Artificial Intelligence 219 (2015) 40–66

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Conflict-based search for optimal multi-agent pathfinding

Guni Sharon [a,*], Roni Stern [a], Ariel Felner [a], Nathan R. Sturtevant [b]

[a] Information Systems Engineering, Ben Gurion University, Be'er Sheba, 85104, Israel
[b] Department of Computer Science, University of Denver, Denver, CO, USA

ARTICLE INFO

Article history:
Received 16 September 2013
Received in revised form 23 November 2014
Accepted 25 November 2014
Available online 2 December 2014

Keywords:
Heuristic search
Multi-agent
Pathfinding

ABSTRACT

In the *multi-agent pathfinding* problem (MAPF) we are given a set of agents each with respective start and goal positions. The task is to find paths for all agents while avoiding collisions. Most previous work on solving this problem optimally has treated the individual agents as a single 'joint agent' and then applied single-agent search variants of the A* algorithm.

In this paper we present the Conflict Based Search (CBS) a new optimal multi-agent pathfinding algorithm. CBS is a two-level algorithm that does not convert the problem into the single 'joint agent' model. At the high level, a search is performed on a *Conflict Tree* (CT) which is a tree based on conflicts between individual agents. Each node in the CT represents a set of constraints on the motion of the agents. At the low level, fast single-agent searches are performed to satisfy the constraints imposed by the high level CT node. In many cases this two-level formulation enables CBS to examine fewer states than A* while still maintaining optimality. We analyze CBS and show its benefits and drawbacks.

Additionally we present the *Meta-Agent CBS* (MA-CBS) algorithm. MA-CBS is a generalization of CBS. Unlike basic CBS, MA-CBS is not restricted to single-agent searches at the low level. Instead, MA-CBS allows agents to be merged into small groups of joint agents. This mitigates some of the drawbacks of basic CBS and further improves performance. In fact, MA-CBS is a framework that can be built on top of any optimal and complete MAPF solver in order to enhance its performance. Experimental results on various problems show a speedup of up to an order of magnitude over previous approaches.

1. Introduction

Single-agent pathfinding is the problem of finding a path between two vertices in a graph. It is a fundamental and important problem in AI that has been researched extensively, as this problem can be found in GPS navigation [49], robot routing [8,3], planning [6,23], network routing [7], and many combinatorial problems (e.g., puzzles) as well [28,27]. Solving pathfinding problems optimally is commonly done with search algorithms based on the A* algorithm [22]. Such algorithms perform a best-first search that is guided by $f(n) = g(n) + h(n)$, where $g(n)$ is the cost of the shortest known path from the start state to state n and $h(n)$ is a heuristic function estimating the cost from n to the nearest goal state. If the heuristic function h is *admissible*, meaning that it never overestimates the shortest path from n to the goal, then A* (and other algorithms that are guided by the same cost function) are guaranteed to find an optimal path from the start state to a goal state, if one exists [11].

* Corresponding author.
E-mail addresses: gunisharon@gmail.com (G. Sharon), roni.stern@gmail.com (R. Stern), felner@bgu.ac.il (A. Felner), sturtevant@cs.du.edu (N.R. Sturtevant).

http://dx.doi.org/10.1016/j.artint.2014.11.006
0004-3702/

The *multi-agent pathfinding* (MAPF) problem is a generalization of the single-agent pathfinding problem for $k > 1$ agents. It consists of a graph and a number of agents. For each agent, a unique start state and a unique goal state are given, and the task is to find paths for all agents from their start states to their goal states, under the constraint that agents cannot collide during their movements. In many cases there is an additional goal of minimizing a cumulative cost function such as the sum of the time steps required for every agent to reach its goal. MAPF has practical applications in video games, traffic control [43,12], robotics [1] and aviation [31].

Algorithms for solving MAPF can be divided into two classes: *optimal* and *sub-optimal* solvers. Finding an optimal solution for the MAPF problem is NP-hard [56], as the state space grows exponentially with the number of agents. Sub-optimal solvers are usually used when the number of agents is large. In such cases, the aim is to quickly find a path for the different agents, and it is often intractable to guarantee that a given solution is optimal.

