Spatio-Temporal A* Algorithms for Offline Multiple Mobile Robot Path Planning

(Extended Abstract)

Wenjie Wang
Nanyang Technological University
School of Computer Engineering
Nanyang Technological University, Singapore
wang0570@e.ntu.edu.sg

Wooi Boon Goh
Nanyang Technological University
School of Computer Engineering
Nanyang Technological University, Singapore
aswbgoh@ntu.edu.sg

ABSTRACT
This paper presents an offline collision-free path planning algorithm for multiple mobile robots using a 2D spatial-time map. In this decoupled approach, a centralized planner uses a Spatio-Temporal A* algorithm to find the lowest time cost path for each robot in a sequentially order based on its assigned priority. Improvements in viable path solutions using wait time insertion and adaptive priority reassignment strategies are discussed.

Categories and Subject Descriptors
I.2.9 [**Artificial Intelligence**]: Robotics – *Workcell and planning.*

General Terms
Algorithms, Experimentation.

Keywords
Path planning, multiple robots.

1. INTRODUCTION
This paper focuses on solving the problem of path planning in multiple robots. During past years, many methods have been proposed to solve the path planning problem of multiple robots. They can be generally divided into coupled or decoupled. In a coupled approach [1], all robots plan their path simultaneously using a centralized planner to avoid colliding into one another. The advantage of a coupled approach is that its solution is complete. However, the dimension of this approach is the sum of degree of freedom of all robots. This means its computational time increases exponentially with the robot count.

An alternative approach is a decoupled approach [2] that reduces the dimension of path planning by making each robot plan its path individually. Associated with the decoupled approach are the issues related to prioritized planning and path coordination. In prioritized planning, each robot is given a priority. The robot with the highest priority plans its path first and its resulting path influences the way the next highest priority robot would plan its path and so on. In path coordination, each robot searches its path independently and then adopts some strategy such as speed modification or stop-and-wait delays to avoid collisions. However, this approach does not guarantee viable solutions even they exist.

In this paper, we introduce a variant of the A* algorithm called the Spatio-Temporal (S-T) A* algorithm for path planning of multiple mobile robots. We have adopted a decoupled approach, where a centralized planner uses the proposed S-T A* algorithm to find the lowest cost path for each robot in a sequentially order based on its assigned priority. Computational time is reduced by searching the path solution in a 2D spatial time map compared to the exhaustive search in a 3D spatial time map [3].

2. SPATIO-TEMPORAL A*
The A* algorithm is a popular path planning algorithm whose solution is complete in a static environment. A dynamic environment could be considered a static one by adding an additional time dimension into the search map. However, long computational time and huge memory resource requirements in large search spaces reduce its usefulness. The proposed S-T A* algorithm solves this problem by searching in a 2D spatial map.

In our decoupled approach, searching order is an important factor that affects the optimality of paths for multiple robots. Our goal is to find a viable solution that will allow all n robots to reach their respective target positions without incurring any collision and to achieve a time-based objective function given by

$$T = \mathrm{argmin}(\max(t_i)) = \max(T_i) \qquad \text{where } i = 1,2\ldots n \qquad (1)$$

Here t_i represents the cooperative time cost of robot R_i, T_i is the individual time cost for robot R_i to reach its destination if there is no other robots. In order to make T as close to $\max(T_i)$ as possible, we adopted a fixed priority that is ordered by the individual time cost T_i. Each robot searches its path sequentially using S-T A* algorithm under fixed priority assignment (S-T-FP A*). As a result of the collision in a crowd environment, a viable solution for all robots using the S-T A* algorithm with fixed priority assignment is difficult to obtain. Two strategies were adopted to improve the performance of our algorithm. The first is a flexible wait time insertion strategy. Our goal is to wait as close to the node where collision has been detected. We insert wait time at the closest possible antecedent node near the collision node. In this way, wait time insertion is not limited to only the starting node [4] but any node in the current path that has already been planned. Preference is given to the node closest to where collision would

Cite as: Spatio-Temporal A* Algorithms for Offline Multiple Mobile Robot Path Planning (Extended Abstract), Wenjie Wang, Wooi Boon Goh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1091-1092.

have happened if the robot did not stop. Under the fixed priority assignment, we call this S-T A* algorithm with wait time strategy the (S-T-W-FP A*) algorithm for short.

The second strategy is a novel adaptive priority re-assignment strategy. In this strategy, given an initial fixed priority assignment above, if the current path searching robot R_i fails to find a viable path to its destination, its priority is raised by one level and is allowed to re-plan its path again. It continues to escalate its priority until it finds a viable path. Combining these two strategies, centralized planner searches the path solutions for all robots using an algorithm named S-T A* with wait time and adaptive priority(S-T-W-AP A*) algorithm. When the number of robots is large, using the adaptive priority strategy can be very time consuming. In addition, since this strategy generates new priority order, a failed priority assignment may be revisited over and over again. We address this by setting an upper bound on the number of new priority assignments and failed priority assignments are noted in order to avoid new priority assignments that will result in failure.

