A Complete Multi-Robot Path-Planning Algorithm*

JAAMAS Track

Ebtehal Turki Saho Alotaibi
Computer Science Department, Al-Imam Mohammad Ibn Saud Islamic University
Riyadh, SA
e.t.otaibi@gmail.com,etalotaibi@imamu.edu.sa

ABSTRACT

In the domain of multi-robot path-planning problems, robots must move from their start locations to their goal locations while avoiding collisions with each other. The goal of the research problem that has been tackled in [2] is to find complete solutions for multi-robot path-planning problem, the theoretical analyses already demonstrated the completeness of Push and Spin (PASp) algorithm. However, the PASp algorithm does not guarantee the optimality of the solution.

In this work, some decisions are found within the complete solution that may optimize PASp performance. Two pluggable methods are proposed for this purpose; smooth operation, that aims to reduce the redundant moves in the generated paths, and heuristic search, that evaluates different alternative paths according to its occupancy by other robots. The improved version of PASp algorithm is referred to as PASp+ algorithm. The experimental results showed that, compared to PASp, Push and Swap (PAS), Push and Rotate (PAR), Bibox and the tractable multi-robot path-planning (MAPP) algorithms, adding heuristic search and smooth operation in PASp+ resulted a significant improvement shown in reducing the number of moves for all problem instances with high percentage reaches to 47.8%. However, in higher computation time.

KEYWORDS

complete algorithms; multi-robot; path planning algorithms

ACM Reference Format:
Ebtehal Turki Saho Alotaibi. 2019. A Complete Multi-Robot Path-Planning Algorithm. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Formally, the multi-robot path-planning problem (MPP) has consisted of a graph and a set of robots. In such problems, each robot must reach its destination in a minimum time with a minimum number of moves.

Push and Spin (PASp) algorithm [2, 3] is complete cooperative algorithm for Multi-robot Path Planning problem with rotation (MPPr) for any instance recognized by the solvability test as solvable without any assumption. For each robot r, the PASp algorithm finds a connected path linking the start location to its goal location. When another robot s is detected on a vertex v in r's path, r has the ability to "push" s away if s has a lower priority. If s has a higher priority (i.e., s is already planned), PASp switches to "spin" operation, that allows two robots to swap their locations without altering the configuration of other robots.

2 THE IMPROVED PUSH AND SPIN (PASP+)

The contribution in this work is to provide a brief exploration of the possibilities to improve solution quality through the use of smooth post-processes and heuristics, namely in the area of selecting which path each robot should choose.

2.1 Smooth operation

The Smooth operation is a post-process on the complete solution aims to reduce the generated redundant moves. Smooth operation takes the list of solutions as lists of moves. Each list is indexed with its time-step. Smooth operation iterates over the list of solution in a reverse manner. For each action $\pi = v \leftarrow b$ (robot b occupies vertex v), all actions after π in the reversed sequence are checked for an occurrence of the final vertex v. If such action π' exists, and both of actions are executed by the same robot b, then, all actions executed by robot b between π and π' can be removed, if and only if there is no robot occupies the vertex v in the time between two actions.

2.2 Heuristic search

As described in [2] , the Spin operation represents the dominant factor in both execution time and total path length metrics. The Spin operation will be called when robot a finds a (blocking) planned-robot n (Figure 1) in its connected path p. Hence, one way to improve the path and eliminate Spin operation calls is to try to find another path $p*$ such that $p*$ contains the minimum number of planned-robots. However, if there is no other path, the completeness must be maintained by considering path p.

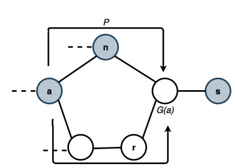
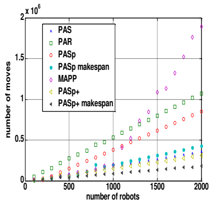


Figure 1: Alternative path computation

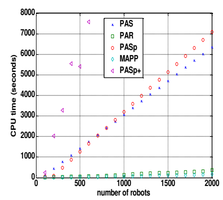
In this scheme, an evaluation function is introduced that evaluates the available paths, according to their lengths and the availability of planned-robots.

*This paper is an extended abstract of an article published in Autonomous Agents and Multi-Agent System [3]

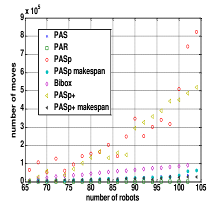
Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.



