Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

RISK-BASED A*: SIMULATION ANALYSIS OF A NOVEL TASK ASSIGNMENT AND PATH PLANNING METHOD

Maojia P. Li
Michael E. Kuhl

Department of Industrial and Systems Engineering
Rochester Institute of Technology
Rochester, NY 14623, USA

Rashmi Ballamajalu

Department of Electrical Engineering
Rochester Institute of Technology
Rochester, NY 14623, USA

Clark Hochgraf

Department of Electrical, Computer, and
Telecommunications Engineering Technology
Rochester Institute of Technology
Rochester, NY 14623, USA

Raymond Ptucha
Amlan Ganguly
Andres Kwasinski

Department of Computer Engineering
Rochester Institute of Technology
Rochester, NY 14623, USA

ABSTRACT

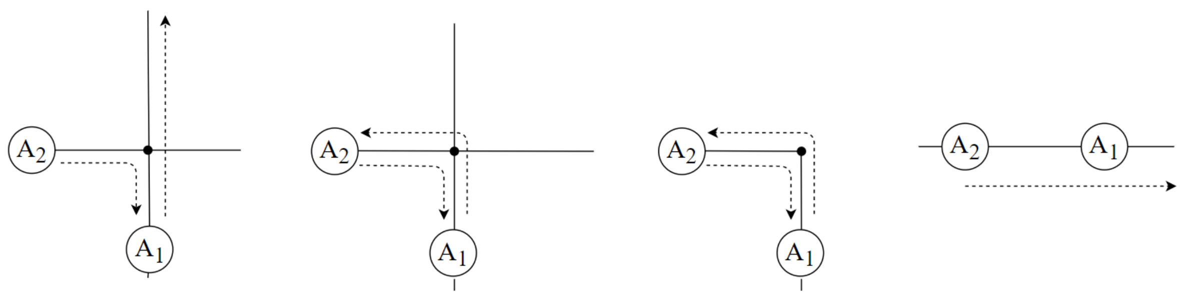
This paper addresses the task assignment and path planning (TAPP) problem for autonomous mobile robots (AMR) in material handling applications. We introduce risk-based A*, a novel TAPP method, that aims to reduce conflict and travel distance for AMRs considering system uncertainties such as travel speed, turning speed, and loading/unloading time. An environment simulator predicts the distribution of future locations for each AMR and constructs a probability map for future AMR locations. A revised A* algorithm generates low-risk paths based on the probability map. A discrete event simulation experiment shows our model significantly reduces the number of conflicts among robots in stochastic systems.

1 INTRODUCTION

In the past decades, material handling robots have been intensively used in distribution centers, container terminals, and manufacturing environments. The previous generations of material handling robots, often referred to automated guided vehicles (AGV), rely on physical guide-paths on the floor or require additional alterations of the environment to localize and navigate themselves. A modern autonomous mobile robots are equipped with self-contained localization systems, such as LiDar or vision systems, that minimize system setup time and improve path flexibility. However, the increase in path flexibility is also challenging for the existing fleet management systems.

Most autonomous material handling systems require a team of robots working together to achieve desired outcomes. The system efficiency and productivity highly depend on the level of collaboration among robots. In this case, a centralized task assignment and path planning system is commonly used to make high-quality decisions. Suppose there is a team of AMRs and a list of delivery tasks, the solution to the task assignment problem determines which task should be assigned to each robot, and solution to the path planning problem indicates how the robots navigate to their destinations. As these two problems are interdependent, a TAPP system solves them simultaneously.