The problem addressed in this paper is to find an *optimal solution* to the MAPF problem. Optimal solvers are usually applied when the number of agents is relatively small and the task is to find an optimal, minimal-cost solution. This can be formalized as a global, single-agent search problem. Therefore, the traditional approach for solving MAPF optimally is by using A*-based searches [35,45]. A node in the search tree consists of the set of locations for all the agents at time t. The start state and goal state consist of the initial and goal locations of the different agents, respectively. Given a graph with branching factor b, there are $O(b)$ possible moves for any single agent and thus the branching factor for an A* search is $O(b^k)$ which is exponential in the number of agents. Naturally, search algorithms that are based on A* can solve this problem optimally, but they may run for a very long time or exhaust the available memory.

The first part of the paper gives a survey on MAPF research. We classify all existing work to two main categories, optimal and sub-optimal. We then further classify the different approaches for solving this problem sub-optimally. This is done through a consistent terminology that helps these classifications. In the second part of the paper we introduce a new approach for optimally solving MAPF. First, we present a novel *conflict-based formalization* for MAPF and a corresponding new algorithm called Conflict Based Search (CBS). CBS is a two-level algorithm, divided into high-level and low-level searches. The agents are initialized with default paths, which may contain conflicts. The *high-level search* is performed in a *constraint tree* (CT) whose nodes contain time and location constraints for a single agent. At each node in the CT, a *low-level search* is performed for all agents. The low-level search returns single-agent paths that are consistent with the set of constraints given at any CT node. If, after running the low level, there are still *conflicts* between agents, i.e. two or more agents are located in the same location at the same time, the associated high-level node is declared a non-goal node and the high-level search continues by adding more nodes with constraints that resolve the new conflict.

We study the behavior of our CBS algorithm and discuss its advantages and drawbacks when compared to A*-based approaches as well as other approaches. Based on characteristics of the problem, we show cases where CBS will be significantly more efficient than the previous approaches. We also discuss the limitations of CBS and show circumstances where CBS is inferior to the A*-based approaches. Experimental results are provided which support our theoretical understandings. While CBS is ineffective in some cases, there are many cases where CBS outperforms EPEA* [15,20], the state-of-the-art A*-based approach for this problem. Specifically, we experimented on open grids as well as on a number of benchmark game maps from Sturtevant's pathfinding database [47]. Results show the superiority of CBS over the A*-based approaches and ICTS [42], another recent approach, on many of these domains.

Next, we mitigate the worst-case performance of CBS by generalizing CBS into a new algorithm called Meta-agent CBS (MA-CBS). In MA-CBS the number of conflicts allowed between any pair of agents is bounded by a predefined parameter B. When the number of conflicts exceeds B, the conflicting agents are merged into a *meta-agent* and then treated as a joint composite agent by the low-level solver. In the low-level search, MA-CBS uses any possible complete MAPF solver to find paths for the meta-agent. Thus, MA-CBS can be viewed as a solving framework where low-level solvers are plugged in. Different *merge policies* give rise to different special cases. The original CBS algorithm corresponds to the extreme case where $B = \infty$ (never merge agents), and the Independence Detection (ID) framework [45] is the other extreme case where $B = 0$ (always merge agents when conflicts occur). Finally, we present experimental results for MA-CBS that show the superiority of MA-CBS over the other approaches on all domains. In addition, in Appendix A we introduce a variant of CBS which has memory requirements that are linear in the depth of the CT.

Preliminary versions of this research have appeared previously [38,39]. This paper contains a more comprehensive description of the CBS algorithm and the MA-CBS framework, with broader theoretical analysis and experimental comparisons to other MAPF algorithms.

2. Problem definition and terminology

Many variants of the MAPF problem exist. We now define the problem and later describe the algorithms in the context of a general, commonly used variant of the problem [45,46,42,38,39]. This variant is as general as possible to allow CBS to be applicable and it includes many sub-variants. Then, in Section 6.1 we describe the specific sub-variant used in our experimental setting. Finally, in Section 7 we briefly discuss how our new algorithm can be modified to other MAPF variants.

2.1. Problem input

The *input* to the *multi-agent pathfinding problem* (MAPF) is: