Finding Bounded Suboptimal Multi-Agent Path Planning Solutions Using Increasing Cost Tree Search (Extended Abstract)

Faten Aljaloud[1]
Department of Computer Science
University of Denver
Denver, CO, USA
faten.aljaloud@gmail.com

Nathan R. Sturtevant
Department of Computer Science
University of Denver
Denver, CO, USA
sturtevant@cs.du.edu

Abstract

The Increasing Cost Tree Search (ICTS) algorithm is used to produce optimal solutions to the multi-agent path finding problem (MAPF). In this problem, multiple agents are trying to reach their goals without conflicting with each other, while minimizing the total cost of the paths. ICTS has been shown to be very effective in finding optimal solutions. In this paper we consider the problem of finding solutions with bounded suboptimality by changing the order in which ICTS searches its increasing cost tree. With a variety of strategies, we are unable to consistently and significantly reduce the cost of ICTS. Further experimentation suggests why significantly more work is needed to modify ICTS to find suboptimal solutions.

Introduction and Motivation

The problem of multi-agent pathfinding (MAPF) has recently attracted significant interest in the search community, with a variety of work on optimal and sub-optimal approaches being developed. This work considers the problem of finding bounded-suboptimal solutions to the MAPF problem. Our hypothesis was that simple modifications to an existing algorithm (ICTS) would allow us to find suboptimal solutions more quickly than optimal solutions. Instead, we found that our modifications do not have the intended affect, and explain why this is the case.

Formally speaking, the MAPF problem is defined by a graph $G = (V, E)$ and a set of agents, $a_1 \dots a_k$. Associated with each agent is a set of start and goal vertices, $\{(s_1, g_1), (s_2, g_2) \dots (s_k, g_k)\}$. The goal is for each agent to find a path between their respective start and goal locations without crossing a vertex or edge at the same time as another agent. MAPF models a wide variety of problems in robotics, games, and other scenarios, such as traffic in an intersection.

Optimal approaches to MAPF must face the combinatorial explosion of agent locations in order to find the shortest solution. Suboptimal solutions fall into several categories. The first possibility is to decouple the problem and consider agents completely independently. With no or little communication between independent agents, these approaches are fast, but may not even be complete. There are also a class of approaches which can find suboptimal solutions in polynomial time (Röger and Helmert 2012).

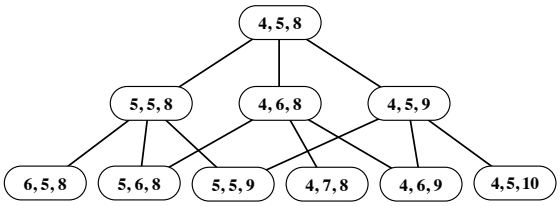

Figure 1: The high-level increasing-cost tree.

One common way of finding suboptimal solutions, not yet well-explored in MAPF, is by modifying optimal algorithms. One example is weighted A* (Pohl 1970), which weights the heuristic by a multiple of w, producing a solution that is at most w times costlier than the optimal solution at a fraction of the cost. The heuristic weighting makes the search more greedy so it progresses more quickly to the goal, without having to exhaustively prove optimality. Wagner and Choset (2011) show that this approach has promise. Using this approach for MAPF should be cheaper than finding an optimal solution, but with better solution quality than the polynomial-time approaches. We chose to investigate modifications of the ICTS algorithm to find suboptimal solutions.

ICTS

The ICTS algorithm (Sharon et al. 2011) uses a two-level approach to find solutions to the MAPF problem. The top-level approach is a tree-search which attempts to find the optimal set of path costs for all agents. Given a potential set of costs from the top-level approach, the low-level search generates, for each agent, all single-agent paths that have the same cost as the top-level solution. Then, these individual paths are combined until either the given cost bound is proven infeasible, or a solution is found. The top-level search is performed in a best-first manner, ensuring that the first solution found will be optimal.

We give a simple example of the top-level approach for a 3-agent problem in Figure 1. The root of the tree is labeled (4, 5, 8). This node asks the question: 'is there a solution to the given MAPF problem with the agents having solution costs 4, 5, and 8?'. This question is resolved by a low-level search. Consider agent 3 with cost 8. Ignoring the

[1]This author received support for this work from Al Imam Mohammad Ibn Saud Islamic University (IMSIU).

Table 1: Suboptimal approaches to ICTS compared to the original algorithm.

	3*3 grid		8*8 grid	
Algorithm	# Solved	Time	# Solved	Time
ICTS	43	6.2s	50	1.2s
AAC	50	191ms	42	4.7s
Makespan	50	30ms	22	18.9s

other agents, all solutions with cost 8 for agent 3 are found and stored in a Multi-Valued Decision Diagrams (MDD). An MDD is an efficient way of storing the exponentially-growing number of paths in space linear in the size of the map (N) and time steps allowed for the solution (t), or $O(Nt)$ space. Suppose that two agents MDDs both have the same node at the same depth of the MDD, with no other nodes at the same depth. Then, when comparing MDDs we can see that no solution exist for both agents, as any solution for either agent must pass through the same location at the same time. This is part of a more general process that merges MDDs together to form the set of all possible solutions.