A major challenge for managing AMRs is avoiding conflict. Figure 1 shows some common AMR conflict situations. We assume two AMRs cannot pass each other within a link (highway, aisle, or cross aisle). Figure 1(a) shows a path-cross conflict, where two AMRs try to occupy the same intercept at the same time. In this case, A1 needs to wait for A2 to pass. Figure 1(b) shows a head-on conflict, where two AMRs need to enter each other's link. In this case, one of the AMRs need to perform a T-turn or cross-turn to resolve the conflict. However, these actions require the use of additional links. As shown in Figure 1(c), if the number of AMRs involved in the conflict is greater than or equal to the number of links connected to the intercept, it becomes impossible to resolve the conflict at the particular location. Figure 1(d) show conflicts on a link. If A1 moves in an opposite direction with A2, a head-on conflict occurs. One of the robots needs to move backwards and exit the link. If A1 moves in the same direction as A2, a following conflict occurs. The maximum speed of A2 is limited by A1. When A1 stops to perform loading or unloading, it becomes a node-occupancy conflict. A2 has to wait until A1 finishes. Another head-on or following conflict may take place after A1 finishes.



(a) (b) (c) (d)

Figure 1: AMR conflict examples. Two AMRs are represented by A1 and A2, The dot and solid lines are intercept and links, respectively. A dash line indicates the travel path of an AMR.

The examples shown above reveal that AMR conflict can slow down the traffic and cause AMRs to perform irregular actions. For the worst case scenario, a conflict can lead to a deadlock that requires human intervention. The conflict problems can be mitigated by employing a sophisticated TAPP system. The system collects local information from each AMR to predict their future locations and generates a conflict-free path for the dispatched AMR. However, most existing solutions, such as optimization and safe interval path planning models, assume a deterministic environment. When predicting the future locations of an active AMR (an AMR that actively works on other tasks), the model often assumes the AMR travels on a constant speed with fixed loading and unloading times. However, these assumptions cannot hold due to the stochastic events such as sliding of wheels, acceleration and deceleration, maximum speed affected by payload, loading time impacted by the pallet condition, and other uncertainties in the real-world system.

This paper introduces an A*-based model that aims to reduce AMR conflicts and travel distance. An innovative aspect of our model is it considers uncertainties in AMR travel speed, turning speed, and loading/unloading time. Rather than predicting a single location, the proposed model predicts a discrete distribution for possible locations of each AMR. A risk map is then generated to indicate high-risk conflict locations at each particular time, and so we call our model, *risk-based A**. Rather than imposing hard restrictions, risk-based A* penalizes the use of high-risk locations based on the probability and severity of potential conflict. After generating a low-risk path for each assignment, our TAPP model selects the task that has the lowest probability of causing a conflict.

This remainder of the paper is organized as follows. Related work is discussed in section 2. After explaining the risk-based A* model in section 3, a discrete event simulation experiment is described in section 4. Finally, we share our conclusion and future work in section 5.

2 RELATED WORK

A TAPP problem is often solved with a scheduling or dispatching model. A scheduling model generates a complete delivery schedule for each robot. Confessore, Fabiano, and Liotta (2013) formulate the TAPP problem into a minimum cost flow (MCF) model. The low cost and high solution quality of MCF model has drawn the attention of many researchers (Rashidi and Tsang 2011; Rashidi 2019; Fazlollahtabar 2018b). Fazlollahtabar (2018a) introduce a MCF model that considers the probability and severity of conflict. To reduce robot conflicts and avoid deadlocks, the conflict-free task assignment and path planning problem has emerged in the literature (Nishi, Hiranaka, and Grossmann 2011; Saidi-Mehrabad, Dehnavi-Arani, Evazabadian, and Mahmoodian 2015; Umar, Ariffin, Ismail, and Tang 2012).

In a highly stochastic environment, it becomes difficult to generate or revise a optimal delivery schedule on-line. In this case, a dispatching model is often more favorable in busy environments (Le-Anh and De Koster 2006). A dispatching model, often working with a separate path planning system, assigns a single task to a robot. Most single-attribute dispatching rules make greedy decisions based on the current condition of the environment. Some popular rules include shortest-travel-distance, nearest-station-first, earliest-due-date, and first-come-first-serve (Egbelu and Tanchoco 1984). Some researchers study multi-attribute dispatching algorithms that consider multiple system attributes (Hwang and Kim 1998; Jeong and Randhawa 2001; Bilge, Esenduran, Varol, Öztürk, Aydın, and Alp 2006). Guan and Dai (2009) introduce a multiple-attribute dispatching model that eliminates deadlocks.

