Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence

Tractable Massively Multi-Agent Pathfinding with Solution Quality and Completeness Guarantees

Ko-Hsin Cindy Wang
The Australian National University and NICTA*
cindy.wang@nicta.com.au

1 Introduction

Pathfinding is an important underlying task for autonomous agents such as mobile robots, computer game characters, and aircrafts on airport taxiways. Abstracting the environment into a navigation graph enables a mobile unit to use heuristic search, such as A*, to plan a path to its goal. When multiple units move simultaneously inside the shared space, the solution also involves navigating every unit to its target without collisions. In a fully known, static, two-dimensional environment, finding an optimal solution to a multi-agent problem is NP-hard [Ratner and Warmuth, 1986]. With both the branching factor and the number of states growing exponentially in the number of units, a centralised search in the combined state space of all units is intractable in practice even on relatively small collections of units. However, problems in applications such as robotics, logistics, military operations planning, disaster rescue, and massively multi-player online games often involve *massively* many agents. My thesis addresses scalability as well as providing tractability and completeness guarantees in multi-agent pathfinding.

Traditional multi-agent pathfinding approaches each have specific strengths. Centralised methods preserve solution optimality and completeness by planning globally. Decentralised methods decompose the problem into a series of smaller searches, which is often much faster, and scale up to much larger problems. However, each approach inherently trades off between optimality and scalability, or completeness and efficiency. For instance, the optimality requirement is very costly in practice. Incorporating decoupled planning for non-interfering subgroups of units, Standley's OD+ID [2010] scales better than centralised planning. But as reported, OD+ID does not solve as many units as the incomplete method, HCA* [Silver, 2005] on the same data set. On the other hand, decentralised methods such as HCA* and WHCA* [Silver, 2005] achieve significant scale-up and speed-up, but formal characterizations of their running time, memory requirements, and solution quality in the worst case are not known. Moreover, they cannot a priori answer whether a given instance can be successfully solved.

Some recent work have started to bridge the missing link between completeness and tractability. Ryan [2008] introduced a complete method which combined multi-robot path planning with hierarchical planning on search graphs with the specific substructures of stacks, halls, cliques and rings. BIBOX [Surynek, 2009] solves problems on bi-connected graphs that have at least 2 unoccupied vertices. Because BIBOX was designed for densely populated problems, it is better suited for scenarios such as automatic packages inside a warehouse, than computer games where there are a lot fewer units than locations on a map.

This work assumes a class of cooperative multi-agent pathfinding problems on undirected graphs that were discretized from fully known, 2-D environments containing static obstacles. Units are the same size and, like circular robots, have no turning constraints. Each unit has a distinct start and target position. A graph node can be occupied by exactly one unit at a time. Units move synchronously to the next unoccupied node per time step. Moving into an adjacent unoccupied node does not depend on other neighbouring nodes (unlike making diagonal moves on a grid map).

The term *massively* is used here to contrast the size of problems we examine, containing 100 to 2000 units, with what can be handled using optimal multi-agent pathfinding algorithms in practice (up to 60 units in Standley [2010]). We use realistic game grid maps[1] containing 13765 to 51586 nodes. On challenging scenarios with 2000 uniformly randomly generated units, our algorithm MAPP solves 92-99.7% of units.

2 Contributions To Date

My approach is to decompose the global search into subproblems of an offline path pre-computation, followed by plan execution with online conflict resolution. We have developed two algorithms in this framework: FAR [Wang and Botea, 2008] and MAPP [Wang and Botea, 2009].

Initially, aiming at improving computation speed and memory usage on large-scale problems, we introduced an efficient search graph structure inspired by real-life road networks, where lanes are strictly 1-way to avoid head-to-head collisions. Our flow annotation restricts movement on a grid map, allowing only one horizontal and vertical direction along each row and column, alternating between adjacent rows and columns. The FAR algorithm runs an indepen-

*NICTA is funded by the Australian Government's *Backing Australia's Ability* initiative.

[1] http://users.rsise.anu.edu.au/~cwang/gamemaps

dent A* search per unit on the flow-annotated search graph, then uses a heuristic procedure to break deadlocks online, repairing plans locally. Experimental results in Wang and Botea [2008] show that FAR plans faster, uses less memory, and can often scale up to more units compared with the recent successful grid map algorithm, WHCA* [Silver, 2006]. Even without diagonal moves, the average solution length ratio between WHCA* (with diagonals) and FAR is 86%.