We built an implementation of ICTS on top of Independence Detection (ID) (Standley 2010). ID starts a search with each agent in a separate group, being considered independent of other agents. The optimal path for each group is then found independently. If the solutions for any groups conflict, these groups are merged and re-solved with ICTS. The cost of search with ID is dominated by the largest group of agents with conflicts.

Suboptimal ICTS

We approach the problem of finding suboptimal solution to MAPF problems by modifying the top level of the ICTS search. Instead of searching the top-level tree in a best-first manner, we propose two alternate approaches; we have tried other variants as well. These are:

All-Agent-Costs (AAC): ICTS only increments the cost of a single agent from a parent to a child in the ICT. The AAC approach increments the cost of all agents by one at each subsequent node in the search tree.

Makespan (MS): The initial bound for each agent is set to the maximum optimal single-agent path cost of all agents. At each subsequent node, we increase the cost of all agents by one. (The ICT grows linearly.)

Results and Analysis

First, we present the results of our suboptimal variants as compared to the original ICTS algorithm in Table 1. The results are shown in 3 by 3 grids using 6 agents and 8 by 8 grids using 10 agents. For each grid, we ran both optimal ICTS and suboptimal approaches. There were 50 problems chosen at random as the input data set. The running time of each problem was limited to 5 minutes. The resulted running time is the average for solved problems in the dataset. In addition, the number of solved problems is shown. Similar results were obtained in other problem variations.

These results show in general the approach works reasonably on small problems, but not on the larger ones. In problems that are very hard but still solvable for ICTS, the suboptimal approaches usually provides a significant improvement. In other problems, however, it often is significantly slower than ICTS.

These results suggest that either we are searching the tree using the wrong policies, or that our approach of modifying the top-level search strategy is not sufficient for finding suboptimal solutions quickly. While there may be better ways to structure the low-level MDD search in order to more easily find suboptimal solutions, we explain here why re-ordering the top-level search does not work well on many problems.

There are three factors that contribute to the cost of ICTS. The first cost is the cost of searching the high-level ICTS tree. This tree grows exponentially if all costs for all agents are considered. The second cost is the cost of building the MDDs for each agent. The cost of a single MDD is $O(Nt)$, so increasing the cost of a solution for an agent increases the cost of building its MDD linearly. The final cost is the cost of merging the MDDs together to find solutions. In general this process can grow exponentially in the number of agents, although in many cases infeasible cases can be quickly detected. The more solutions that are available for the agents, the larger the merged MDDs.

The key relationship is that the deeper we search in the ICT tree, the more expensive it becomes to find solutions. Thus, if we jump deeper into the ICT tree looking for suboptimal solutions, we often incur more overhead searching at nodes deeper in the tree than we save by avoiding nodes shallower in the tree. While weighted A* gives room for a greedy search to find the solution more quickly, allowing more time in the MDDs actually makes more expensive to merge agents' paths and find feasible solutions.

Conclusion

In this short paper, we have proposed two approaches for searching the top-level ICT tree with the aim of finding bounded suboptimal solutions quickly. But, instead of finding solutions significantly more quickly, we find that the cost of merging low-level ICT nodes outweighs the savings from avoiding top-level ICT nodes. While this can be seen as a negative result, this work provides a deeper understanding of ICTS and provides a result that was not obvious when we undertook this study. It is an open question how the problem topology influences these results and whether there is other information that can be used to leverage ICTS to find suboptimal solutions quickly.

References

[Pohl 1970] Pohl, I. 1970. Heuristic search viewed as path finding in a graph. *Artificial Intelligence* 1:193–204.

[Röger and Helmert 2012] Röger, G., and Helmert, M. 2012. Non-optimal multi-agent pathfinding is solved (since 1984). In Borrajo, D.; Felner, A.; Korf, R. E.; Likhachev, M.; López, C. L.; Ruml, W.; and Sturtevant, N. R., eds., *SOCS*. AAAI Press.

[Sharon et al. 2011] Sharon, G.; Stern, R.; Goldenberg, M.; and Felner, A. 2011. The increasing cost tree search for optimal multi-agent pathfinding. In *IJCAI*, 662–667.

[Standley 2010] Standley, T. 2010. Finding optimal solutions to cooperative pathfinding problems. In *AAAI*, 173–178.

[Wagner and Choset 2011] Wagner, G., and Choset, H. 2011. M*: A complete multirobot path planning algorithm with performance bounds. In *IROS*, 3260–3267. IEEE.