One of the most popular path planning algorithms used with dispatching rules is A*. A* is a graph-based algorithm uses a heuristic, or an estimation function to direct its search from start to goal location. A* search requires a pre-defined map or a grid. The cost of moving through this grid is computed and updated at each point as it traverses through the given map. The costs grid cell x^+ is determined by

$$g(x^+) = g(x) + c(x, x^+).$$

$$f(x^+) = g(x^+ + h(x^+)$$

where x is the predecessor of x^+, $g(x^+)$ is the actual cost from start to current cell, $c(x,x^+)$ is the cost moving from x to x^+, $h(x)$ is an estimate from current cell to goal, $f(x^+)$ is indicates the priority of evaluating cell x^+.

The heuristic function distinguishes A* from Dijkstra's by resulting in a directed search, or an educated guess as to where the goal might be. Some of the most frequently used heuristic functions are the Euclidean distance and Manhattan distance measurements. The A* algorithm has been further expanded over the years to accommodate dynamic obstacles. The addition of time as another dimension to the search-space is the general approach. However, this can be computationally expensive and require longer time for the graph to converge to a solution. A method to avoid adding time as another dimension while maintaining a safe interval is the Safe Interval Path Planning (SIPP) concept (Phillips and Likhachev 2011). In SIpp a safe time intervals are finite and defined to be between two unsafe intervals. This results in only a few states for every configuration within the graph. A continuation of the work has been presented in (Narayanan, Phillips, and Likhachev 2012) where an anytime planner has been used based on SIPP. The algorithm arrives at a quick solution and improves the path over time. The experimental results presented show optimal paths under real-time constraints.

The k-shortest path planning problem has been solved in (Wang, Wang, Qin, Wu, Duan, Li, Cao, Ou, Su, Li, et al. 2015) where the shortest paths are computed and the number of edges counted and stored. Based on this information, the path with the least number of turns is selected. Also, the vehicle's speed is modeled as the time dimension and the heuristic value changed at every node after the it has turned. This way the conflicts are divided into two definite states and reduces complexity. A similar approach has been shown in (Jia, Ren, Chen, and Xu 2017) where the speed of the vehicle during straight paths and turns are modeled and the heuristic is modified to fit these time windows.

3 METHODOLOGY

This is section, we explain the details of risk-based A*. As shown in Figure 2, when an AMR becomes available (dispatched AMR), it will send a dispatching request to the TAPP model. The TAPP model collects robot information from active AMRs and waiting task information from the task management system. The gathered information is used to construct a simulation environment that predicts future locations of each AMR at different time steps. The A* algorithm searches for a most favorable assignment and path based on constraints and costs.

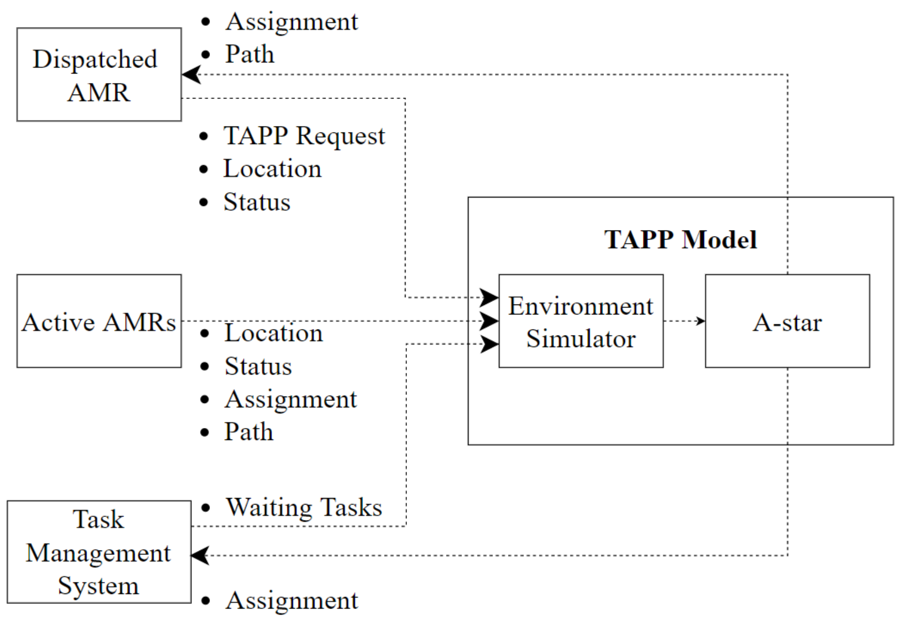


Figure 2: Task Assignment and Path Planning Model Framework.