Since a higher priority robot R_1 will not take into account the path planned by a lower priority robot R_2, a collision may happen when R_2 reaches its destination and remain at rest while R_1 has to pass through R_2's destination point. To address this, a lower priority robot will not terminate its search if it has reached its destination until all higher priority robots have got to their destinations. After the lower priority robot R_n reaches its destination, it will check collision at its destination continuously until all higher priority robots R_1 to R_{n-1} has reached their respective destinations. If there is collision, R_n will wait at appropriate node or find an alternative path before reaching its destination.

3. EXPERIMENT RESULT

The simulation results presented were obtained using an Intel(R) core™ 2 quad (2.83 GHz) with 3.25 GB of memory. The simulation program is written in the Java language on the Eclipse development environment. We first compared the ability of each of the three variants of the S-T A* algorithm to find viable paths for all n robots (i.e. success rate) as the number of robots n is increased. The wait time insertion strategy can increase the number of available nodes in the path search map which is not occupied by other robots while search is being preformed. In Figure 1, the better success rate of the S-T-W-FP A* algorithm compared to the S-T-FP A* algorithm shows that the increase in available nodes in the map does improve success rates. However, the S-T-W-FP A* algorithm still performs poorly in a crowded environment. A rigid fixed priority scheme means that when a lower priority robot's path becomes blocked by higher priority ones, no recourse is available but to declare this to be an unsuccessful run. However, when adaptive priority reordering is use in the S-T-W-AP A* algorithm (see Figure 1), significant improvement in success rate is obtained.

Figure 2 show the percentages of runs from all simulation that satisfy the time-based objective function T defined in (1). Under fixed priority assignment, the S-T-W-FP A* algorithm met the objective function better than the S-T-FP A* algorithm. The wait time strategy not only increased success rate but also the number of simulation runs that satisfies objective function T. Unfortunately, the fixed priority strategy falters as the number of robots increased. Under these circumstances, the novel adaptive priority strategy in the S-T-W-FAP A* algorithm is much better in producing higher percentage of runs that can meet the objective function T, besides producing more successful runs.

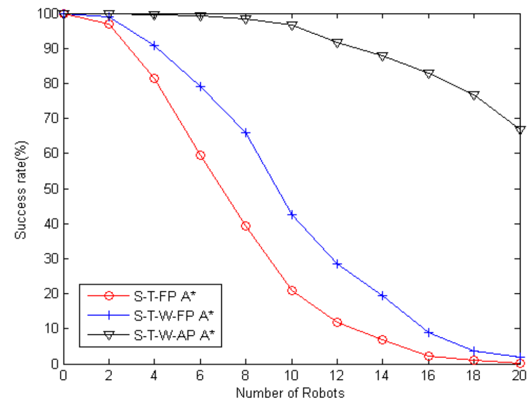

Figure 1. Success rate of S-T-FP A*, S-T-W-FP A*, and S-T-W-AP A* algorithm.

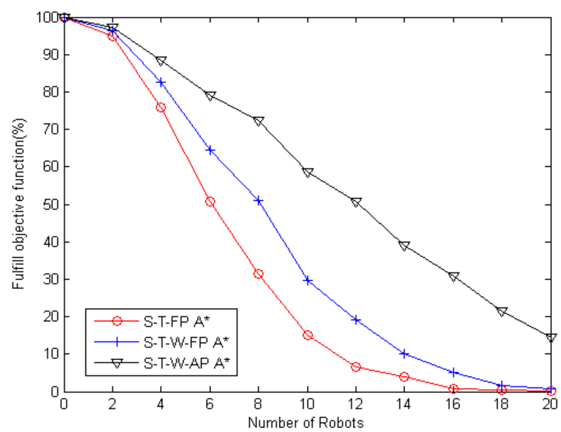

Figure 2. The percentage of runs from all simulation runs that satisfy the objective function *T*, for all three algorithms.

Acknowledgement

This research is funded by the Singapore National Research Foundation IDM for Education grant NRF2008-IDM001-017.

4. REFERENCE

[1] P. Svestka and M. Overmars, "Coordinated path planning for multiple robots," Robotics and Autonomous Systems, vol. 23, no. 3, pp. 125-152, 1998.

[2] Y. Guo and L. E. Parker, A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots, Proceedings of the 2002 IEEE International Conference on Robotics & Automation, pp. 2612 – 2619, 2002.

[3] D. Silver. Cooperative pathfinding. In *AIIDE*, pp. 117-122, 2005.

[4] S.H. Ji, J.S. Choi, and B.H. Lee, A Computational Interactive Approach to Multi-agent Motion Planning, International Journal of Control, Automation, and Systems, vol.5,no.3,pp. 295-306, 2007