Int j simul model 20 (2021) 3, 583-594
Original scientific paper

SIMULATION ANALYSIS OF ROBOTIC MOBILE FULFILMENT SYSTEM BASED ON CELLULAR AUTOMATA

Li, W.; Miao, L. & Yang, P.#
Research Center for Modern Logistics, Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518055, China
E-Mail: yang.peng@sz.tsinghua.edu.cn (# Corresponding author)

Abstract
This paper analyses the picking performance of a robotic mobile fulfilment system (RMFS) and proposes a Simulation Framework of RMFS based on cellular automata (SFRMFSCA). Many previous RMFS simulation platforms stipulate all aisles to be set up in a fixed directional road network for one-way lines. The warehouse robot had to travel an unnecessarily long distance to perform tasks. We relax the one-way constraint on aisles and cross aisles in the warehouse and allocate the right of way among the warehouse aisles and cross-aisles intersection by referring the idea of traffic light and traffic flow control to the RMFS warehouse scenario. To improve the efficiency of RMFS order picking, this paper designs a comprehensive strategy combining adaptive traffic light update rule, deadlock detection and recovery algorithm, and traffic control to improve the traffic flow of the system. A series of numerical experiments show that the comprehensive strategy designed in this paper can effectively improve the order picking efficiency of RMFS and reduce the probability of scale deadlock. These results and strategies provide a useful reference for designers who initially set up the RMFS warehouse.
(Received in May 2021, accepted in July 2021. This paper was with the authors 1 month for 2 revisions.)

Key Words: Robotic Mobile Fulfilment System (RMFS), Warehouse Performance, Cellular Automaton Model, Simulation

1. INTRODUCTION

As the key hub in logistic operations, the warehouse faces the challenges of small orders, large assortment, tight delivery schedules, and varying workloads depending on special sales or discounts [1]. A warehouse is labour-intensive, and companies are increasingly seeking new storage and order-picking technologies to reduce operational costs and increase throughput, especially e-commerce companies that sell fast-turnover goods and experience strong fluctuations in demand [2]. According to [3], order picking is the highest priority activity to improve productivity in a warehouse due to its high contribution to the total operating costs of a distribution centre. It can be even stated that the service-level and performances of the whole supply chain rest upon the efficiency of the ordering picking system (OPS) [4].

The robotic mobile fulfilment system (RMFS) is a new type of automated storage and picking system that is more efficient than the traditional parts-to-picker system. This technology has been widely used in various e-commerce shopping platforms, such as Amazon, Taobao and JD.COM in China. In addition, as shown in [3, 5], the picking rate of a parts-to-picker system may be as much as double that of traditional picker-to-parts systems, where 50 % of the picker time is spent travelling around the warehouse. However, installing an RMFS typically requires a multimillion-dollar investment, most of which is spent on the robots that carry the pods. The question thus arises of how to improve the warehouse picking efficiency by reducing the robot travel distance or travel time, which is especially important given that time is of great economic value for the planning and control of the system. Many previous research studies have stipulated that the aisles and cross aisles are one-way [6-10], the main purpose is to increase the space utilization rate of the warehouse. As a result, warehouse robot has to travel longer distances to complete its tasks. This is equivalent to taking the picking time of order in exchange for space utilization, which will reduce the order picking efficiency more or less in a long run. Besides, some RMFS have a long aisles layout, the extra distance may be very long. And the method to allocate the right of way in the crossroad between aisles and cross aisles is also a problem that need to be studied.

This paper studies bidirectional aisles and cross aisles of the two-way road plus virtual traffic lights to model the problem, and develops the Simulation Framework of RMFS based on Cellular Automata (SFRMFSCA), which can help to explore the effect to different warehouse settings. Collision and deadlock have always been hot topics in RMFS modelling research, and this paper enriches the current RMFS simulation research from a new perspective. And this paper also proves the idea that the order picking efficiency of the RMFS system can be optimized through traffic control in the warehouse. Finally, this paper designs a comprehensive strategy combining adaptive traffic light update strategy, deadlock detection and recovery algorithm and traffic control to improve traffic flow of the RMFS, and designed experiment proves the effectiveness of the comprehensive strategy.

