Proceedings of the Sixteenth International Symposium on Combinatorial Search (SoCS 2023)

Multi-Agent Path-Finding and Algorithmic Graph Theory (Student Abstract)

David Fairbairn[1, 2]
[1]Durham University
Department of Mathematical Sciences
Upper Mountjoy Campus, Stockton Rd, Durham DH1 3LE, United Kingdom
[2]Tharsus Limited
Coniston Road, Blyth, Northumberland NE24 4RF, United Kingdom
david.l.fairbairn@durham.ac.uk

Introduction

I specialise in conducting research in Multi-Agent Path-Finding (MAPF) and Algorithmic Graph Theory. Specifically, I investigate the impact of geometric constraints on a given instance of MAPF, as well as the expansion of MAPF to include resource constraints, target assignment path-finding (TAPF), and academic problems that are relevant to industry. In Algorithmic Graph Theory, I extend the capabilities of standard and novel MAPF solvers to temporal graphs, and explore clustering techniques and ideas that utilise Graph Classifications on MAPF domains to reduce the computational complexity of MAPF. Furthermore, I research the implementation and application of massively parallelised computing techniques on MAPF, especially in relation to distance matrix computation and parallelised centralised MAPF.

Aside from my PhD research, I have the pleasure to collaborate with researchers at Tharsus Limited, to directly apply my research to novel industrial problems and develop benchmarks that are relevant to both the industry and the research community for Multi-Agent Systems.

Multi-Agent Path-Finding

Multi-Agent Path-Finding (MAPF) (Stern et al. 2019) is a computational problem that involves finding paths for multiple agents to reach their respective destinations without colliding with each other. MAPF has many real-world applications, such as in robotics, logistics, and traffic management.

Geometric Constraints on MAPF Domains

The underlying geometry of a MAPF problem plays a crucial role in determining the complexity of a problem. Making use of the geometric invariants of a domain can produce quick and scalable algorithms on certain domains. I have co-authored a research paper (Atzmon et al. 2023) published at ICAPS 2023 that introduces an algorithm which exploits only the invariant distances between agents start and goal locations on a MAPF problem. I will be extending this research over the coming years to include further results including an extension to the algorithm to include invariants on subgraphs, connectivity and the use of the algorithm in a parallelised setting.

Additional Resource Constraints on MAPF Domains

MAPF domains in the real-world typically have additional resource constraints. In robotics problems this could be things such as limited battery life, limited capacity, time constraints or a restricted area for which a limit on the number of agents can traverse at once. These factors add to the complexity of the problem by adding additional constraints that need to be satisfied. I actively collaborate with industry partners to translate these real-world problems into academic problems and develop novel algorithms to solve them. Our initial results within this area have been promising, and I will be formalising these results over the coming years.

Target Assignment Path Finding (TAPF)

Target Assignment Path-Finding (TAPF) (Ma and Koenig 2016) is a variant of MAPF where the agents have to be assigned specific targets to visit and these are not necessarily constrained to a single target per agent. Agents may be given a set of target locations as a team, and we simply require that each agent reaches one of the target locations. This problem has many applications in industry, such as in warehouse automation and autonomous vehicle routing, where many agents have the ability to complete the same task. As part of my PhD I am developing novel algorithms to solve this problem utilising network flow algorithms, ideas within graph connectivity and graph clustering. I will be exploring the applications of this problem in industry and developing benchmarks to test the scalability of these algorithms.

Algorithmic Graph Theory

Algorithmic Graph Theory (AGT) is the study of algorithms that operate on graphs. Graphs are mathematical structures that consist of vertices (nodes) representing an object and edges (arcs) that connect the vertices representing a relationship between these objects. AGT has numerous applications, such as in network analysis, data mining, computational biology, failure and fault analysis, search, and many more. Studying algorithms on general graphs can yield new insights into algorithms on specific graph classes, such as trees, planar graphs and grid graphs (which are nominally used in MAPF). Likewise, insights from MAPF can be applied to AGT to develop novel algorithms for general graph problems, such as problems on temporal graphs, problems in graph clustering and the k-disjoint paths problem.

