Spectral Clustering in Rule-based Algorithms for Multi-agent Path Finding (Extended Abstract)

Irene Saccani[1], **Kristýna Janovská**[2], **Pavel Surynek**[2]
[1]University of Parma, Parma, Italy
[2]Czech Technical University in Prague, Faculty of Information Technology
Thákurova 9, 160 00 Praha 6, Czechia
irene.saccani@unipr.it, {janovkri, pavel.surynek}@fit.cvut.cz

Abstract

We address rule-based algorithms for multi-agent path finding (MAPF). MAPF is a task of finding non-conflicting paths connecting agents' initial and goal positions in a shared environment specified via an undirected graph. Rule-based algorithms use a fixed set of predefined primitive operations to move agents to their goal positions in a complete manner. We propose to apply spectral clustering on the underlying graph to decompose the graph into highly connected components and move agents to their goal cluster first before the rule-based algorithm is applied. The benefit of this approach is twofold: (1) the rule-based algorithms are often more efficient on highly connected clusters and (2) we can potentially run the algorithms in parallel on individual clusters.

Introduction

The goal of multi-agent path finding (MAPF) is to navigate a set of agents from their initial positions to their assigned destinations without any collision. This task is often modeled on an undirected graph $G = (V, E)$, where at most one agent can occupy a vertex at any given time, and agents can traverse edges but cannot swap positions across a single edge. In this work, we focus on rule-based algorithms for MAPF (Luna and Bekris 2011; Surynek 2009), where agents move in a graph (G) based on predefined primitive operations (rules). These rules often treat a configuration of agents on the vertices of G as a permutation in which the rules make local transformations. The advantages of this approach include its scalability for the number of agents and its computational speed. However, these algorithms do not provide optimal solutions with respect to common objectives such as *makespan* and *sum of costs*.

We propose two novel modifications based on the Push-and-Swap (Luna and Bekris 2011) and BiBOX (Surynek 2009) rule-based MAPF algorithms. We introduce a hierarchical approach where the first step involves decomposing the input graph into clusters of high connectivity and then their bi-connected components. This is achieved through spectral decomposition (Luo, Wilson, and Hancock 2003; Von Luxburg 2007) of the input graph G. These components then serve as input for the modified Push-and-Swap, which moves agents into their target component. Finally, BiBOX is run for every subset of agents inside each component.

Since we know that BiBOX is more efficient for densely populated and highly connected graphs, while Push-and-Swap works better for moving agents over large distances, we expect better solutions from this hierarchical approach.

Spectral Decomposition

We want to decompose the graph into several sub-graphs. A simple way to achieve this is by using **Spectral Clustering** (Luo, Wilson, and Hancock 2003; Von Luxburg 2007).

First, we calculate the optimal number of clusters. This can be done in several ways based on the graph's structure. In this paper, the number of clusters is computed using the *Eigengap Heuristic*. To do so, we find the normalized Laplacian L of the adjacency matrix of the input graph G. Then we compute the eigenvalues and eigenvectors of L and sort them according to the eigenvalues. The ideal number of clusters is given by the index of the maximal difference between one eigenvalue and the next. Having found the optimal number of clusters c, we create the matrix U where the columns are the first c eigenvectors and normalize the rows of U.

Finally, we use the rows to compute the clusters using a clustering algorithm such as K-means. The total time complexity of spectral clustering is $O(n^3) + O(cnT)$.

Then we create a new graph where every vertex represents a cluster and two clusters are linked if two of their vertices are neighbors; this graph is called *cluster graph*. Each cluster is then analyzed and divided into bi-connected components. A new graph is then built, in which each vertex represents a different bi-connected component, and an edge is added between two bi-connected components if there exists, in the original graph, an edge linking two vertices of those components. The search for bi-connected components in a cluster i is achieved using a depth-first search, with a complexity of $O(n_i)$ where n_i is the number of vertices in the cluster. Therefore, the search for biconnected components in all clusters has the total complexity of $O(n)$, where $n = |V|$ is the number of vertices in the original graph G. The search for the edges of the new graph has a complexity of $O(n^2)$. Hence, this part of the algorithm has a complexity of $O(n) + O(n^2)$.

The decomposition has been tested on *NetworkX* standard graphs and bi-connected random graphs. To ensure that even

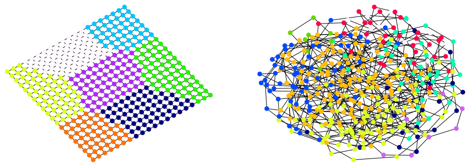

Figure 1: A 20x20 grid graph and a random bi-connected graph are colored based on spectral clustering.

highly connected graphs have enough clusters for the decomposition to be used for the subsequent use of the algorithm, a minimum value for the number of clusters must be imposed. In figure 1 a grid graph and a random bi-connected graph are colored based on the cluster of each vertex obtained from the spectral decomposition.