The remainder of the paper is organized as follows: In section 2 a short literature review about order picking efficiency (*OPE*) of RMFS is presented. In this section, all relevant questions regarding the topic are explained. In section 3 the SFRMFSCA was presented. In section 4 the numerical experiments are designed and the result was analysed. The last section represents concluding remarks.

2. LITERATURE REVIEW

In this chapter, the literature review regarding the aspects relevant to this paper is presented. The focus is set on the research about RMFS, cellular automata and all relevant questions that follows.

2.1 Robotic mobile fulfilment system

The design of the warehouse systems can be grouped into three hierarchical categories [4]: system analysis, design optimization, and operations planning and control. From the system analysis level, which is essentially the viewpoint of the present research, many warehouse simulation models have been designed to study warehouse picking efficiency [7]. These include the closed queueing network model for RMFS proposed by Nigam et al. [11], which estimates order throughput time for single-line orders in an RMFS with a turnover class-based storage policy. Lamballais et al. and Azadeh et al. extended the work of Nigam et al. by deriving travel-time expressions for multi-line and single-line orders in an RMFS with storage zones [2, 12]. They developed a semi-open queueing network to estimate the average order cycle time and optimize the use of robots and workstations [2]. Zou et al. also proposed a semi-open queueing network [13] and found that the handling-speed-based assignment rule outperforms the random assignment rule when workers differ significantly in handling time and that the neighbourhood-search approach is very nearly an optimal assignment rule (requires less time). Yuan and Gong [14] built a queueing network model to describe an RMFS with two protocols for sharing robots between pickers and calculating the optimal robot number and speed, and they provide effective design rules for an RMFS. In 2017, Merschformann et al. [8] built a fine-grained simulation framework called "RAWSim-O" that demonstrates a real-world application of a simulation framework by integrating simple robot prototypes based on vacuum-cleaning robots, which is extremely helpful for studying how to control systems involving multiple mezzanine floors. While, all of this paper are assume that aisles and cross aisles are directional [2], and there can only be vehicles travelling in one direction at the same time. However, in some actual warehouse system, to improve the efficiency of order picking, there are still parallel road vehicles that can accommodate two-way roads. Inspired by this kind of roadway environment, we developed a traffic light control strategy suitable for the RMFS system and proposed an

RMFS simulation framework based on cellular automata, which broaden the methods used in RMFS simulation.

2.2 Cellular automata

The concept of cellular automata was originally developed in the 1940s and is a mathematical idealization of a physical system in which time and space are discrete, and every physical quantity can take on only a finite set of discrete values. A cellular automaton (CA) consists of a regular uniform lattice (or "array"), usually infinite in extent, with a discrete variable at each site. A CA evolves in discrete time steps, with the value of the variable at one site being affected by the values of variables at sites in the "neighbourhood" of the previous time step. The neighbourhood of a site is typically the site itself and all immediately adjacent sites. The cellular variables at each site are updated simultaneously ("synchronously") based on the values of the variables in their neighbourhood at the preceding time step and according to a definite set of "local rules" [15].

Cellular automata are famous for allowing efficient computer simulations that express how the neighbourhood influences the individual (or "cell"). This approach is already widely used in many computer simulations fields, be it in the natural sciences or humanities, such as biology, computer science, physics, transportation, the military, human behaviour etc. It is an essential approach to analyse and understand traffic-flow dynamics and has received significant research attention [16-23]. Some classical CA models have been proposed, such as 184 models [22], the Nagel–Schreckenberg (NS) model [24] and its transformation, the Biham–Middleton–Levine (BML) model [25], velocity-dependent randomization model [16], and velocity effect model [26]. Among these models, the Nagel-Schreckenberg model is the most popular and most applied cellular automaton model by far [24].

In summary, the article researches the order picking process of RMFS and uses a cellular automaton model to simulate the moving behaviour of ROBOTs, which broadens the method used in warehouse simulation. Besides, the effect of traffic light's updated rules and their variance were also studied.