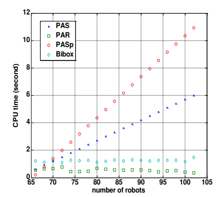
(a) number of moves for AR0603SR map



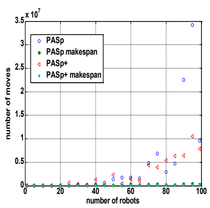
(b) CPU time for AR0603SR map



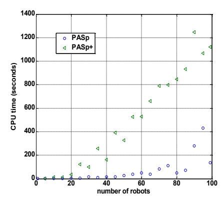
(c) number of moves on random biconnected instances



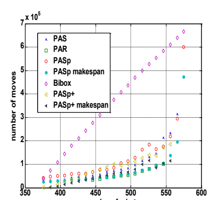
(d) CPU time on random biconnected instances



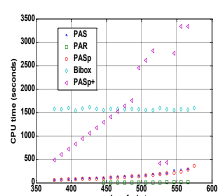
(e) number of moves for random non-tree instances



(f) CPU time for random non-tree instances



(g) number of moves for 24X24 grid



(h) CPU time for 24X24 grid

Figure 2: Comparison results

3 RESULTS AND DISCUSSIONS

The performance of the improved Push and Spin (PASp+) is compared with that of Push and Spin (PASp)[2], tractable multi-agent path planning algorithm for undirected graphs (MAPP)[7], Push and Swap (PAS)[5], Push and Rotate (PAR)[4] and Bibox algorithms[6]. The details of the experimental framework are reported in [1]. Three main factors are evaluated; (1) the total number of moves, (2) CPU time and (3) the makespan on four kind of problem instances; public benchmark instances from Baldurs Gate II [1](Figure 2(a,b)), random biconnected graphs (Figure 2(c,d)), random non-tree graphs (Figure 2(e,f)) and random grids (Figure 2(g,h)).

[1]http://movingai.com/benchmarks/

Table 1: Impact of Heuristic and Postprocess on Push and Spin performance

Instance		Path reduction (average of %)	Makespan reduction (average of %)	Spin calls reduction (max of %)
Large public benchmark instances.	AR0603SR	42.4%	47.5%	25.0%
	AR0307SR	21.6%	18.7%	28.3%
Random biconnected instances.	FG−VR [1]	20.7%	28.4%	24.3%
	FR−VG [2]	47.8%	54.0%	72.7%
Random instances.	FG−VR	32.8%	49.3%	25.3%
	FR−VG	42.9%	63.5%	26.1%
Random grid instances.	FG−VR	20.3%	7.6%	29.3%
	FR−VG	44.9%	52.1%	56.1%

[1] Fixed graph size - Variable robot number [2] Fixed robot number - Variable graph size

4 CONCLUSIONS

This paper succeeds in presenting a heuristic approach to compute a favorable set of moves with which to move a robot towards its destination vertex in the goal configuration to optimize the complete algorithm.

The mathematical proofs already demonstrate that Push and Spin is a more complete algorithm for a wider class of problem instances than the class solvable by the Push and Swap, Push and Rotate, Bibox or MAPP algorithms, and solves any graph recognized to be solvable without any assumptions. The implemented heuristic aims to avoid the paths containing finished robot in order to eliminate Spin operation calls, the smooth operation is a post-process aims to remove the redundant moves in the final solution.

Figure 2 shows that the heuristics and post-process achieved a significant improvement since PASp+ provides the lowest number of moves among all others. Consequently, the makespan of PASp+ shows, also, further improvement. However, PASp+ algorithm consumes it shows too much time to find a solution which one of optimization techniques limitations [8]. The impact of these improvements is summarized in Table.1.

In summary, the following interesting observations can be made from the experiments:

- The original versions of the Push and Spin, Push and Swap, and Push and Rotate algorithms provide closer-to-linear performance on the public benchmark instances when the congestion rate increases.
- The improved Push and Spin algorithm combines the advantages of completeness and optimization techniques.

REFERENCES

[1] Ebtehal Turki Saho Alotaibi and Hisham Al-Rawi. 2016. MRPPSim: A Multi-Robot Path Planning Simulation. *International Journal of Advanced Computer Science and Applications* 7, 8 (2016).

[2] E. T. S. Alotaibi and H. Al-Rawi. 2016. Push and spin: A complete multi-robot path planning algorithm. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 1–8. https://doi.org/10.1109/ICARCV.2016.7838836

[3] Ebtehal Turki Saho Alotaibi and Hisham Al-Rawi. 2018. A complete multi-robot path-planning algorithm. *Autonomous Agents and Multi-Agent Systems* (2018), 1–48.

[4] Boris De Wilde, Adriaan W Ter Mors, and Cees Witteveen. 2014. Push and rotate: a complete multi-agent pathfinding algorithm. *Journal of Artificial Intelligence Research* 51 (2014), 443–492.

[5] Ryan Luna and Kostas E Bekris. 2011. Push and swap: Fast cooperative path-finding with completeness guarantees. In *IJCAI*. 294–300.

[6] Pavel Surynek. 2009. A novel approach to path planning for multiple robots in bi-connected graphs. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on.* IEEE, 3613–3619.

[7] Ko-Hsin Cindy Wang and Adi Botea. 2011. MAPP: a scalable multi-agent path planning algorithm with tractability and completeness guarantees. *Journal of Artificial Intelligence Research* 42 (2011), 55–90.

[8] Jingjin Yu and Steven M LaValle. 2013. Multi-agent path planning and network flow. In *Algorithmic foundations of robotics X*. Springer, 157–173.