

# Divergence -Based Investigation for Optimal Agent Pathfinding

**Bhagyashri Abhang and Balaji Kamble**

Samarth Group of Institution College of Engineering, Belhe, Maharashtra, India

**Abstract:** *We are assumed a customary of mediators in the multi-agent pathfinding problem (MAPF), each of which has its own start and goal positions. The objective is to discovery paths for each agent without causing any collisions. The majority of previous work on the optimal solution to this issue has useful single-agent exploration variations of the A\* algorithm and treated each agent as a single "joint agent." The Conflict Based Search (CBS) optimum multi-agent pathfinding algorithm is the subject of this paper. CBS has two layer algorithm that does not use a single "joint agent" model to solve the problem. A Conflict Tree (CT), which is a tree based on conflicts between individual agents, is the high-level search engine. In the CT, each node represents a set of constraints on the agents' motion. To meet the high-level CT node's constraints, quick single-agent searches are carried out at the low level. CBS can frequently examine fewer states than A\* while still maintaining optimality thanks to this two-level formulation. We discuss CBS's advantages and disadvantages.*

**Keywords:** Pathfinding Problem

## I. INTRODUCTION

The problem of finding a path between two vertices in a graph is called single-agent pathfinding. As this problem can be found in GPS navigation [49], robot routing [8,3], planning [6,23], network routing [7], and numerous combinatorial problems (such as puzzles) as well [28,27], it is a fundamental and important AI problem that has received extensive research. Search algorithms based on the A\* algorithm are commonly used to solve optimal pathfinding problems [22]. A best-first search is guided by  $f(n) = g(n) + h(n)$ , where  $h(n)$  is a heuristic function that estimates the cost from  $n$  to the closest goal state.  $g(n)$  is the cost of the shortest known path from the start state to state  $n$ . The multi-agent pathfinding (MAPF) problem is a generalization of the single-agent pathfinding problem for  $k > 1$  agents. If the heuristic function  $h$  is admissible, which means that it never overestimates the shortest path from  $n$  to the goal, then A\* and other algorithms 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]. A graph and a number of agents make up this. The task is to find paths for all agents from their start states to their goal states with the constraint that agents cannot collide during their movements. Each agent has a distinct goal state and start state. Minimizing a cumulative cost function, such as the sum of the time steps required for each agent to achieve its goal, is frequently an additional objective. MAPF has practical applications in traffic control, robotics, aviation, video

games, and traffic control [43, 12], among other fields.

Our CBS algorithm's behavior is studied, and its advantages and disadvantages are discussed in relation to A\*-based and other methods. We demonstrate situations in which CBS will be significantly more effective than previous methods based on the characteristics of the problem. In addition, we demonstrate instances in which A\*-based approaches outperform CBS and discuss the limitations of CBS. Our theoretical understandings are bolstered by the experimental findings. While EPEA\* [15,20], the most advanced A\*-based approach for this problem, performs better in many instances than CBS, CBS is ineffective in some cases. We specifically experimented with open grids and several benchmark game maps from Sturtevant's pathfinding database [47]. On many of these domains, the results demonstrate that CBS outperforms both the A\*-based approaches and the more recent ICTS [42].

This study's preliminary versions have been published previously [38,39]. With a broader theoretical analysis and experimental comparisons to other MAPF algorithms, this paper provides a more in-depth description of the CBS algorithm and the MA-CBS framework.

## II. PROBLEM DEFINITION AND TERMINOLOGY

The MAPF problem can take many different forms. In the context of a general, widely used variant of the problem, we now define the issue and then describe the algorithms [45,46,42,38,39]. In order to make CBS applicable, this variant is as broad as possible and contains numerous subvariants. The specific sub-variant utilized in our experimental setting is then described in Section 6.1. In conclusion, we provide a brief explanation of how our new algorithm can be applied to other MAPF variants.

### 2.1 Problem Input

The multi-agent pathfinding problem (MAPF) takes the following as its input:

Each agent has the ability to take either a wait action to remain idle at its current vertex or a move action between successive time points. The possibility of a chain of agents following each other in a given time step can be dealt with in a number of different ways. This may not be permitted, may only be permitted if the first agent of the chain moves to an unoccupied location, or may be permitted even in a cyclic chain with no empty locations. All of these variations can use our algorithm.

### 2.2 MAPF Constraints

The main restriction of MAPF is that only one agent can occupy each vertex at a time. Additionally, there may be a constraint that prevents multiple agents from traversing the

same edge at the same time. A conflict occurs when a constraint is broken.

### **2.3 MAPF Task**

A set of non-conflicting paths, one for each agent, is one way to solve the MAPF problem. A path for agent  $a_i$  is a series of "move, wait" actions that, if it starts at  $start_i$ , will get it to  $goal_i$ .

There are two types of distributed versus centralized MAPF problems: centralized and distributed. In a distributed environment, each agent has its own computing power, and various communication paradigms (such as message passing, broadcasting, etc.) can be assumed. The distributed setting has been the subject of a lot of research [18,21,2]. In contrast, in a centralized setting, all agents must be solved by a single central computing power. In the same way, the centralized setting also includes the situation in which each agent has its own CPU, but full knowledge sharing is assumed and all agents are controlled by a centralized problem solver. This paper only addresses centralized approaches, of which many are discussed in the following section.