3. DESCRIPTION OF SFRMFSCA MODEL

This section introduces assumptions and preliminary settings for the SFRMFSCA. First, we consider a warehouse with blocks' layout have $2 \times n$ pods, where n is the adjacent pallet number in one block at its length (see the warehouse configuration in Fig. 1, in which $n = 6$). The order picking process can be described as follows. Once the order is assigned to a workstation, robots can fetch the goods for it. Goods are stored on inventory pods. A robot moves to a pod (from blue lattice V to orange lattice P in Fig. 1), lifts it, and brings the pod to a workstation (from lattice P to workstation 1 in Fig. 1) for order picking, using the aisles and cross-aisles. Once the picker has retrieved the required goods from the pod, the robot transports the pod to a storage location (original place or somewhere else that storage policy required) and stores it there (from workstation 1 to the yellow lattice P for fixed storage). During the move in the RMFS, the intersection of the aisles and cross-aisles have a virtual traffic light to allocate the right of way for the warehouse robot to avoid crashing. The status of traffic lights has only two types, green light (GL) or red light, and these states guide the robot's road selection for the next time step, which means that, if the traffic light turn green, then the robot can go straight, turn left or right, or make a U-turn (which corresponding to i, ii, iii and iv, respectively for a car in front of aisles A, light purple in Fig. 1), and if the light stays green for one incoming street for a time *T*, then it stays red for the other streets for the same time interval *T*. However, if its state is red, the robot must wait at the end of the road until the light changes to green. For example, an intersection (see the light yellow circle area in Fig. 1) connect 4 routes A, B, C and

D. If the traffic light in the head of aisles A turn green, then the light to B, C and D must turn red at the same time. Besides, how to arrange the order of the light show green is also a topic for traffic control, such as clockwise order (the traffic light in A shown green, then B, C and D, respectively, light yellow circle in Fig. 1), anticlockwise ($A \to D \to C \to B$), 8-like ($A \to C \to B \to D$) and anti-8-like ($A \to C \to D \to B$).

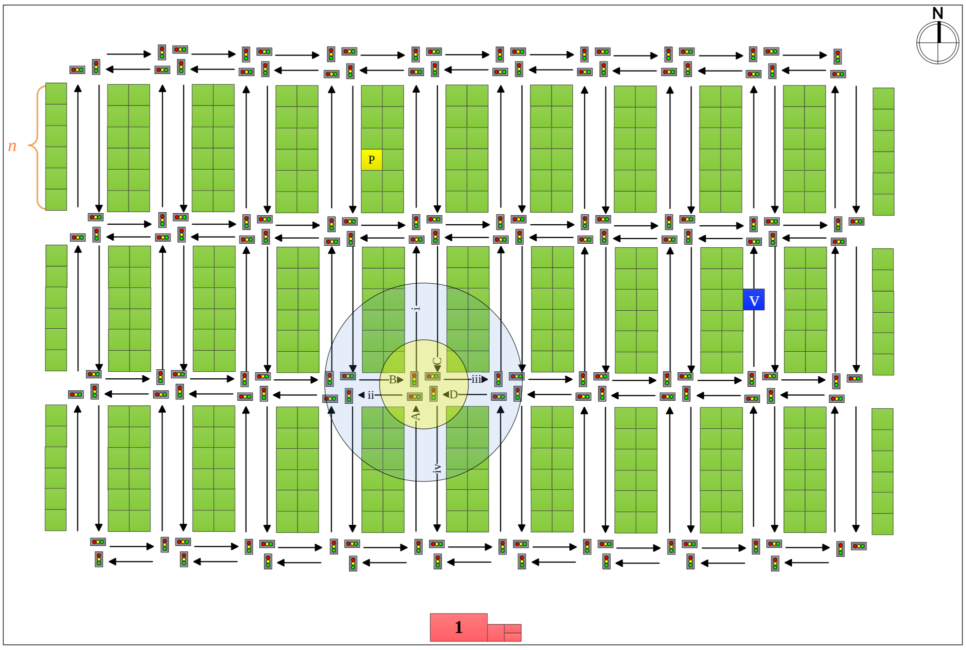


Figure 1: The warehouse map of SFRMFSCA.