We implement two A* models that both consider traffic avoidance. The planning horizon is formulated into discrete time $t \in T$ and the environment is divided into cell grids $x \in X$. We assume an AMR can move from one cell to one of its neighbors or remain at the current cell at each time step. The event that active an agent $k \in K$ occupies a cell x at time t is denoted by $a^t_{k,x}$. For the SIPP A* algorithm, the environment simulator predicts a single cell for each active AMR at each time step with the highest probability, $loc^t_k = \arg\max_{x \in X} \mathrm{P}(a^t_{k,x})$. The SIPP algorithm restricts the use of these occupied cells, and the only cost factor in the g-value is the time to reach an available cell. For risk-based A*, the simulator generates a discrete distribute for possible cells of active AMRs. The algorithm only keeps those cells with a probability higher than a minimum threshold, such that $loc^t_k = \{x \in X | \mathrm{P}(a^t_{k,x}) > \underline{\mathrm{P}}\}$. The prediction of moving trajectory is formulated into a sequence of dependent events, such that

$$\mathrm{P}(a^t_{k,x} \cap a^{t-1}_{k,x^-}) = \mathrm{P}(a^t_{k,x} | a^{t-1}_{k,x^-}) \cdot \mathrm{P}(a^{t-1}_{k,x^-}),$$

where $x^- \in loc^{t-1}_k$ is a possible cell for the AMR at $t-1$. The overall probability for event $a^t_{k,x}$ is then expressed as,

$$\mathrm{P}(a^t_{k,x}) = \sum_{x^- \in loc^{t-1}_k} \mathrm{P}(a^t_{k,x} \cap a^{t-1}_{k,x^-}).$$

When t is small, the distribution loc_k^t is narrow as there are fewer possible locations and each of them has a relatively high probability. As t gets larger, the distribution becomes wider, which reflects the fact that the future location of active agent becomes more difficult to predict. The overall probability that cell x is occupied by any active AMR at time t is

$$\mathrm{P}(o_x^t) = 1 - \prod_{k \in K} \left(1 - \mathrm{P}(a_{k,x}^t)\right).$$

Algorithm 1 demonstrates the search process of risk-based A*. Each state s is a pair of cell and time. The *OPEN* set contains a set of states waiting to be examined with their corresponding f-values. In line 5, the $getChild(s)$ function returns a set S^+ containing all possible ending states s^+ with corresponding action a and r-value $r(s,s^+)$. When generating a child, the algorithm defines a set of legal actions $A(s)$ where each action transits the robot into an ending state $s^+ = trans(s,a)$. We define the r-value as the probability any conflict occurs during the transition period $c(s,s^+)$ multiplied by a penalizing constant λ_{x^+}, such that

$$r(s,s^+) = \left(1 - \prod_{i=1}^{c(s,s^+)} \left(1 - \mathrm{P}(o_{x^+}^{t+i})\right)\right) \lambda_{x^+}.$$

