Two Techniques That Enhance the Performance of Multi-robot Prioritized Path Planning

2 authors, including:

Anton Andreychuk
Peoples' Friendship University of Russia
39 PUBLICATIONS   **189** CITATIONS

Two Techniques That Enhance the Performance of Multi-robot Prioritized Path Planning

Extended Abstract

Anton Andreychuk
Peoples' Friendship University of Russia
(RUDN University)
Moscow, Russia
andreychuk@mail.com

Konstantin Yakovlev
Federal Research Center "Computer Science and Control"
of Russian Academy of Sciences
National Research University Higher School of Economics
Moscow, Russia
yakovlev@isa.ru

ABSTRACT

We introduce and empirically evaluate two techniques aimed at enhancing the performance of multi-robot prioritized path planning. The first technique is the deterministic procedure for re-scheduling (as opposed to well-known approach based on random restarts), the second one is the heuristic procedure that modifies the search-space of the individual planner involved in the prioritized path finding.

KEYWORDS

Multi-robot systems; multi-robot path planning; multi-agent path finding; prioritized planning; random restarts

1 PROBLEM STATEMENT

Consider n open-disk robots of equal radii r intended to simultaneously move towards their goals on a grid comprised of blocked and unblocked cells. The task is to plan the set of feasible trajectories (one per each robot) such that each two trajectories for different robots are collision-free, e.g. robots following them never collide. The spatial component of the trajectory, e.g. the path, is the sequence of straight-line segments connecting the centers of grid cells. We assume that all robots follow their paths with identical speed and can start/stop instantaneously. Depending on the application domain the cost of the solution can be either *makespan*, i.e. the time by which the last robot reaches its goal, or *flowtime*, which is the sum of traversal times across all the robots involved in the instance.

2 METHOD

2.1 Related work

Complete (and optimal or bounded-suboptimal) solvers to the described problem exist, like the ones introduced in [13], [9], [8], [11], [5], [7] etc., but they typically require significant computational effort, especially when the number of robots increases. Appealing alternative is prioritized planning [6], which is widely used in robotics [4], [3] due to its simplicity and ability to scale well to large problems. Prioritized planners are incomplete in general but can be tweaked to increase the chance of finding the solution (and decrease flowtime and/or makespan). The most widespread approach to do so is to randomly re-assign priorities in case of failure and re-plan, see [1], [2] for details.

2.2 Contribution

In this work we introduce a novel deterministic technique for re-assigning the priorities (re-scheduling) in case planning with initial priorities fails. We also introduce the notion of safe-start-interval (*SSI*) in order to prohibit the individual planner, involved in prioritized planning, from interfering with the start positions of low-priority robots for certain amount of time. Both suggested techniques enhance the performance of planning (as shown experimentally) and can be embedded into any prioritized planner.

2.3 Deterministic re-scheduling

Prioritized planners are incremental in nature, e.g. they plan individual trajectories sequentially one by one in accordance with the imposed priority ordering. The planner may fail to solve an instance only by failing to find an individual trajectory for some robot in a sequence, which, in turn, happens only when the high-priority robots prevent it from doing so by their motion[1]. Thus, in case of failure, we suggest raising the priority of the failed robot to 1 (the highest priority) and re-planning. Obviously this robot will not be the cause of multi-agent planner's failure anymore (which is not true with random re-scheduling) and chances are the overall solution will be found. If no, deterministic re-scheduling process repeats until the planner succeeds or until it comes across the priority sequence that has already been investigated (in this case "failure" is reported).