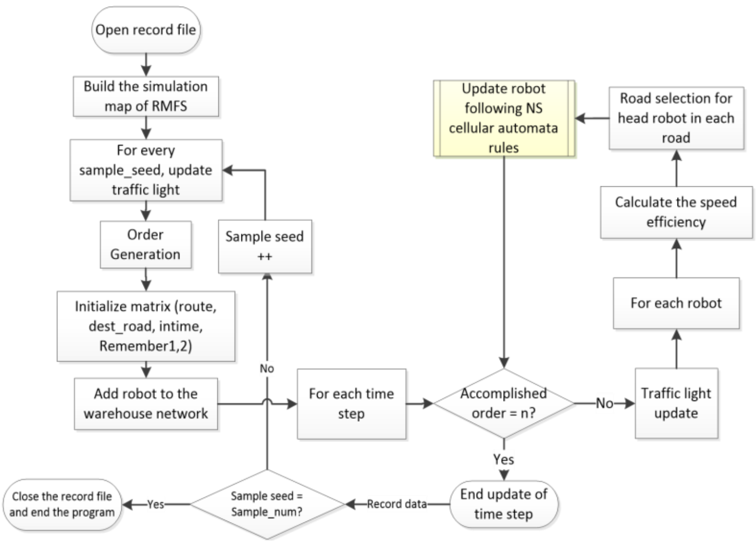
Second, we will describe the order execution system. As shown in the scenario above, there are three OD (origin-destination) pairs for an order to be picked. For each order, the robot moves along the shortest road from origin to destination. This path-finding strategy is guided by the Dijkstra shortest-path algorithm. If more than one shortest road exists for a given OD pair, then the robot chooses the path with the maximum mean velocity to the next intersection. Li et al. [19] have tested the efficiency of the strategy. If the maximum mean velocity is also the same, then this shortest path is selected at random. In addition, the time required for a robot to lift or drop a shelf and the order-picking time at the workstation is omitted, which means that the robot has infinite efficiency for lifting and dropping a pod and that the workstations in the SFRMFSCA have infinite picking efficiency. The reason we set the scene in this way is trying to focus on the robots' order fulfilment process and their interaction in doing their tasks. This is necessary and important especially when there are a huge number of robots in the system shuttling between storage area and workstation(s).

Third, we introduce robots' updating system based on Nagel-Schreckenberg (NS) CA model. Robot displacement in the SFRMFSCA is vital for the problem description. We redesigned the update rule of each robot in the system according to the NS model as follows: **Acceleration:** If the velocity v of a robot is less than $v_{\max}$ and if the distance value to the next robot ahead is greater than $v + 1$, the speed is incremented by one [$v \to v + 1$]. **Slowing down (due to other cars):** If a robot at site i sees the next robot at site $i + j$ (with $j \leq v$), it reduces its speed to $j - 1$ | [$v \to j - 1$], and **Car motion:** Each robot advances v sites. When simulation starts, time is discretized into time steps (TS), which can be regarded as the simulation clock. In each time step, the update rule above is implemented. Each robot has a maximum velocity v_{max}, which means that, if the robot moves at maximum velocity, it can advance v_{max} lattice steps, so we can have only three update rules compared with the four rules of the classical NS model (the missing one is randomization). From the SFRMFSCA' standpoint, an ideal warehouse environment is controllable, so randomization is negligible. After explaining the three main parts of the SFRMFSCA model above, we list the detail of the simulation parameters default settings in Table I.

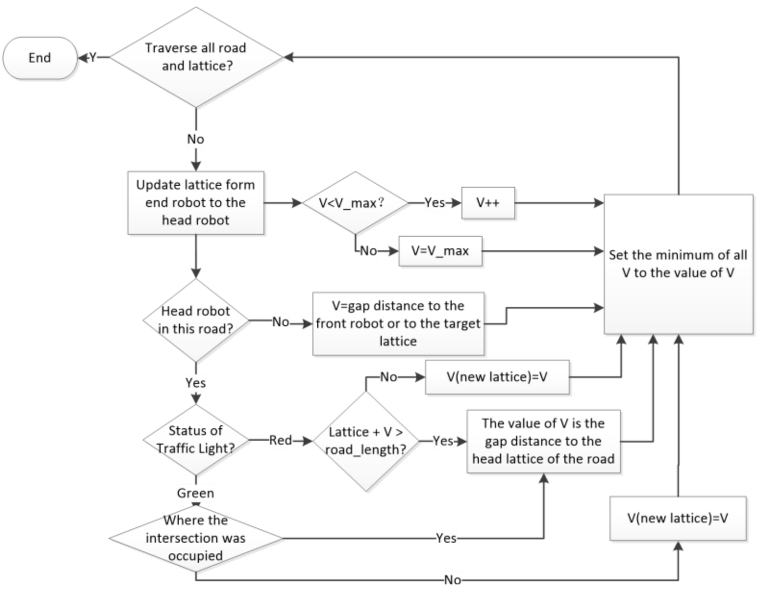
Table I: Parameters of SFRMFSCA warehouse (default value).

