Shaar

**Multi agent path finding**

Background:

Multi-Agent Pathfinding (MAPF) is a complex problem that arises in various domains, such as robotics, transportation, and video games. It involves the task of determining collision-free paths for multiple agents moving in a shared environment while satisfying various constraints and objectives. These agents can be physical entities like robots or virtual entities like characters in a game. In recent years, there has been a significant surge in interest in MAPF due to its practical applications in autonomous systems and the need for efficient coordination among multiple agents.

Problem Statement:

The primary objective of MAPF is to find a set of collision-free paths for a group of agents so that they can reach their respective goals without colliding with each other or violating any constraints.

To achieve this objective, several key aspects of MAPF need to be addressed:

a) Path Planning: One of the critical challenges in MAPF is determining feasible paths for each agent to navigate from their initial positions to their respective goal locations while avoiding collisions with other agents and obstacles present in the environment.

b) Conflict Resolution: Conflicts can arise when agents' paths intersect or when there are conflicting goals or constraints. Resolving these conflicts is essential to ensure smooth and collision-free movement of the agents.

c) Optimality: Finding optimal or near-optimal solutions is another crucial aspect of MAPF. Optimization can involve minimizing various cost metrics such as total travel time, energy consumption, or the number of agents involved in conflicts. Obtaining such optimal or near-optimal solutions can greatly improve the efficiency and effectiveness of multi-agent systems.

d) Scalability: The ability to handle large-scale MAPF instances with numerous agents and complex environments is crucial. Developing efficient algorithms that can address scalability challenges is necessary to make MAPF practical and applicable in real-world scenarios.

Existing Approaches:

Various approaches have been proposed to tackle the MAPF problem. These approaches can be broadly classified into centralized and decentralized methods.

Centralized approaches formulate MAPF as a single optimization problem and aim to find a joint plan for all agents simultaneously. These methods often provide global optimality guarantees, but they can face challenges when it comes to scalability. As the number of agents or the complexity of the environment increases, the computational requirements of centralized approaches can suffer from scalability issues.

Decentralized approaches, on the other hand, decompose the MAPF problem into individual agent subproblems. Each agent plans its path independently while considering the presence of other agents. These methods scale well and can handle larger instances of MAPF efficiently. However, decentralized approaches may result in suboptimal solutions due to the lack of global coordination among agents.

**Conflict Based Search**

Background

Conflict-Based Search (CBS) is a widely used and effective algorithm for solving Multi-Agent Pathfinding (MAPF) problems. MAPF involves determining collision-free paths for multiple agents moving in a shared environment while satisfying various constraints and objectives. CBS addresses the challenges of coordinating multiple agents by employing conflict resolution techniques to find optimal or near-optimal solutions. CBS has gained significant attention in recent years due to its ability to handle large-scale MAPF instances with efficiency and scalability.

CBS Approach

Conflict-Based Search adopts a two-phase process: the search phase and the conflict resolution phase.

a) Search Phase: In this phase, Conflict-Based Search (CBS) involves the utilization of a search algorithm, such as A\* (A-star) or Dijkstra's algorithm, to explore the space of possible paths for the agents. CBS treats each agent separately, enabling the search algorithm to generate a set of individual paths for each agent.

b) Conflict Resolution Phase: Once the search phase is complete, conflicts between agents are identified based on their intersecting paths or shared goals. CBS then applies conflict resolution techniques to modify the individual paths, ensuring that conflicts are resolved and the agents can reach their goals without colliding. Common conflict resolution methods include prioritizing agents, swapping paths, or performing path recomputation.

Key Features and Advantages of CBS

Conflict-Based Search offers several key features and advantages:

a) Optimality: CBS is capable of finding optimal solutions by iteratively resolving conflicts. Through the conflict resolution phase, the algorithm aims to minimize the total cost, such as travel time or energy consumption, or other relevant metrics.

b) Scalability: CBS is known for its scalability, allowing it to handle large-scale MAPF instances with numerous agents and complex environments. By decomposing the problem into individual agent subproblems, CBS can efficiently find solutions even in challenging scenarios.

c) Applicability: CBS has been successfully applied in various domains, including robotics, transportation, and game AI. Its versatility makes it suitable for a wide range of real-world applications that involve multi-agent coordination and path planning.

**Distributed Conflict Based Search**

Distributed Conflict-Based Search (DCBS) is an extension of Conflict-Based Search (CBS) that addresses the challenges of multi-agent coordination in a distributed setting. Multi-Agent Pathfinding (MAPF) involves finding collision-free paths for multiple agents in a shared environment while satisfying various constraints. DCBS aims to distribute the computational load and coordination efforts among agents, allowing for efficient and scalable solutions to MAPF problems.

DCBS Approach:

1. Search Phase: In this phase, similarly to normal CBS, a path is calculated for each agent. The main difference is that each agent calculates the path independently.
2. Conflict Resolution Phase: The conflict resolution phase in Distributed Conflict-Based Search (DCBS) aims to achieve a conflict-free solution, similarly to CBS. However, in DCBS, conflicts are handled through communication and collaboration among agents. Each agent takes responsibility for resolving its individual conflicts by interacting and exchanging information with other agents.

**DCBS - pseudo code:**

1. Each agent plan individually - broadcast all
2. After getting all plans:

If Valid plans - broadcast all

else if agent has no conflicts - broadcast all

else agent has conflict

**Constrain agent** to avoid the conflict

***and***

broadcast to **other agent** to avoid the conflict

Communication

Message passing is a crucial aspect of communication among agents in Distributed Conflict-Based Search (DCBS). Various types of messages are exchanged, each serving a specific purpose and conveying particular information.

Message types:

1. Path For Agent Message:

During the initial step of communication in Distributed Conflict-Based Search (DCBS), agents send this type of message containing their respective paths. These messages are broadcast from each agent to all other agents involved in the system.

Handle receiving message, by taking the path from the message and put the path in the initial solution. If the agent receives all the paths, he creates root CBS node.

1. Declare Solution Message

During the individual search, when a particular agent has reached a feasible solution, agents send message contains a feasible solution to the problem.

These messages are broadcast from each agent to all other agents involved in the system.

Handle receiving message, by checking if the message solution is better than his current solution. And update his current solution if necessary.

1. Declare Conflict Message

During the individual search, when a certain agent finds a conflict that it is involved in with another agent, the first agent sends a message containing the CBS node and the conflict to the second agent.

Handle receiving message, by creating a new CBS node with a constraint according to the conflict.

1. Declare Others Conflict Message

During the individual search, when a certain agent does not find a conflict in which it is involved, but the solution is not feasible, the agent sends a message containing CBS node.

These messages are broadcast from the agent to all other agents involved in the system.

Handle receiving message, by checking a CBS node. If the agent has no conflict, he ignores it. If he has conflict, he creates a new CBS node with a constraint according to the conflict.

Experiments

Results

Conclusions

Future research