2.4 Start-safe intervals

In conventional prioritized planning high priority robots are allowed to pass through the start locations of the low-priority ones at any moment of time. This might be undesirable as further on some robot may fail to plan its trajectory due to the fact it's immediately knocked off by the high-priority robot and there is no room for

---

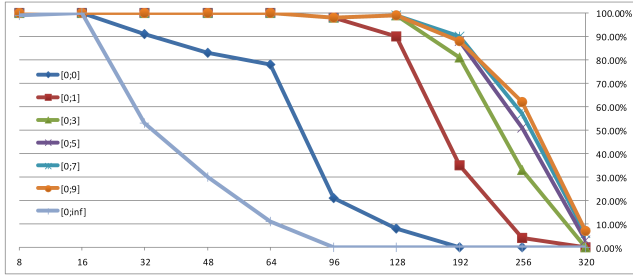[1]The case when no path exists due to static obstacles is not considered in this work as being out of interest.

Figure 1: Success rate (y-axis) of the prioritized planner with different safe-start-intervals for 8-320 agents (x-axis) on empty 32x32 grid.

Table 1: Success rate (SR) and average runtime (t), makespan (Msp) and flowtime (Flt) of the prioritized planner with random and deterministic re-scheduling for 16-160 agents (warehouse environment).

	Random re-scheduling				Deterministic re-scheduling			
	SR	t	Msp	Flt	SR	t	Msp	Flt
16	1.00	0.01	33.31	285.15	1.00	0.01	33.31	285.15
32	1.00	0.04	36.68	591.30	1.00	0.04	36.68	591.30
64	1.00	0.22	41.25	1274.94	1.00	0.17	42.05	1252.82
96	1.00	0.68	50.05	2212.14	1.00	0.49	50.42	2149.31
128	1.00	5.73	62.23	3543.92	1.00	1.40	66.15	3322.52
160	0.42	117.82	79.73	5220.90	0.99	28.28	100.50	5298.85
192	0.00	—	—	—	0.00	—	—	—

him to step away. To mitigate this issue we introduce the notion of *safe-start-interval* (*SSI*). *SSI* is a time interval $[0, k], k \geq 0$, during which any robot is prohibited to interfere with the start location of any other robot (and it is allowed afterwards). This introduces flexibility that can positively influence planner's effectiveness, e.g. the ability to solve "unsolvable" instances (conducted experimental evaluation supports this claim).

3 EXPERIMENTAL EVALUATION

To evaluate the proposed techniques and to estimate how they influence the performance of prioritized planning we experimented with two different types of environments, e.g. the one with no obstacles present (32 x 32 empty grid) and the one modeling the warehouse from [7] (21x35 grid with 100 blocked cells out of 735). The number of agents differed from 8 to 320 depending on the environment. 100 instances per each number of agents per environment were randomly generated and used for evaluation. Initial scheduling was "shortest first", which means that robots with the lower estimates of path lengths are assigned higher priorities. All experiments were conducted using C++ implementation of the state-of-the-art prioritized planner AA-SIPP(m) [12][2].

We will now highlight only a small subset of experimental results that provide a strong support for the claim that introducing *SSI* and deterministic re-scheduling positively influences the performance of prioritized multi-robot path planning compared to baseline (no re-scheduling, no safe-start-intervals) and to random re-scheduling.

Figure 1 depicts the success rate of the planner on 32x32 empty grid with no re-scheduling and *SSI* endpoints set to 0 (baseline, no *SSI*), 1, 3, 5, 7, 9 and ∞ (traversing the start locations of the low-priority robots is completely prohibited). As one can see introducing *SSI* leads to a significant increase of the success rate, especially when the number of robots is large. For example, more than 80% of instances for 192 agents (density 18.75%) are solved when *SSI* endpoint is set to $k \geq 3$ compared to 0% for planning without *SSI*. Setting k to infinity is the worst option, although it is known from previous research [3] that such setting guarantees finding a solution in case given path planning instance satisfies certain criteria.

Table 1 shows the success rate, runtime and quality indicators (makespan, flowtime) of the planner relying on random re-prioritization and on the suggested deterministic re-prioritization

for warehouse-like environment from [7]. *SSI* was set to 5 and 5 minutes time cap was introduced for this experiment as the planner relying on random re-scheduling might run for inadequately long times (as number of restarts is not limited). Randomized planner was invoked 10 times on each instance and all the collected data was used for averaging.

