Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence

Solution Quality Improvements for Massively Multi-Agent Pathfinding

Ko-Hsin Cindy Wang and **Adi Botea** and **Philip Kilby**

NICTA* and The Australian National University
cindy.wang@nicta.com.au
http://users.cecs.anu.edu.au/~cwang

Introduction

In multi-agent pathfinding, MAPP (Wang and Botea 2009; 2010) has previously been shown to be state-of-the-art in terms of scalability and success ratio (i.e., percentage of solved units), on problems involving significantly larger numbers of mobile units than can be tractably handled using optimal algorithms. MAPP further provides a formal characterization of problems it can solve, and low-polynomial upper bounds on the resources required. However, until now, MAPP's solution quality had not been extensively analyzed.

In this work we empirically analyze 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 introduce enhancements that have shown significant improvements. On average, the sum of actions is cut to half. We maintain MAPP's advantages on different performance criteria, such as scalability, success ratio, complexity upper bounds, and ability to tell apriori if it will succeed in the instance at hand. The improved MAPP becomes state-of-the-art in terms of solution quality, being competitive with FAR (Wang and Botea 2008) and WHCA* (Silver 2005), two successful algorithms from the literature. On the other hand, FAR and WHCA* lack the ability to a priori decide whether they can solve an instance, and their upper bounds on resources required are not known.

Since finding optimal solutions is NP-complete (Surynek 2010; Ratner and Warmuth 1986), optimal algorithms have limited scalability. To evaluate the quality of the solutions provided by suboptimal algorithms, we compare their solutions to lower bounds of optimal values, which are cheap to compute. These lower bounds are obtained from counting moves only, ignoring wait actions. In the cases of total travel distance and sum of actions, we sum the shortest path from each start to target. For makespan, we take the number of moves in the longest path. Results show that MAPP's solutions have a reasonable quality. For instance, MAPP's total travel distance is on average 19% longer than an A* lower bound on the optimal value.

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

MAPP and Solution Quality Improvements

We provide a high-level description of the previous MAPP algorithm, focusing on features relevant to our work. More details can be found in Wang and Botea (2009; 2010).

At the start of the algorithm, a path is computed from the start to the target of each unit. These paths, called π-paths, must satisfy additional constraints defining the SLIDEABLE class (Wang and Botea 2009), or variations of SLIDEABLE with an extended completeness range (Wang and Botea 2010). MAPP is complete on these classes of problems.

A solution plan consists of alternating *progression* and *repositioning* stages. During progression, unsolved units advance along their π-paths, and attempt to push other units away to clear the way. At the end of each progression stage, at least one unit, and usually more, get solved. The remaining units could be pushed off their π-paths. Repositioning brings these units back on track, so they can reach their targets as planned. A repositioning stage is performed by undoing part of the moves in the previous progression stage. The next progression stage will solve at least one more unit, and the algorithm eventually terminates.

We identified causes that can affect the quality of solutions computed with the existing MAPP, and present the following enhancements that address such limitations.

- *Spreading out the paths*. In path computation, each π-path is planned independently. As units prefer shorter paths, π's can overlap, creating online traffic jams. The blocking and repositioning moves lengthen the overall solution, and also appear less aesthetic. Hence we encourage π-paths to avoid already busy locations, using a global traffic report. With a hash map implementation, the memory overhead is very reasonable. Furthermore, spreading out is an added (soft) constraint to existing versions of MAPP, an does not affect its completeness range or low-polynomial time and memory upper bounds.
- *Dynamic repositioning*. With progression and repositioning clearly separated, units that moved off their π-paths had to wait until the end of the current progression before attempting to get back, resulting in many wait actions. Hence, we allow some repositioning moves to interleave with progression moves of others, under well specified conditions. Global repositioning can still take place, but is significantly reduced. Dynamic repositioning preserves

	DR	SP	SP+DR
Repositioning stages	-76.3%	-21.2%	-82.1%
Total distance	-2.5%	-2.9%	-9.5%
Makespan	-47.7%	-18.9%	-56.7%
Sum of actions	-40.8%	-20.2%	-49.7%

Table 1: Average percentage reductions. Negative signs indicate improvements over old MAPP.

the memory, time, and solution length upper bounds of the existing MAPP, since these moves are a subset of push-aside moves performed in the current progression step.