ChiBox and Chi-Push-and-Swap

We propose a modification that builds on existing MAPF algorithms BiBOX and Push-and-Swap introducing a hierarchical algorithm combining them. First, a graph decomposition takes place which results in clusters of high connectivity.

The first part is the modified Push-and-Swap algorithm called Chi-Push-and-Swap. The input graph of the algorithm is a component graph G_C, an abstraction of G. Each vertex in G_C represents one bi-connected sub-graph of G and vertices of G_C are linked if any two vertices of their respective clusters are connected in G.

The goal of Chi-Push-and-Swap is to move agents between these bi-connected clusters. The modification acts in several ways. First, when an agent reaches its goal vertex in the original version, the resolve operation takes place. This solves a situation where agents that are already at their goal positions are moved out by the agent. As goal positions within the goal cluster are not given, this operation is unnecessary. Specific goal positions within the goal cluster are then reached via a modification of BiBOX, which we call Chi-BOX.

Individual clusters are connected by at least one vertex of each respective sub-graph, called the *entrance vertex*. If a sub-graph is connected to more than one other sub-graph, it will always have more than one entrance vertex. These vertices have to be unoccupied to allow the passing of agents between different clusters. It is always possible to have at least two free vertices in each cluster, as it is a precondition for BiBOX. To maintain the entrance vertex free, we propose a new recursive operation called Component-Push.

Component-Push pushes agents in a cluster further into this sub-graph so that an agent occupying the entrance vertex may free it while staying in its goal sub-graph. All of the agents consider their neighboring vertex laying in the opposite direction from the entrance vertex as their goal position, so in this operation, agents move by one vertex. No other agents enter the sub-graph during this operation.

The result of Chi-Push-and-Swap algorithm is a set of sub-graphs (clusters) where agents have their goals within the same sub-graph as they are currently located. However, the agents are still not in their goal positions.

Multiple instances of BiBOX are then run depending on the number of clusters resulting from spectral decomposition to move agents to their goal positions.

This offers the possibility of parallelization of BiBOX, as the instances within individual clusters are disjoint. Due to the $O(n^3)$ complexity of the BiBOX, it turns out better to run multiple instances of BiBOX on smaller sub-graphs of G than to use one instance for the entire input graph G.

Conclusion

We propose a novel approach for rule-based algorithms for multi-agent path finding, consisting of three steps. Firstly, spectral decomposition is applied to partition the input graph into highly connected clusters through spectral clustering, a numerical technique based on the calculation of eigenvalues of the Laplacian matrix of the input graph. These highly connected clusters are subsequently divided into their bi-connected components.The agents are then moved to their target components using the Chi-Push-and-Swap algorithm, a modification of the existing Push-and-Swap algorithm (Luna and Bekris 2011). Chi-Push-and-Swap works with an abstract graph where each vertex represents a sub-graph (component) of the input graph. Agents are moved to their goal sub-graph but not necessarily to their final goal vertex.The final part is a variant of the BiBOX (Surynek 2009) algorithm called ChiBOX, which allows the parallel computation of disjoint instances of BiBOX on individual sub-graphs to finalize the movement of agents to their goal positions.Preliminary experiments have shown that this new approach can produce solutions with fewer agent movements compared to directly applying the rule-based algorithm to the unprocessed graph. The reduction in the number of agent movements is reflected in improved makespan or sum-of-costs.For future research, our focus will be on fine-tuning the spectral clustering method by adjusting its parameters to generate clusters better suited for specific rule-based algorithms.

Acknowledgements

This work has been supported by GAČR - the Czech Science Foundation, grant registration number 22-31346S.

References

Luna, R.; and Bekris, K. E. 2011. Push and Swap: Fast Cooperative Path-Finding with Completeness Guarantees. In *International Joint Conference on Artificial Intelligence*.

Luo, B.; Wilson, R. C.; and Hancock, E. R. 2003. Spectral Clustering of Graphs. In *Graph Based Representations in Pattern Recognition, 4th IAPR Workshop, GbRPR 2003, Proceedings*, volume 2726 of *LNCS*, 190–201. Springer.

Surynek, P. 2009. A novel approach to path planning for multiple robots in bi-connected graphs. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009*, 3613–3619. IEEE.

Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and computing*, 17: 395–416.