Parameter	**Value**	**Parameter**	**Value**
Warehouse area	1360 m^2 (34×40)	GT/Traffic cycle	2TS/8TS
Order size for each simulation	50, 100 or 500 orders	Update order	Anticlockwise
Block shape	2×6	Robot number	14 or 1-30
Warehouse's block	3×10	Workstation number	1
Order dispatch rule	Sequential processing	Lift-right preference	Lift
Storage policy	Random	Maximum Speed	2 lattice/TS
Road selection	Shortest distance $\rightarrow$ Maximum average speed $\rightarrow$ Random (SMR)		

We diagram the algorithm in Fig. 2 as follows.



a) Flow chart of main programming



b) Flow chart of NS cellular automata

Figure 2: Flow chart of the SFRMFSCA program.

4. NUMERICAL EXPERIMENT

The efficiency of order picking for an RMFS can be affected by many factors, such as the number of warehouse robot, the traffic light update rule used and the traffic control of the robots in the system. We conducted a series of numerical experiments to study the sensitivity of different warehouse configurations to order picking efficiency (*OPE*) and Time step used (*TSU*) to verify the effectiveness of the designed adapted traffic control strategy. All computations were executed on a PC with an AMD Ryzen 5 4600H CPU@ (3 GHz) and 16 GB RAM and with the implementation of the C++ in Visual Studio 2017. Each experiment was repeated more than 50 times under the same experimental conditions, with just the random seed varying among experiments.

4.1 Optimal warehouse robot number

How to decide the optimal number of robots for an RMFS warehouse? This question is very basic and fundamental for the parameter setting. If the number of warehouse robot sets too small, the efficiency of the RMFS cannot be fully utilized. Otherwise, if the number sets too large, it may decrease the efficiency and even cause waste of cost. To verify the optimal warehouse robot number for the warehouse of Fig. 1, we conducted an experiment (# 1) to verify the number and the parameter setting of the experiment was shown in Table II. What the experiment intended to do is try to find a value about the warehouse robots number to maximize the *OPE* for a warehouse like Fig. 1.

Table II: Parameters settings of experiment # 1.

Parameter	Value	Parameter	Value
Warehouse area	1360 m^2 (34 × 40)	GT/Traffic cycle	2TS/8TS
Order size for each simulation	60 orders	Update order	Anticlockwise
Block shape	2 × 6	ROBOT number	**4-20**
Warehouse's block	3 × 10	Workstation number/Pos.	1/mid-bottom
Order dispatch rule	Sequential processing	Lift-right preference	Lift
Storage policy	Random	Maximum speed	2 lattice/TS
Road selection	SMR	Repeated time	50