As one can see the success rate tends to instantaneous drop-down after some critical density of agents is reached, but the proposed deterministic re-scheduling technique leads to better success rate, when this density is near-critical (160 robots in the considered case). Moreover, utilizing the suggested approach results in slightly better flowtime. Finally, the proposed technique is much faster compared to random re-scheduling. The difference in runtimes is evident for higher densities when the need for attempting different priority orderings is vital. For example for 160 agents (density 25%) our approach, besides solving almost 100% of tasks, compared to 42% achieved by random restarts, produced solutions 4 times faster. The only downside of the suggested deterministic re-scheduling procedure is the increase of makespan. In case it is of extreme importance one might consider to change the initial priority scheduling scheme to "longest first" as suggested in [10] and to apply randomized restarts.

4 CONCLUSIONS

In this work we have studied the behavior of prioritized multi-robot path finders applied to the problem of navigating disk-shaped robots on uniform cost grids. We have suggested two novel procedures that can be implemented on top of any prioritized solver and evaluate them empirically. The experiments demonstrate that they drastically reduce the number of failures when compared to baseline algorithm. At the same time, the planner utilizing them outperforms the one relying on random restarts (the most commonly used re-scheduling technique in prioritized path planning) both in terms of effectiveness, e.g. success rate, and efficiency, e.g. runtime.

REFERENCES

[1] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. 2001. Optimizing schedules for prioritized path planning of multi-robot systems. In *Proceedings of The 2001 IEEE International Conference on Robotics and Automation (ICRA-2001)*, Vol. 1. IEEE, 271–276.

[2] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. 2002. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and autonomous systems* 41, 2 (2002), 89–99.

[3] M. Čáp, P. Novák, A. Kleiner, and M. Selecký. 2015. Prioritized planning algorithms for trajectory coordination of multiple mobile robots. *IEEE Transactions on Automation Science and Engineering* 12, 3 (2015), 835–849.

[4] Christopher M Clark and Stephen Rock. 2001. Randomized motion planning for groups of nonholonomic robots. In *Proceeding of The 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space*. Canadian Space Agency.

[5] L. Cohen, T. Uras, T.S. Kumar, H. Xu, N. Ayanian, and S.Koenig. 2016. Improved solvers for bounded-suboptimal multiagent path finding. In *Proceedings of The 25th International Joint Conference on Artificial Intelligence (IJCAI-2016)*. 3067–3074.

[6] M. Erdmann and T. Lozano-Pérez. 1987. On multiple moving objects. *Algorithmica* 2 (1987), 1419–1424.

[7] Hang Ma, Jiaoyang Li, TK Kumar, and Sven Koenig. 2017. Lifelong Multi-Agent Path Finding for Online Pickup and Delivery Tasks. In *Proceedings of The 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS-2017)*. International Foundation for Autonomous Agents and Multiagent Systems, 837–845.

[8] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant. 2015. Conflict-based search for optimal multiagent path finding. *Artificial Intelligence Journal* 218 (2015), 40–66.

[9] T. S. Standley. 2010. Finding optimal solutions to cooperative pathfinding problems. In *Proceedings of The 24th AAAI Conference on Artificial Intelligence (AAAI-2010)*. 173–178.

[10] Jur P Van Den Berg and Mark H Overmars. 2005. Prioritized motion planning for multiple robots. In *Proceedings of The 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2005)*. IEEE, 430–435.

[11] Glenn Wagner and Howie Choset. 2011. M*: A complete multirobot path planning algorithm with performance bounds. In *Proceedings of The 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2011)*. 3260–3267.

[12] Konstantin Yakovlev and Anton Andreychuk. 2017. Any-Angle Pathfinding for Multiple Agents Based on SIPP Algorithm. In *Proceedings of The 27th International Conference on Automated Planning and Scheduling (ICAPS-2017)*. 586–593.

[13] Jingjin Yu and Steven M LaValle. 2016. Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. *IEEE Transactions on Robotics* 32, 5 (2016), 1163–1177.