or intractable.

Decentralized MAPF faces significant challenges under multi-hop communication assumptions, such as complexities in ensuring timely and accurate data exchange. The assumption that networks can manage planning without assigning specific communication routes or planning roles often fails in practice [4], [5]. To address these shortcomings, we leverage concepts from Hierarchical Composition Conflict-Based Search (HC-CBS) [6] to introduce our extension, Hierarchical Composition for Multi-hop Distributed Communication (HCMDC). This approach strategically determines communication paths within distributed networks along which the MAPF problem is built and solved. By doing so, we consider the intricacies of multi-hop communication while ensuring safe and efficient navigation. Our preliminary experiments show the advantages of distributed planning in random environments.

II. METHODOLOGY

In HC-CBS, we introduce a framework to parallelize Conflict-Based Search (CBS) [7] by distributing the search process across multiple threads, breaking the MAPF problem into subproblems and hierarchically merging their solutions. Building on HC-CBS, we now present a method to utilize constraint-based search within a multi-hop communication network to address the distributed MAPF problem. This approach includes the network navigator, which identifies communication paths, and the pathfinder, which constructs and solves MAPF subproblems along these paths.

[1] University of Illinois Urbana-Champaign, USA. {hannah9, jmotes2, moralesa, namato}@illinois.edu.
[2] MIT Lincoln Laboratory, USA. {hannah.lee, zachary.serlin}@ll.mit.edu.
[3] ITAM, Ciudad de México, 01080, México. {marco.morales}@itam.mx.
DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.

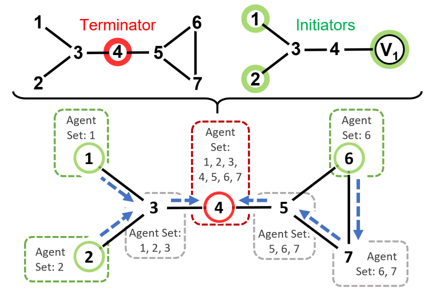

Fig. 1. Example network where green initiators relay information through blue arrows to a red terminator. This process incrementally builds subproblems that merge to solve the full problem at the terminator.

A. Network Navigator

The network navigator leverages the communication network to assign roles to agents, designating them as a terminator, initiator, or intermediary, as illustrated in Figure 1. The navigator assigns these roles and establishes a simple information path from initiators to the terminator. Within a network, there can be multiple initiators and intermediaries, but only a single designated terminator. Initiators start the search process and initiate the information path. Intermediaries receive information from other nodes but can transmit it only once. The terminator marks the end of the information path, completing the search process.

The network navigator selects the optimal terminator by identifying the node that best balances the computational workload among agents, which is the node with the highest average and lowest standard deviation of agents transmitted by neighboring nodes. Each neighboring node transmits information about all of its preceding agents; thus, by maximizing and evenly distributing the data transmitted from neighboring nodes, the navigator ensures a balanced influx of data at the terminator, maintaining an even workload distribution among agents. After identifying the terminator, the network is reduced to be acyclic by representing strongly connected components as virtual nodes. Nodes with a single neighbor are designated as initiators while all others are assigned as intermediaries. Virtual nodes assigned as initiators are further analyzed to identify specific initiators within the component to ensure all nodes are visited via a simple path.

B. Pathfinder

Given a set of communication paths from the initiators to the terminator, the pathfinder leverages HC-CBS to solve the MAPF problem. An example of this process is shown in Figure 2. Each initiator initially plans its optimal path and transmits it to the next agent. The receiving agent merges its

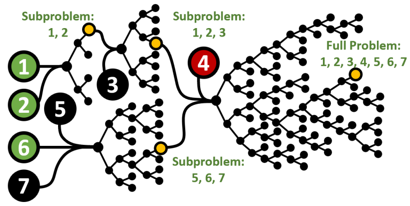

Fig. 2. Example of Figure 1's search trees - each hop merges prior agents' solutions (orange search node) with a new agent to create a new subproblem.

optimal path with the solution information from preceding agents, using this data to create the starting point for its subproblem search. To reduce network traffic, only the solution node is transmitted between agents. With each hop, an agent is added to increasingly complete the formulation of the full MAPF problem. As with HC-CBS, the goal of the navigator and pathfinder is to maximize the work completed within each subproblem before reaching the terminator, effectively distributing the workload among the network agents.

To illustrate an example, let us refer to Figure 2 and denote agents as A_i. A_1 starts by sending its path to A_2. A_2 merges its path with A_1's, creating subproblem $S_{1,2}$. HC-CBS then finds conflict-free paths for both agents. This solution is passed to A_3, forming subproblem $S_{1,2,3}$. HC-CBS is again used to find conflict-free paths, and the process continues until the terminator agent, A_4. At A_4, the paths from $S_{1,2,3}$ and $S_{5,6,7}$ are merged with A_4. The full MAPF problem is then solved, and the solution paths are sent back to each agent.

