Monte Carlo Tree Search with Useful Cycles for Motion Planning

Bilal Kartal

***Abstract*— An autonomous robot team can be employed for continuous and strategic coverage of arbitrary environments for different missions. Stochastic multi-robot planning is a very powerful technique for several problem domains. Many planning problems, e.g. swarm robotics, coverage planning, and multi-robot patrolling, require high degree of coordination which yields scalability issues for traditional joint-space planners. The other main challenge for traditional joint-space planners is the exploration versus exploitation trade-off during policy search. Exploration versus exploitation dilemma is very well studied in the context of Multi-armed bandit problem. Stochastic sampling based planners employ the multi-armed bandit theory to address the aforementioned challenges. Particularly in this work, we have been investigating stochastic tree search approaches in policy space for the multi-robot patrolling problems. We proposed a new variant of Monte Carlo Tree Search algorithm for life-long policies by exploiting periodic trajectories of the robot team.**

I. Introduction

Multi-robot systems are nowadays commonly used to perform critical tasks, such as search and rescue operations, intelligent farming, mine sweeping and environmental monitoring [9], [11], [12]. All of these problems require coverage of environments based on some optimization criteria. One instance of these coverage planning problems is the multi-robot patrolling problem where multiple robots must cover patrol terrain strategically in a coordinated fashion to prevent intrusions. The multi-robot patrolling problem has been an active research area within the last decade, especially as more and more autonomous robots are available for surveillance tasks at low costs. One common problem formulation is an optimization based one minimizing the idleness, i.e. the maximum time difference between any visits to any node, and this problem is $\mathcal{NP}$-hard as it can be reduced to the well-known TSP problem. Recent works includes extending the longevity of the patrolling task by considering robot batteries [1], decentralized patrolling based on Gaussian processes [7], and patrolling in case of coordinated attacks [10].

II. Contributions

In this section, we present our current contributions and ongoing research. We study the multi-robot patrolling problem for two types of intruder models, i.e. probabilistic static, and dynamic intruders. We formulate the patrolling policy generation problem as a tree search problem and as a baseline we employ Monte Carlo Tree Search (MCTS) [5]. MCTS is a breakthrough algorithm in particularly AI community and it has been also applied to several other domains [4], [6], [8]. MCTS can successfully search in large domains by using random sampling. The algorithm is anytime and converges to optimal solutions given enough time and memory for finite-horizon problems.

The author is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA. {bilal}@cs.umn.edu

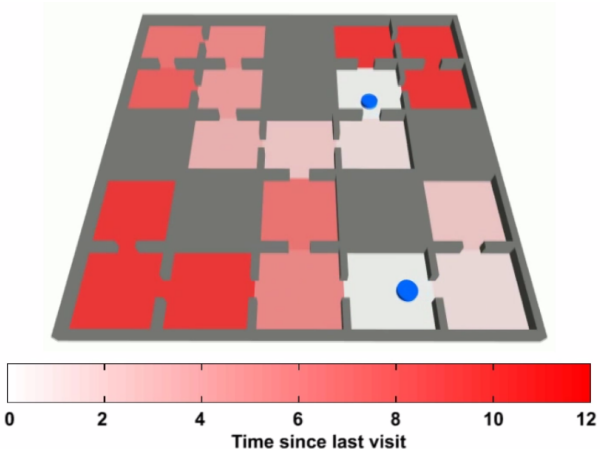

Fig. 1. MCTS-UC patrol policy is employed by 2 patrollers. Both patrollers start at top left cell, but the grid is partitioned into two cycles emergently and covered continuously by patrollers.

A. Monte Carlo Tree Search with Useful Cycles

One of the main challenges to adapt MCTS to the patrolling domain is the ability to generate infinite length policies; the policies generated by MCTS are valid for a small time horizon while patrolling task has to be performed continuously. This is a very important difference from the single time coverage problems. We address the continuous patrolling challenge by introducing Monte Carlo Tree Search with Useful Cycles, MCTS-UC, which augments standard MCTS with *cyclic nodes* to return infinite, cyclic policies [2] with convergence guarantees for finite horizon problems. We present an overview of our main contribution, MCTS-UC, in Figure 2.

We define a useful cycle as a set of patrolling trajectories that starts and ends in the same vertex set for the robot team. Therefore, we exploit the spatial similarity of visited vertices of patrollers, i.e. whether the same set of vertices are visited between any two states or not, to determine a useful cycle. In terms of search tree structure, MCTS-UC creates artificial cyclic nodes which represent continuous policies. These nodes will be part of the tree search during exploration-exploitation.

Consider, for example, two equivalent nodes A and B as shown in Fig. 2(a). Given these nodes, a cyclic node, node C, is created as a sibling arm to node B and its cyclic parent