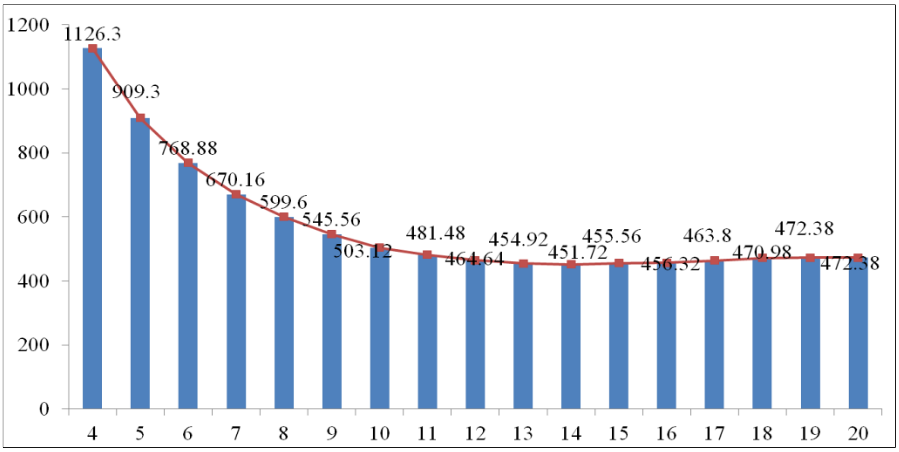


Figure 3: Time step used with the increase of robots' number for 60 orders.

The *TSU* result of the simulation shown in Fig. 3. We can get that when the number of robot increase from 4 to 20, the decrease of the order picking time is not a linear function. It can be observed that the picking time for a certain number of orders (60 orders) is first reduced and then increased, rather than a continuous decrease process. We can also get the optimal warehouse robot number is 14 because under this scenario the *OPE* is highest compared with other settings for the time used for a fixed number of order picking. And its value is 451.72 – the least of all the *TSU* results. By comparing and observing the order picking process of different robots in the visual interface, we found that when the number of robots increases to a certain amount, the interaction between vehicles will have greater impact on the order picking process, especially near the workstation area. The collection and dispersal of such vehicles can easily cause traffic congestion. Therefore, the robot needs to wait or change lines to complete the order, thus causes more time wastage.

4.2 Effect of traffic light's update rule

To verify the effect of the traffic light's update rule to the *OPE*, we analysed several numerical experiments run with optimal settings for one workstation with 4-30 robots. The variable was marked by bold font in Table III. This experiment (# 2) was repeated 100 times to ensure that the results are stable and reliable.

Table III: Parameters settings of experiment # 2.

Parameter	Value	Parameter	Value
Warehouse area	1360 m^2 (34× 40)	GT/Traffic cycle	2TS/8TS
Order size for each simulation	500 orders	Update order	**Clock/Anti/8/Anti-8**
Block shape	2 × 6	ROBOT number	**4-30**
Warehouse's block	3 × 10	Workstation number/Pos.	1/right-bottom
Order dispatch rule	Sequential processing	Lift-right preference	Lift
Storage policy	Random	Maximum speed	2 lattice/TS
Road selection	SMR	Repeated time	100

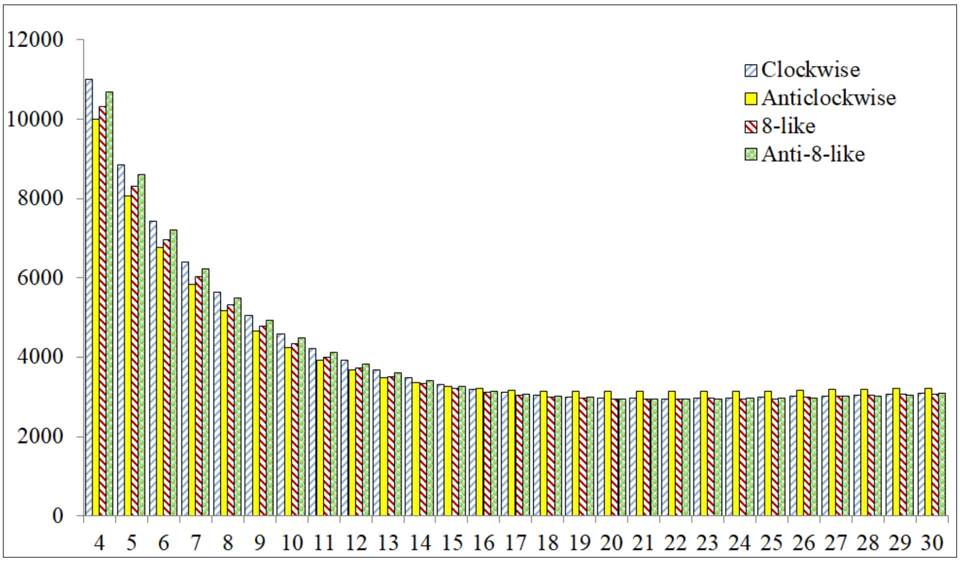


Figure 4: *TSU* for different number of warehouse robots and different traffic light update rule.