While significantly improving speed and scalability with a decentralised approach, the inability to a priori determine whether a given problem instance can be solved by our algorithm is a serious drawback. In most real life applications, it is unacceptable to launch an algorithm without knowing whether it can return a solution, or will fail by either timing out or first using up all the computing resources. To combine the strengths of completeness guarantees, tractability, and scalability, our subsequent work focused on extracting information from features of the problem instance at hand, for designing an algorithm that identifies a tractable subclass of multi-agent pathfinding problems. The original SLIDEABLE class has three polynomial time verifiable conditions. 1) *alternate connectivity* exists for every consecutive triple locations along a path, i.e., an alternate path, Ω, connects the two ends without going through the middle; 2) a blank (unoccupied location) can be found in front of each unit in the initial state; 3) paths do not cross over other units' targets. These three conditions allow a blocked unit to attempt to bring a blank to its front from nearby, by sliding some blocking units along its Ω-path. This blank travelling operation ensures units can make progress on their pre-computed paths, and is at the heart of our MAPP algorithm. Although incomplete for the general case, Basic MAPP is guaranteed to solve units that fall into the SLIDEABLE class with time and length of solutions under low-polynomial bounds. For algorithm details and formal proofs, please refer to Wang and Botea [2009].

After implementing MAPP and integrating it in the HOG[2] framework, we evaluated its performance in practice, including scalability, completeness range, running time, and solution quality. The empirical studies, similar to FAR, were done on grid map problems, using the randomly generated data set from Wang and Botea [2008]. The input maps were 10 of the largest from the game Baldur's Gate, with different configurations of obstacles forming rooms, corridors, and narrow tunnels. We test each map with 100 to 2000 mobile units in increments of 100. Preliminary experiments identified Basic MAPP's key bottlenecks, based on which we extended its completeness range, and reduced travel distance by pruning unnecessary moves. Over the entire test data set, enhanced MAPP solved 98.82% of units, FAR solved 81.87%, while 77.84% and 80.87% are solved by WHCA* with and without diagonal moves allowed, respectively. MAPP is also competitive in speed [Wang and Botea, 2010]. We analyzed the quality of MAPP's solutions using multiple quality criteria: total travel distance, makespan, and sum of actions (including move and wait actions). We introduced offline and online enhancements that significantly reduced waiting and congestion, while maintaining MAPP's previous advantages on the performance criteria. On average, the sum of actions is cut to half. The improved MAPP becomes state-of-the-art in terms of solution quality, being competitive with FAR and WHCA*. Comparing the solutions of all 3 suboptimal algorithms to lower bounds of optimal values shows they have reasonable quality. For instance, MAPP's total travel distance is on average 19% longer than a lower bound on the optimal value.

[2] http://webdocs.cs.ualberta.ca/~nathanst/hog.html

3 Conclusions and Future Plans

Suboptimal multi-agent pathfinding algorithms scale well beyond the capabilities of optimal methods. The FAR algorithm traded optimality and completeness to improve efficiency, like many other approaches in the literature. Results demonstrated that FAR can be very fast and effective in many cases, however, it has shortcomings as with previous decentralised approaches, as mentioned earlier. MAPP, on the other hand, bridges the gap between scalability, tractability, and providing formal completeness guarantees. MAPP has a significantly better success ratio than FAR and WHCA*. In instances that all three algorithms can fully solve, MAPP is also better or at least as competitive in speed and solution quality, as these state-of-the-art decentralised algorithms.

In future work, we plan to continue to extend the MAPP algorithm. In particular, some initially non-SLIDEABLE units become solvable as other units are being solved. We will explore other possible optimizations, also investigate a measure of how tightly coupled units are in a large multi-agent pathfinding problem, and use it to refine our theoretical study and to design heuristic enhancements. In the long term, MAPP can be part of an algorithm portfolio, since we can cheaply detect when it is guaranteed to solve an instance.

References

[Ratner and Warmuth, 1986] D. Ratner and M. Warmuth. Finding a shortest solution for the $N \times N$ extension of the 15-puzzle is intractable. In *AAAI*, pages 168–172, 1986.

[Ryan, 2008] M. R. K. Ryan. Exploiting Subgraph Structure in Multi-Robot Path Planning. *JAIR*, 31:497–542, 2008.

[Silver, 2005] D. Silver. Cooperative Pathfinding. In *AIIDE*, pages 117–122, 2005.

[Silver, 2006] D. Silver. Cooperative pathfinding. *AI Programming Wisdom*, 3:99–111, 2006.

[Standley, 2010] Trevor Standley. Finding Optimal Solutions to Cooperative Pathfinding Problems. In *AAAI*, pages 173–178, 2010.

[Surynek, 2009] P. Surynek. A novel approach to path planning for multiple robots in bi-connected graphs. In *ICRA*, pages 3613–3619, 2009.

[Wang and Botea, 2008] K.-H. C. Wang and A. Botea. Fast and Memory-Efficient Multi-Agent Pathfinding. In *ICAPS*, pages 380–387, 2008.

[Wang and Botea, 2009] K.-H. C. Wang and A. Botea. Tractable Multi-Agent Path Planning on Grid Maps. In *IJCAI*, pages 1870–1875, 2009.

[Wang and Botea, 2010] K.-H. C. Wang and A. Botea. Scalable Multi-Agent Pathfinding on Grid Maps with Tractability and Completeness Guarantees. In *ECAI*, pages 977–978, 2010.