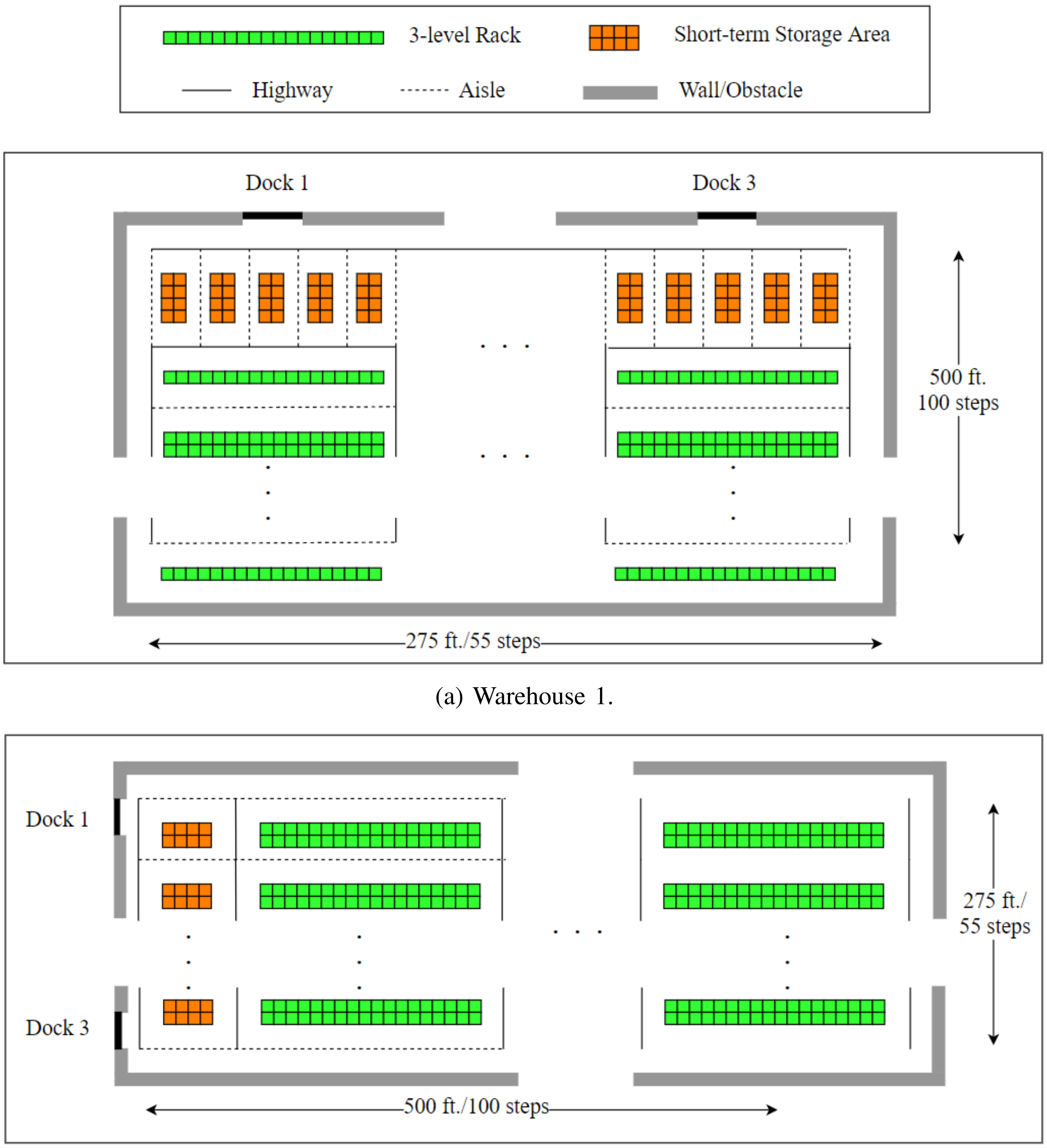
where x^+ is the ending cell of s^+ and t is the ending time in s. The value of λ_{x^+} depends on the geometric location. A good value can be the average time to solve a conflict at the location so that $r(s,s^+)$ becomes the expected resolving time for possible conflict. The transition periods for waiting, moving forward, turning, loading and unloading vary. In line 7 to 9, any state that has $r(s,s^+) > \bar{r}$ is considered a high-risk transition and will be eliminated. In line 10 to 12, if the action is loading/unloading, the algorithm finds a low risk solution. In line 13 to 17, if the ending state has not been examined before or has a lower g-value compared to the history, the algorithm updates g-value, f-value and appends the ending state to *OPEN*.