The results are shown in Fig. 4. and we can observe that when the number of robots is less than 13, the Anticlockwise traffic light update rule is the best traffic control light update rule among these four methods tested and then come to the 8-like update rule, the anti-8-like update rule and clockwise traffic light update rule, respectively. However, with the increase of the number of robots, the influence of different traffic light update rules on order picking efficiency gradually decreases, and the counter clockwise traffic update rule gradually become the worst among the four traffic light update rules, which the 8-like traffic light update rules gradually gains the advantage. It was also noted that the clockwise update rule was the worst performer, with a time-step difference of less than 10 % between the two traffic light updates. In order to better understand the detailed information of the difference between counter clockwise traffic light updating rules and clockwise traffic light updating rules, we also calculated the ratio of the difference between clockwise traffic light updating rules and counter clockwise traffic light updating rules. It can be observed from the figure that the time-step waste rate caused by the regulation of traffic light update rules is not in a linear relationship with the number of robots, but shows a trend of decreasing first and then increasing. The minimum value is obtained when the number of robots is about 15 or 16, which is also the key point where the clockwise and counter clockwise traffic signal light update rules reverse the efficiency of traffic control for order picking. It can be mutually verified with our previous analysis conclusion.

In addition, there is a “by-product” conclusion in this experiment. When we adopt different traffic light update strategies, the optimal number of robots for the RMFS is different. For example, it can be observed from Fig. 4 that when the RMFS adopt the 8-like or anti-8-like traffic light update rule, the optimal number of robots is 21. In contrast, the number of robots is 22 when the clockwise traffic update rule is adopted. Though this difference is not very big, we can also get the info that the optimal number of robots of an RMFS can be affected by changing the traffic light update rule.

4.3 Adaptive traffic light control strategy

It can be seen from the chapter 4.2 that traffic light can be used to improve the efficiency of order picking by changing the rules used in traffic light update. To fully utilize these characteristics, we designed a new adaptive traffic light control strategy (ATLCS) to optimize the *OPE* by regulating traffic flow. Under the adaptive traffic light control strategy, if the number of robot vehicles on the current road reaches more than half of the road length (discrete space can be compared with integer value), the traffic lights at the road head (every aisle or cross-aisle have two roads, and the robot follows left move preference, from the tail of the road in and the head of the road out) will automatically turn green to prevent the occurrence of greater congestion, which is more efficient than a fixed update rule. The basic parameter setting of the experiment (# 3) to verify this efficiency is shown in Table IV.

Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process. In our CARMFSCA simulation, the process is order picking and the resource is lattice space in the warehouse. Collision and deadlock are an essential research topic for RMFS simulation no matter in industry or academia. Especially scale deadlock is fatal to RMFS simulation because its occurrences will cause the paralysis of the entire RMFS system.

Table IV: Parameters settings of experiment # 3.

Parameter	Value	Parameter	Value
Warehouse area	1360 m^2 (34 × 40)	GT/Traffic cycle	1TS/4TS
Order size for each simulation	500 orders	Update order	**Anticlockwise/ATLCS**
Block shape	2 × 6	ROBOT number	**4-30**
Warehouse’s block	3 × 10	Workstation number/Pos.	1/right-bottom
Order dispatch rule	Sequential processing	Lift-right preference	Lift
Storage policy	Random	Maximum speed	2 lattice/TS
Road selection	SMR	Repeated time	100

In this experiment, we compare the efficiency of order picking between the anticlockwise traffic light update rule and ATLCS with the increase of warehouse robot number. Besides, with the increase of robots' number, scale deadlock may occur in RMFS. As can be seen from the Fig. 4, the new and adaptive traffic light rule can effectively reduce the *TSU* and improve the *OPE*. Scale deadlock rate increases rapidly after there are more than 23 robots in the RMFS warehouse system. Besides, when the number of robots reaches 27, scale deadlock will occur at a probability of more than 95 % in the SFRMFSCA warehouse.