### **2.4 Maps from DAO**

We also experimented with three benchmark maps from Dragon Age: Sources [47]. Our goal was to demonstrate the evaluated algorithms' overall performance in this section. In contrast to the previous results, we perform ID on top of each of the evaluated algorithms in this case. A problem instance with a predetermined number of  $k$  agents and uniformly randomized start and goal locations is presented to the various algorithms. ID executes the associated algorithm on each subproblem separately after breaking these problems down into subproblems. We only present results for the three most effective algorithms: CBS, ICTS, and EPEA\*

1. The best solvers are [45,15,42,38,39]. Search-based solvers that are suboptimal [43,12] Procedure-based solvers that are not optimal [9,30,25,37,10] Hybrid solvers that aren't optimal [52,24,53,35] The CBS algorithm was then introduced. CBS is a revolutionary optimal MAPF solver. CBS is unique in that it produces optimal solutions despite carrying out all low-level searches using a single agent. The problem's structure determines CBS's performance. We have shown examples of bottlenecks (Fig.1, Section 2.7) where CBS does well and where there are open spaces (Fig.6, Section 5.3.2), CBS's poor performance. We looked at these cases and talked about how they affect CBS's performance.

The MA-CBS framework, which is a generalization of the CBS algorithm, was the next topic of discussion. Any MAPF solver that serves as a low-level solver can be used on top of MA-CBS. In addition, the Independence Detection (ID) framework developed by Standley [45] can be seen as an extension of MA-CBS.

MA-CBS connects CBS and other optimal MAPF solvers, such as A\*, A\* OD [45] and EPEA\* [15], via a bridge. It begins as a standard CBS solver in which a single agent performs

the low-level search at a time. If MA-CBS determines that two agents frequently engage in conflict, it groups them together. The given MAPF solver, such as A\*, is utilized by the low-level solver, which views this group as a single composite agent and seeks solutions for that group. Consequently, MA-CBS is adaptable and can select when to group agents together to reap the advantages of both CBS and conventional optimal solvers. We introduced the conflict bound parameter B as a straightforward yet efficient method for deciding when to group agents. The tendency of MA-CBS to create large groups of agents and solve them as a single unit is represented by the B parameter. When B is equal to CBS and MA-CBS converges to ID, respectively. MA-CBS can converge to a single meta-agent problem that includes all or most of the conflicting agents with the flexibility provided by setting 0 B. This allows MA-CBS to behave like CBS in situations where there are few conflicts, while MA-CBS can behave like CBS in situations where conflicts are numerous. Testbed map problems' experimental results back up our theoretical claims. The presented domains have varying rates of bottlenecks and open spaces. When corridors and bottlenecks dominate, MA-CBS with a high B value (100, 500) performs better than other algorithms. Additionally, experiments demonstrated that MA-CBS outperforms CBS and other cutting-edge MAPF algorithms when B has non-extreme values (i.e., neither B 0 nor B). The findings lead to the conclusion that, in the majority of cases, group agents benefit the most. When compared to CBS's method of never grouping agents and all previous optimal solvers' methods of never grouping agents, this results in a faster solving process.

## REFERENCES

- [1]. ZahyBnaya,RoniStern,ArielFelner,RoieZivan,StevenOkamoto,Multi-agentpathfindingforselfinterestedagents,in:SymposiumonCombinatorialSearch(SOCS),2013.
- [2]. BlaiBonet,HéctorGeffner,Planningasheuristicsearch,Artif.Intell.129(1)(2001)5–33.
- [3]. JoshBroch,DavidA.Maltz,DavidB.Johnson,Yih-ChunHu,JorjetaJetcheva,Aperformancecomparisonofmulti-hopwirelessadhocnetworkroutingprotocols,in:ProceedingsoftheInternationalConferenceonMobileComputingandNetworking,ACM,1998,pp.85–97.
- [4]. PaulSpirakis,DanielKornhauser,GaryMiller,Coordinatingpebblemotionongraphs,thediameterofpermutationgroups,andapplications,in:SymposiumonFoundationsofComputerScience,IEEE,1984,pp.241–250.
- [5]. RinaDechter,JudeaPearl,Generalizedbest-firstsearchstrategiesandtheoptimalityofA\*,J.ACM32(3)(1985)505–536.
- [6]. Esra Erdem, Doga G. Kisa, Umut Oztok, Peter Schueller, A general formal framework for pathfinding problems with multiple agents,in:AAAI,2013.
- [7]. MichaelErdmann,TomasLozano-Perez,Onmultiplemovingobjects,Algorithmica2(1–4)(1987)477–521.



Impact Factor: 5.731

## International Journal of Emerging Technologies and Innovative Research (IJETIR)

Volume 4, Issue 2, February 2024

- [8]. R. Ayeswarya and N. Amutha Prabha, “Fractional wavelet transform based OFDM system with cancellation of ICI,” *Journal of Ambient Intelligent Humanized Computing*, 2019.
- [9]. S. Tripathi, A. Rastogi, K. Sachdeva, M. K. Sharma and P. Sharma, “PAPR reduction in OFDM system using DWT with nonlinear high-power amplifier,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol.2, no.5, pp. 2278–3075, 2013.