Temporal Graphs

Temporal graphs are graphs, where the vertices and edges have specific time steps when they are present in the graph. They are used in modelling dynamic systems, such as computer networks, transportation networks, and biological networks. Application of standard algorithms on temporal graphs can lead to incorrect results, and hence, new algorithms need to be developed to handle temporal graphs. It is possible that ideas from MAPF can be applied to temporal graphs to develop novel algorithms for temporal graphs. My research looks at the application of MAPF algorithms to give solutions and complexity results for problems within temporal graphs for certain classes of graph.

Clustering Techniques for MAPF

Clustering is a technique that involves dividing a set of objects into groups (clusters) based on their similarities. Clustering techniques are widely used in data mining, machine learning, and network analysis. For example, these techniques can be applied to group agents within a MAPF problem into clusters. Once the agents are clustered, the problem can be solved independently for each cluster, which reduces the overall computational complexity. I am currently investigating fast algorithms for clustering within MAPF problems, which can then be used to solve the problem using parallelised computing techniques.

Massively Parallelised Computing

Massively parallelised computing (MPC) refers to the use of multiple processing units to solve computationally intensive problems. MPC has become increasingly popular in recent years, with the advent of high-performance computing (HPC) and cloud computing. As part of my PhD and in collaboration with industry partners, I am investigating the implementation of MPC techniques on problems within MAPF and AGT. In particular, I am investigating how NVIDIA's CUDA platform can be used to implement MPC techniques on these problems.

Distance Matrix Computation

Distance Matrix Computation is a critical problem in many applications, such as in network analysis, clustering, machine learning and MAPF. Distance matrix computation involves computing the pairwise distances between a set of points, which can be represented as a matrix. Knowing the distance matrix is important as it can be used to help guide the search for a solution to a MAPF problem. The computational complexity of distance matrix computation can quickly become prohibitive for large datasets. MPC can be used to speed up the computation by dividing the dataset into smaller subsets and computing the distances in parallel. My research looks to extend algorithms for application in MPC to compute the distance matrix for MAPF domains and those which are not necessary metric spaces.

Parallelised Centralised MAPF

Parallelised centralised MAPF refers to the use of multiple processing units to solve a single MAPF problem. Centralised MAPF involves a single entity that coordinates the movement of all the agents to their respective destinations. The computational complexity of centralised MAPF can quickly become prohibitive for large numbers of agents. MPC can be used to speed up the computation by dividing the problem into smaller sub-problems and solving them in parallel. I am combining my research within AGT and MAPF to develop novel clustering techniques for MAPF problems that can then be solved with no communication required between parallel solvers.

Conclusion

In conclusion, my research focuses on advancing the state-of-the-art in Multi-Agent Path-Finding (MAPF) and Algorithmic Graph Theory (AGT) to address academic and industry problems.

Collaborating with industry partners like Tharsus Limited allows me to directly apply my research findings to practical problems and establish benchmarks that benefit both the research community and industrial practitioners. Furthermore, massively parallelised computing techniques, specifically in distance matrix computation and parallelised centralised MAPF, aim to harness the power of parallel processing for efficiently solving large-scale MAPF and AGT problems.

I am excited about the multitude of research endeavors I am engaged in and the potential impact they can have in the field of Multi-Agent Path-Finding and Algorithmic Graph Theory. It would be a privilege to be part of the Doctoral Consortium at SOCS 2023, where I can further strengthen my knowledge, network with fellow researchers, and contribute to the ongoing advancements in the field.

Acknowledgements

This PhD project is funded by Tharsus Limited.

References

Atzmon, D.; Bernardini, S.; Fagnani, F.; and Fairbairn, D. 2023. Exploiting Geometric Constraints in Multi-Agent Pathfinding. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*. Prague, Czech Republic.

Ma, H.; and Koenig, S. 2016. Optimal Target Assignment and Path Finding for Teams of Agents. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 1144–1152.

Stern, R.; Sturtevant, N.; Felner, A.; Koenig, S.; Ma, H.; Walker, T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, S.; Boyarski, E.; and Bartak, R. 2019. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks [Position Paper]. In *Proceedings of the Symposium on Combinatorial Search (SoCS)*.