III. PRELIMINARY RESULTS

We conducted preliminary experiments using an environment with randomly generated obstacles, from [8], in which we ran 25 different scenarios with randomly sampled motion tasks. The average runtime results, capped at 10 minutes, compare our distributed HCMDC with a network navigator (NetNav) and a random navigator (Rand) against CBS, our centralized baseline. The networks were generated assuming a line-of-sight communication protocol, with agents further connected by proximity to ensure a connected network. Rand HCMDC, used to test the effectiveness NetNav HCMDC, chooses a random terminator and the minimum set of receivers to ensure valid information paths. The work analysis shows the average total and evaluated node counts per subproblem's search tree, calculated by averaging the node counts from each subproblem for a given problem instance.

As shown in Figure 3, our results indicate that distributing the workload among agents in HCMDC improves scalability as the number of agents increases. Despite transmitting only the solution node, rather than the entire search frontier as in HC-CBS, we observed no significant degradation in plan quality. Additionally, we report the average node counts per subproblem to show the impact of using the network navigator as a heuristic for grouping and merging subproblems. The lower average node counts for NetNav HCMDC compared to Rand HCMDC indicate that the heuristic effectively distributes meaningful work amongst the subproblems. However, the random navigator performs surprisingly well,

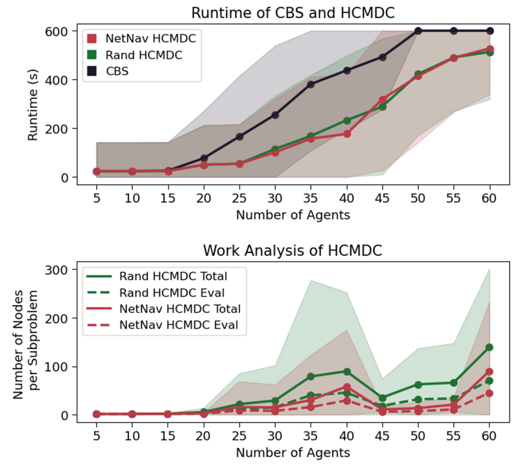

Fig. 3. Plots showing runtime and work analysis. The skewed runtime distribution is due to capping scenarios when they exceed the runtime limit. For detailed simulation results, please refer to the linked video[4].

indicating the need for further analysis of various network navigator heuristics across different environments. Despite the lower node counts, the runtimes for the network navigator and the random navigator are very similar. We suspect these heuristics may demonstrate clearer advantages when applied to environments with different topologies or in networks with varying connectivity. In conclusion, our experiments show the potential of using HCMDC in distributing workloads under multi-hop communication assumptions.

IV. CONCLUSION

Our main contribution is a new heuristic for creating subproblems in distributed settings, utilizing network communication and greedy solution passing to reduce network traffic. We present promising results with both a network navigator and a random navigator to explore different types of network heuristics. This lays the initial groundwork for future exploration of the best heuristics to apply in various environments and communication networks on robots with varying planning and communication capabilities. While we assume homogeneous agents, heterogeneous teams could have roles assigned based on their capabilities, dynamics, or environmental restrictions. Lastly, although we only transmitted the solution node in our examples, transmitting more of the search frontier would enhance solution quality. HC-CBS maintains completeness by merging the entire search frontier, but to reduce network traffic, we chose to transmit only the solution node. HCMDC holds significant potential for improvement and our work effectively addresses the distributed MAPF problem with multi-hop communication.

V. ACKNOWLEDGEMENTS

This work was supported in part by the IBM-Illinois Discovery Accelerator Institute, by the Center for Networked Intelligent Components and Environments (C-NICE) at the University of Illinois, and by Asociación Mexicana de Cultura AC.

[4]`https://youtu.be/minRJbehKwQ`

References

[1] L. Custodio and R. Machado, "Flexible automated warehouse: a literature review and an innovative framework," *The International Journal of Advanced Manufacturing Technology*, vol. 106, pp. 533–558, 2020.

[2] S. Xie, J. Hu, P. Bhowmick, Z. Ding, and F. Arvin, "Distributed motion planning for safe autonomous vehicle overtaking via artificial potential field," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21531–21547, 2022.

[3] J. L. Lázaro, J. Garcıa, M. Mazo, A. Gardel, P. Martın, I. Fernández, and M. Marrón, "Distributed architecture for control and path planning of autonomous vehicles," *Microprocessors and microsystems*, vol. 25, no. 3, pp. 159–166, 2001.

[4] O. Salzman and R. Stern, "Research challenges and opportunities in multi-agent path finding and multi-agent pickup and delivery problems," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1711–1715, 2020.

[5] J. H. Jung, S. Park, and S.-L. Kim, "Multi-robot path finding with wireless multihop communications," *IEEE Communications Magazine*, vol. 48, no. 7, pp. 126–132, 2010.

[6] H. Lee, J. Motes, M. Morales, and N. M. Amato, "Parallel hierarchical composition conflict-based search for optimal multi-agent pathfinding," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7001–7008, 2021.

[7] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.

[8] R. Stern, N. Sturtevant, A. Felner, S. Koenig, H. Ma, T. Walker, J. Li, D. Atzmon, L. Cohen, T. Kumar, *et al.*, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Proceedings of the International Symposium on Combinatorial Search*, vol. 10, pp. 151–158, 2019.