Algorithm 1: Risk-based A* Algorithm Pseudo-code

1: Initialize $g(s_{start}) = 0$, $OPEN = \emptyset$
2: Insert s_{start} into *OPEN* with $f(s_{start}) = h(s_{start})$
3: **while** not **terminate do**
4:     Remove s with the smallest f-value from *OPEN*
5:     $S^+ = getChild(s)$
6:     **for** each $(s^+, r(s,s^+), a)$ in S^+ **do**
7:         **if** $r(s,s^+) > \bar{r}$ **then**
8:             **continue**
9:         **end if**
10:         **if** $a = loading$ **or** $a = unloading$ **then**
11:             **terminate**
12:         **end if**
13:         **if** s^+ was not examined before **or** $g(s^+) > g(s) + c(s,s^+) + r(s,s^+)$ **then**
14:             $g(s^+) = g(s) + c(s,s^+) + r(s,s^+)$
15:             $f(s^+) = g(s^+) + h(s^+)$
16:             insert s^+ into *OPEN* with $f(s^+)$
17:         **end if**
18:     **end for**
19: **end while**

4 EXPERIMENTATION

The risk-based A* algorithm is compared with regular A* and SIPP A* in the two warehouse systems shown in Figure 3. Both warehouses are roughly 14,000 ft^2 consisting of 3 docks. There are 60 short-term storage locations (STSL) for each dock. The capacity of each warehouse is 9,828 pallet loads in racks. A task is to either to deliver a load from STSL to a rack location (receiving task) or from a rack location to a STSL (shipping task). The pallets can only be loaded or unloaded from the aisles. The warehouse is divided into 4ft. $\times$ 4ft. grid cells. We assume an AMR travels at 4ft./s. It takes 5 seconds to perform a turn and 20 seconds to load/unload a pallet.



(a) Warehouse 1.

(b) Warehouse 2.

Figure 3: Warehouse Layouts.

The A* algorithms are examined in deterministic and stochastic systems. In a stochastic system, we introduce a 2% variance the travel speed, turning time, loading/unloading time. In each system, we test 3 fleet sizes for 100 trials. At each trail, the system generates random initial locations for $K+1$ AMRs. The first K AMRs are randomly assigned to some tasks with travel paths determined by regular A*. The examined A* selects a task from $N \sim Uniform(1,10)$ tasks and generates pickup and delivery paths. Each task has 50% to be shipping or receiving task. The STSL and rack location of a task is uniformly selected from all possible locations.

Table 1 shows the number of conflicts, average travel distance, and average computation time in Warehouse 1 with deterministic and stochastic paths. For the deterministic case, when using regular A*, the number of conflicts is always above 90. For SIPP and risk-based A*, there is no conflict across 100 trials. The average travel distances between SIPP and risk-based A* are also very close. The distance increases as the fleet size increases. The shortest and longest computation time are always from regular A* and risk-based A*, respectively. For the stochastic setup in warehouse 1, the conflict for SIPP becomes much higher than risk-based A*. As the fleet size increase, the conflicts increase as well as travel distance and computation cost for risk-based A* and SIPP.

Table 2 demonstrates the results from deterministic and stochastic setups in Warehouse 2. In the deterministic case, both SIPP and risk-based A* avoid conflicts successfully. The travel distances from SIPP and risk-based A* are very close. However, the conflicts of SIPP becomes much higher in the stochastic cases. When using the same number of AMRs with the same A*, the conflicts and average travel distance generally decrease compare to warehouse 1. The decrease in travel distance also reduce the computation time for risk-based A* significantly.

Table 1: Warehouse 1 – Number of conflicts (Conf.), average travel distance (Dist.) in feet, and average computation time (Time) in ms under deterministic and stochastic scenarios.