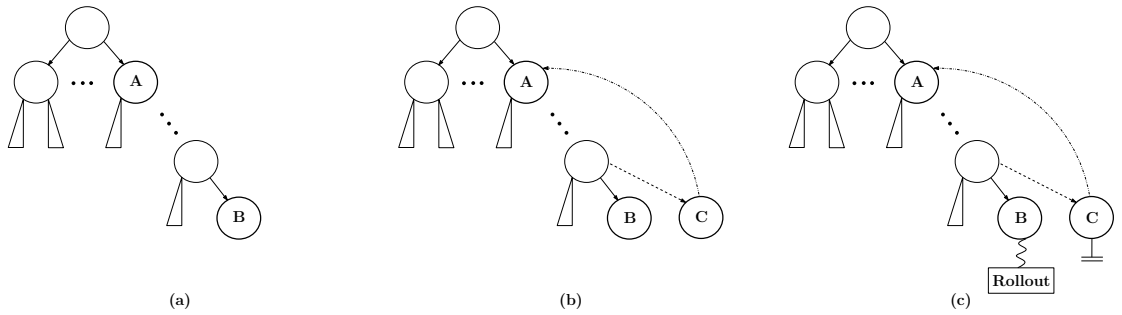

Fig. 2. Overview of Monte Carlo Tree Search with Useful Cycles: (a) During the back-propagation of node B, node A is found to have the same state. (b) A new cyclic node C is created to capture the cycle. (c) While node B is evaluated with standard rollouts, node C is evaluated cyclically.

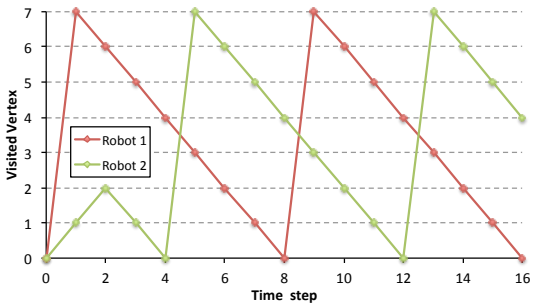

Fig. 3. Trajectories of 2 robots on a perimeter, $|V| = 8$. Both robots start at the same vertex and they adjust their placements to equidistant vertices and patrol in the same direction continuously. V_7 and V_0 are adjacent vertices.

node is set to node A as depicted in Fig. 2(b). Node C will be part of search just as any ordinary arm. When node B is selected after creating node C, we do a roll-out and expand the tree as in standard MCTS. However, if the cyclic node C is selected in further iterations of MCTS-UC, our algorithm creates a cyclic action buffer by pushing actions one by one while walking up the tree from itself to its cyclic parent (i.e. node A) and continuously performs these actions (Fig. 2(c)), evaluates the cyclic policy and back-propagates the score through the *non-cyclic* parent nodes as in standard MCTS.

Our proposed method can generate near-optimal policies for a team of robots for small environments in real-time (and in larger environments in under a minute). By incorporating additional planning heuristics, i.e. Iterative search heuristic that we proposed earlier [3], we are able to plan coordinated patrolling paths for teams of several robots in large environments quickly on commodity hardware. We present trajectories of 2 robots patrolling on a perimeter on Figure 3. A video presenting patrolling policies in different environments can be seen at `http://motion.cs.umn.edu/r/MCTS-UC`

III. Conclusion and Future work

Given that MCTS-UC is a sparse sampling based search technique which does not require an admissible heuristic function to evaluate policies, it can easily be extended to arbitrary patrolling and/or pursuit evasion problem scenarios where intruder or environment models can vary. Therefore, current system can be used as a base framework for future extensions.

For future work, we plan to further speed up search by employing parallelization techniques which will also increase level of scalability for larger problem instances. Secondly, we plan to investigate patrolling problem by considering uncertainty on the patrollers observation models. We envision that patrolling problem becomes more realistic and interesting with imperfect observation model. Lastly, as a long term goal, we want to study intruders that can learn and adapt to the patroller strategies where the trade-off between policy randomization and coverage of environments becomes more challenging.

References

[1] E. Jensen, M. Franklin, S. Lahr, and M. Gini. Sustainable multi-robot patrol of an open polyline. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4792–4797. IEEE, 2011.

[2] B. Kartal, J. Godoy, I. Karamouzas, and S. J. Guy. Stochastic tree search with useful cycles for patrolling problems. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015.

[3] B. Kartal, J. Koenig, and S. J. Guy. Generating believable stories in large domains. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[4] B. Kartal, J. Koenig, and S. J. Guy. User-driven narrative variation in large story domains using monte carlo tree search. In *Autonomous agents and multi-agent systems*, pages 69–76, 2014.

[5] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, pages 282–293. Springer, 2006.

[6] V. Lisý, M. Lanctot, and M. Bowling. Online monte carlo counterfactual regret minimization for search in imperfect information games. 2015.

[7] A. Marino, G. Antonelli, A. P. Aguiar, and A. Pascoal. A new approach to multi-robot harbour patrolling: Theory and experiments. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1760–1765. IEEE, 2012.

[8] D. Perez, E. J. Powley, D. Whitehouse, P. Rohlfshagen, S. Samothrakis, P. I. Cowling, and S. M. Lucas. Solving the physical traveling salesman problem: Tree search and macro actions. *Computational Intelligence and AI in Games, IEEE Transactions on*, 2014.

[9] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. B. Kosmatopoulos. Multi-robot three dimensional coverage of unknown areas. *The International Journal of Robotics Research*, 2012.

[10] E. Sless, N. Agmon, and S. Kraus. Multi-robot adversarial patrolling: facing coordinated attacks. In *Autonomous agents and multi-agent systems*, pages 1093–1100, 2014.

[11] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler. Sensor planning for a symbiotic uav and ugv system for precision agriculture. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5321–5326. IEEE, 2013.

[12] J. Yu, S. Karaman, and D. Rus. Persistent monitoring of events with stochastic arrivals at multiple stations. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5758–5765. IEEE, 2014.