Massively Multi-Agent Pathfinding made Tractable, Efficient, and with Completeness Guarantees

(Extended Abstract)

Ko-Hsin Cindy Wang
The Australian National University / NICTA
Cindy.Wang@rsise.anu.edu.au

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*heuristic methods, plan execution, formation, and generation*

General Terms

Algorithms, performance

Keywords

mobile agents, multi-agent planning, agent cooperation

1. INTRODUCTION

Pathfinding is an important underlying task for many autonomous agents. Abstracting the environment into a navigation graph (e.g., a grid map) enables a mobile unit to plan its path to goal using heuristic search. For example, an A* search finds an optimal path. With multiple units moving simultaneously inside a shared space, the goal is to navigate each unit to its target without colliding into static obstacles or other units. This problem is much harder. Even without motion constraints, finding optimal solutions in a fully known, two-dimensional environment is NP-complete [1, 6]. With both branching factor and 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 mobile units. However, problems in applications such as robotics, logistics, military operations planning, disaster rescue, and computer games often involve 'massively' large numbers of agents.

Traditional multi-agent path planning approaches each has its particular strengths. Centralised methods preserve solution optimality and completeness by planning globally, and sharing information centrally. Decentralised methods decompose the problem into a series of smaller searches, which can be much faster, and scale up to much larger problems. However, each approach also has serious drawbacks. For instance, the optimality requirement is very costly in practice. [4] incorporates decentralised planning for non-interfering subgroups of units (ID) to an improved centralised planning (OD), and scales much better than a standard centralised A*. But as reported in the paper, the incomplete method, HCA* [3], was solving more units than OD+ID on the same data set. Furthermore, the problem instances used in [4] contain at least 2 orders of magnitude fewer agents than our experiments in [7, 9]. On the other hand, decentralised methods such as [3] trade off optimality and completeness for scalability and efficiency, but formal characterizations of their running time, memory requirements, and quality of solutions in the worst case are not known, and they lack the ability to answer a priori whether a given problem can be successfully solved.

To bridge a missing link between completeness and tractability, some recent work take a bounded suboptimal approach. [2] 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 [5] solves problems on bi-connected graphs that have at least 2 unoccupied vertices. But because of the high density of units in the test problems, BIBOX was only tested up to 400 nodes. In comparison, the Baldur's Gate game maps we use[1] contain 13765 to 51586 nodes.

My thesis addresses the important issues in multi-agent pathfinding hand in hand, by providing tractability and completeness guarantees, as well as being scalable and efficient. This work assumes a class of cooperative multi-agent pathfinding problems on undirected graphs that were discretized from fully known, 2-D workspaces containing static obstacles. Units are the same size, and like circular robots, have no turning constraints. Each unit has distinct start and target positions. 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 in the grid map setting).

The term *massively* is used here to contrast the scalability of our algorithms, FAR [7] and MAPP [8], with previous state-of-the-art algorithms. MAPP solves 92–99.7% of units on challenging scenarios with 2000 uniformly randomly generated units on realistic game grid maps, significantly more than previous algorithms that were experimented on problems of 1 to 2 magnitudes fewer units.

2. CONTRIBUTIONS TO DATE

My approach is to decompose the global search into an offline path pre-computation, followed by plan execution with online conflict resolution. We have developed two algorithms in this framework: FAR [7] and MAPP [8].

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

Cite as: Massively Multi-Agent Pathfinding made Tractable, Efficient, and with Completeness Guarantees (Extended Abstract), K.-H. C. Wang, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1343-1344.

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 one vertical direction along each row and column, and alternates between rows and columns. The FAR algorithm runs an independent A* search per unit on the flow-annotated search graph, then repairs plans locally and online, using a heuristic procedure to break deadlocks. Experimental results in [7] 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* [3]. Even without diagonal moves, the average solution length ratio between WHCA* (with diagonals) and FAR is 86%.

While achieving significant speed-up and scalability with this 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 (partial) completeness, tractability, and scalability, we extract information from features of the problem instance at hand to design 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* existence: for every consecutive triple locations along a path, an alternate path 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) targets are isolated from all other paths. These conditions allow a unit, blocked on its path to goal, to attempt to bring a blank to its front by sliding other units along an alternate path. This blank travelling operation enables units to make progress on their pre-computed paths, and is at the heart of our MAPP algorithm. Although incomplete for the general case, MAPP is guaranteed to solve units that fall into the SLIDEABLE class with time and solutions under low-polynomial bounds [8].

After implementing MAPP and integrating it in the HOG 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. Experiments were run on the data set of randomly generated instances used in [7]. The input maps were 10 of the largest from the game Baldur's Gate, with various configurations of obstacles forming rooms, corridors, and narrow tunnels. We test each map with 100 to 2000 mobile units in increments of 100. Preliminary results identified Basic MAPP's key bottlenecks, based on which we made extensions to enlarge its completeness range, plus improvements such as reducing unnecessary moves. Extended MAPP scales significantly better: with 2000 units, FAR solved as few as 17.5%, WHCA* solved 12.3% (with diagonal moves) and 16.7% (without), while MAPP solved at least 92%. Over the entire 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. A summary of these results is reported in [9].

We analyzed the quality of MAPP's solutions using multiple quality criteria such as 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 advantages on the previous 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.

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 for an improved efficiency, like many other approaches in the literature. Results demonstrated that FAR can be very effective in many cases. However, as with previous methods, FAR has shortcomings of incompleteness, and provides no criteria to distinguish between problems it can or cannot solve, nor guarantees with respect to the running time and the quality of its computed solutions. MAPP, on the other hand, bridges the gap between scalability and efficiency in practice with providing formal completeness guarantees. Providing these guarantees in multi-agent pathfinding does not pose as strong a limiting factor as the optimality requirement on problems that can be solved in practice. MAPP has even better scalability and success ratio than FAR and WHCA*. In instances that all 3 algorithms can fully solve, MAPP is also better or at least as competitive in speed and solution quality.

In future work, we plan to continue to extend the MAPP algorithm. In particular, some initially non-SLIDEABLE units could become solvable as other units are being solved. We will explore other possible optimizations, and also investigate a measure of how tightly coupled units are in a large multi-agent pathfinding problem, and to 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. Hence it is also worthwhile to find formal tractable subclasses of incomplete algorithms such as FAR.

4. REFERENCES

[1] 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.

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

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

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

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

[6] P. Surynek. An Optimization Variant of Multi-Robot Path Planning is Intractable. In *AAAI*, 2010.

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

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

[9] 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.