		Regular A*			SIPP A*			Risk-base A*		
Scenario	Fleet	Conf.	Dist.	Time	Conf.	Dist.	Time	Conf.	Dist.	Time
Deterministic	10	95	1010.9	93.6	0	1115.5	112.5	0	1114.6	780.3
	15	97	971.2	94.7	0	1149.2	123.4	0	1148.9	1145.6
	20	98	971.1	97.9	0	1154.7	134.3	0	1154.6	1319.5
Stochastic	10	94	969.1	91.1	33	1071.7	109.5	4	1095.6	827.9
	15	98	971.3	105.7	42	1132.6	136.4	5	1166.8	1322.1
	20	99	967.3	102.5	54	1174.6	140.6	6	1220.0	2013.2

Table 2: Warehouse 2 – Number of conflicts (Conf.), average travel distance (Dist.) in feet, and average computation time (Time) in ms under deterministic and stochastic scenarios.

		Regular A*			SIPP A*			Risk-base A*		
Scenario	Fleet size	Conf.	Dist.	Time	Conf.	Dist.	Time	Conf.	Dist.	Time
Deterministic	10	73	951.5	86.3	0	1007.4	101.6	0	1006.8	387.9
	15	85	872.9	85.5	0	943.8	107.9	0	943.8	386.4
	20	90	900.2	90.0	0	990.1	119.4	0	989.6	559.4
Stochastic	10	74	903.2	84.7	15	957.7	98.8	0	968.2	441.0
	15	85	938.5	87.4	18	1002.0	108.6	4	1012.9	472.7
	20	91	932.4	88.8	23	1021.7	118.3	4	1036.5	761.7

5 CONCLUSION

In this paper, we introduce a risk-based A* algorithm that aims to reduce AMR conflicts and hence improves system efficiency. The proposed algorithm is compared with regular A* and SIPP A* through single delivery experiments. The numerical results show SIPP and risk-based A* can both find conflict-free path with similar travel distance in deterministic systems. After introducing variance in AMR travel speed, turning time, loading and unloading time, the risk-based A* outperforms SIPP for conflict avoidance. The impact of risk-based A* on system productivity over a long period awaits further study.

ACKNOWLEDGEMENTS

This work is sponsored in part by a grant from Toyota Material Handling of North America and The Raymond Corporation.

REFERENCES

Bilge, Ü., G. Esenduran, N. Varol, Z. Öztürk, B. Aydın, and A. Alp. 2006. "Multi-attribute Responsive Dispatching Strategies for Automated Guided Vehicles". *International Journal of Production Economics* 100(1):65–75.

Confessore, G., M. Fabiano, and G. Liotta. 2013. "A Network Flow Based Heuristic Approach for Optimising AGV Movements". *Journal of Intelligent Manufacturing* 24(2):405–419.

Egbelu, P. J., and J. M. Tanchoco. 1984. "Characterization of Automatic Guided Vehicle Dispatching Rules". *The International Journal of Production Research* 22(3):359–374.

Fazlollahtabar, H. 2018a. "Risk-based Minimum Cost Flow Model for AGVs Path Planning". *The Open Automation and Control Systems Journal* 10(1):1–7.

Fazlollahtabar, H. 2018b. "Scheduling of Multiple Autonomous Guided Vehicles for an Assembly Line Using Minimum Cost Network Flow". *Journal of Optimization in Industrial Engineering* 11(1):185–193.

Guan, X., and X. Dai. 2009. "Deadlock-free Multi-attribute Dispatching Method for AGV Systems". *The International Journal of Advanced Manufacturing Technology* 45(5-6):603.

Hwang, H., and S. H. Kim. 1998. "Development of Dispatching Rules for Automated Guided Vehicle Systems". *Journal of Manufacturing Systems* 17(2):137–143.

Jeong, B. H., and S. U. Randhawa. 2001. "A Multi-attribute Dispatching Rule for Automated Guided Vehicle Systems". *International Journal of Production Research* 39(13):2817–2832.

Jia, F., C. Ren, Y. Chen, and Z. Xu. 2017. "A System Control Strategy of a Conflict-free Multi-AGV Routing Based on Improved A* Algorithm". In *Proceedings of 24th International Conference on Mechatronics and Machine Vision in Practice*, 1–6. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Le-Anh, T., and M. De Koster. 2006. "A Review of Design and Control of Automated Guided Vehicle Systems". *European Journal of Operational Research* 171(1):1–23.