Experiments

Experiments were run on data used in previous work (Wang and Botea 2010; 2008). The 10 largest maps, with 13765 to 51586 traversable tiles, are taken from the game Baldur's Gate. They contain non trivial configurations of obstacles. We test each map with 100 to 2000 units in increments of 100. For each number of units on each map, 10 instances were generated with random start and target locations. Unless stated otherwise, the maps are 4-connected grids.

In Table 1, the best existing MAPP (Wang and Botea 2010) is compared with our new versions. DR MAPP, using dynamic repositioning, is more effective in reducing waiting actions than spreading out paths (SP MAPP). Combining the two, SP+DR MAPP, produces even stronger results in all criteria. We compared SP+DR MAPP with two algorithms that scale well, FAR and WHCA*. We used Sturtevant and Buro's (2006) WHCA* extension with spatial abstraction (but no unit priority system for replanning), WHCA*(w, a). As the success percentages of the 3 algorithms are quite different (MAPP solved 98.8% of units, while FAR and WHCA* solved 81.9% and 80.9%, respectively), we compare results on the subset of instances fully solved by all algorithms. Figure 1 illustrates an average behaviour from one sample map. As shown, MAPP is competitive with FAR and WHCA*.

We also compared SP+DR MAPP's solutions with lower bounds of the theoretical optimal solution. "A* lowerbound" is computed with only straight cardinal moves. "A*+d lowerbound" has diagonal moves enabled. Notice that, by only counting moves and ignoring wait actions, the lower bounds of the sum of actions and the makespan are more optimistic than actual optimal values.

Conclusion

Suboptimal multi-agent pathfinding algorithms scale well beyond the capabilities of optimal methods. MAPP is state-of-the-art in scalability and success ratio, and provides formal guarantees. We introduced enhancements that reduce its collisions and waiting time very effectively. Results show that we have advanced MAPP significantly, bringing its solution quality to a state-of-the-art level. Combined with its existing strengths, MAPP is either better or at least as competitive as other *massively* multi-agent pathfinding algorithms on most major performance criteria.

In future, we plan to test different penalties in SP MAPP, and further improve makespan by pre-ordering units.

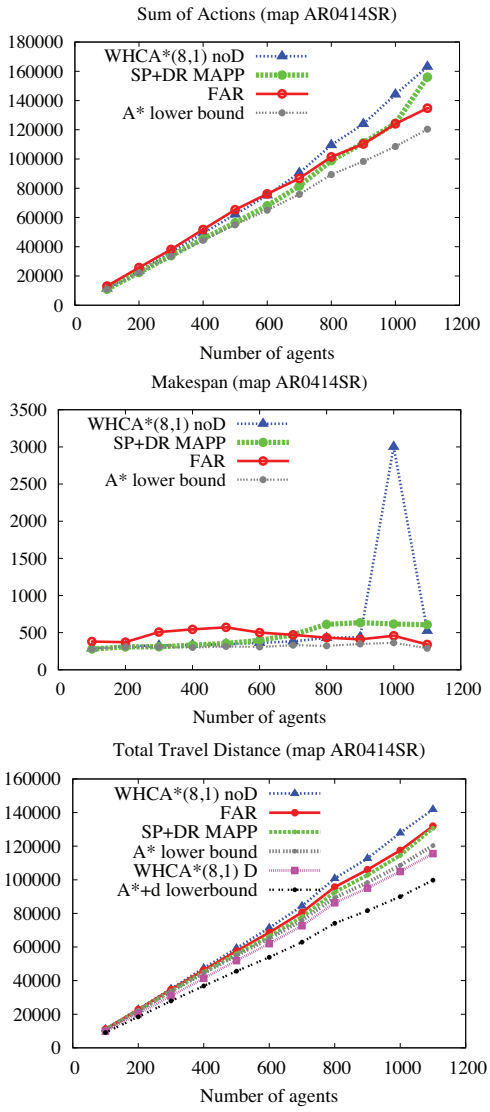

Figure 1: SP+DR MAPP's solution quality compared with FAR and WHCA* on map AR0414SR.

References

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

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

Sturtevant, N. R., and Buro, M. 2006. Improving collaborative pathfinding using map abstraction. In *AIIDE*, 80–85.

Surynek, P. 2010. An Optimization Variant of Multi-Robot Path Planning is Intractable. In *AAAI*, 1261–1263.

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

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

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