Narayanan, V., M. Phillips, and M. Likhachev. 2012. "Anytime Safe Interval Path Planning for Dynamic Environments". In *Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4708–4715. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Nishi, T., Y. Hiranaka, and I. E. Grossmann. 2011. "A Bilevel Decomposition Algorithm for Simultaneous Production Scheduling and Conflict-free Routing for Automated Guided Vehicles". *Computers & Operations Research* 38(5):876–888.

Phillips, M., and M. Likhachev. 2011. "Sipp: Safe Interval Path Planning for Dynamic Environments". In *Proceedings of 2011 IEEE International Conference on Robotics and Automation*, 5628–5635. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rashidi, H. 2019. "Simulation and Evaluation of Network Simplex Algorithm and Its Extensions for Vehicle Scheduling Problems in Ports". *International Journal of Maritime Technology* 11:1–12.

Rashidi, H., and E. P. Tsang. 2011. "A Complete and an Incomplete Algorithm for Automated Guided Vehicle Scheduling in Container Terminals". *Computers & Mathematics with Applications* 61(3):630–641.

Saidi-Mehrabad, M., S. Dehnavi-Arani, F. Evazabadian, and V. Mahmoodian. 2015. "An Ant Colony Algorithm (ACA) for Solving the New Integrated Model of Job Shop Scheduling and Conflict-free Routing of AGVs". *Computers & Industrial Engineering* 86:2–13.

Umar, U. A., M. Ariffin, N. Ismail, and S. Tang. 2012. "Priority-based Genetic Algorithm for Conflict-free Automated Guided Vehicle Routing". *Procedia Engineering* 50:732–739.

Wang, C., L. Wang, J. Qin, Z. Wu, L. Duan, Z. Li, M. Cao, X. Ou, X. Su, W. Li et al. 2015. "Path Planning of Automated Guided Vehicles Based on Improved A-Star Algorithm". In *Proceedings of 2015 IEEE International Conference on Information and Automation*, 2071–2076. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

MAOJIA P. LI is a Ph.D. student in Engineering at Rochester Institute of Technology. His research interests include simulation modeling and analysis, machine learning, and deep learning. His email address is mxl8487@rit.edu.

MICHAEL E. KUHL is a Professor in the Industrial and Systems Engineering Department at Rochester Institute of Technology. His research interests include modeling and simulation of stochastic arrival processes, and the application of simulation to autonomous material handling, healthcare, and manufacturing systems. He is a member of the WSC Board of Director representing the INFORMS Simulation Society. He has also served WSC as Proceedings Editor (2005), Program Chair (2013), and Mobile App Chair (2014-2019). His email address is Michael.Kuhl@rit.edu.

RASHMI BALLAMAJALU is a master student in Electrical Engineering at Rochester Institute of Technology. His research interests include robotics, simulation, and autonomous metarial hnadling systems. His email address is rb4609@rit.edu.

CLARK HOCHGRAF Associate Professor in the Department of Electrical, Computer, and Telecommunications Engineering Technology at Rochester Institute of Technology. His research interests include machine learning, deep learning, and robotics. His email address is cghiee@rit.edu.

RAYMOND PTUCHA is an Associate Professor in the Computer Engineering Department at Rochester Institute of Technology. His research interests include machine learning, computer vision, robotics, graph processing and signal processing, all with an emphasis with deep learning. His the chair of the local IEEE Signal Processing Society. His email address is rwpeec@rit.edu.

AMLAN GANGULY is an Associate Professor in the Computer Engineering Department at Rochester Institute of Technology. His research interests are in energy-efficient interconnection architectures for multicore chips and multichip systems using novel technologies such as wireless and photonic interconnects and data center networks. His email address is axgeec@rit.edu.

ANDRES KWASINSKI is a Professor in the Computer Engineering Department at Rochester Institute of Technology. He is Chief Editor for the IEEE SigPort and Area Editor for the IEEE Signal Processing Magazine. His research interests include cognitive radio and networking, multimedia communications and networking, and Internet-of-Things. His email address is axkeec@rit.edu.