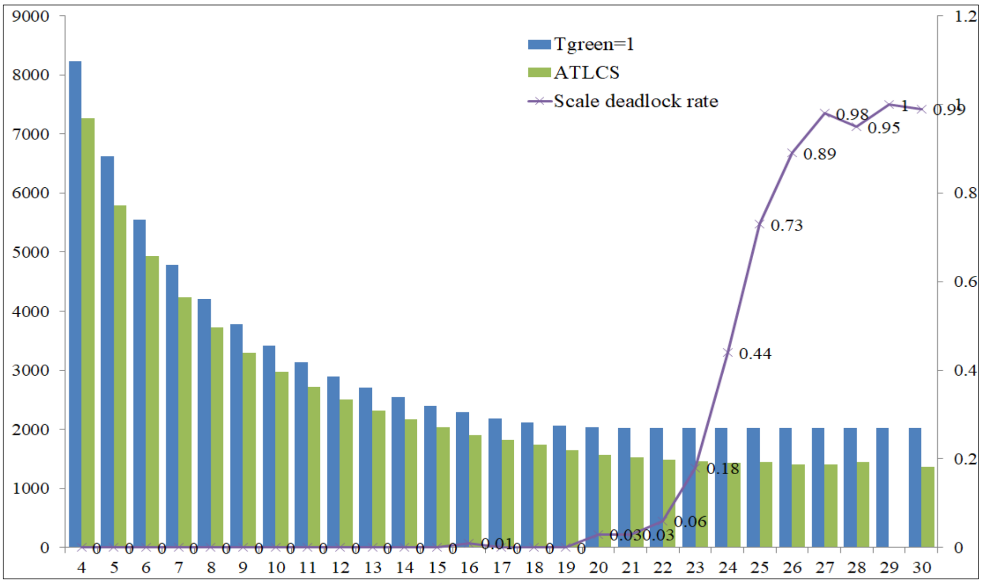


Figure 5: *TSU* and scale deadlock rate with the increase of warehouse robots number.

4.4 Comprehensive strategy

After the analysis of experiment # 3, we focus on solving the deadlock problem in a warehouse from a traffic control and optimization perspective, design a comprehensive strategy (CS) that combining adaptive traffic light system, deadlock detection and recovery and traffic control at bottleneck area. The deadlock detection and recovery algorithm use a depth-first method to search for the deadlock road rings. If a ring was found, a recovery algorithm will be called to solve the deadlock by changing the destination road of a head robot on one road. Besides, adaptive traffic light update strategy and traffic control were also integrated into the comprehensive strategy to optimize the traffic flow and to improve the *OPE*. The parameter of experiment # 4 was shown in Table V.

Table V: Parameters settings of experiment # 4.

Parameter	**Value**	**Parameter**	**Value**
Warehouse area	1360 m^2 (34 × 40)	GT/Traffic cycle	1TS/4TS
Order size for each simulation	500 orders	Update order	**Anticlock./ATLCS/CS**
Block shape	2 × 6	ROBOT number	**10-30**
Warehouse's block	3 × 10	Workstation number/Pos.	1/right-bottom
Order dispatch rule	Sequential processing	Lift-right preference	Lift
Storage policy	Random	Maximum speed	2 lattice/TS
Road selection	SMR	Repeated time	100

The following conclusions can be drawn from Fig. 6 that (a) The comprehensive strategy can effectively improve the *OPE* and such improvement will be more and more significant with the increase of the number of robots. (b) Although the adaptive traffic light control strategy can effectively improve the efficiency of order picking, the probability of scale deadlock increases rapidly after there are more than 20 robots, and the deadlock rate is close to 97 % when there are about 29 robots. (c) The CS is not as efficient as the ATLCS in *OPE* when the number of robots is small, but with the increase of the number of robots in the system, the difference is gradually decreasing, and the strategy of ATLCS alone has been risky due to the high incidence of scale deadlock. (d) This CS combines the high efficiency of the adaptive traffic light control strategy with the obstacle avoidance mechanism of deadlock detection and recovery algorithm and traffic control, and effectively solves the traffic problems at specific congestion points, thus realizing the unity of deadlock avoidance and high *OPE*.

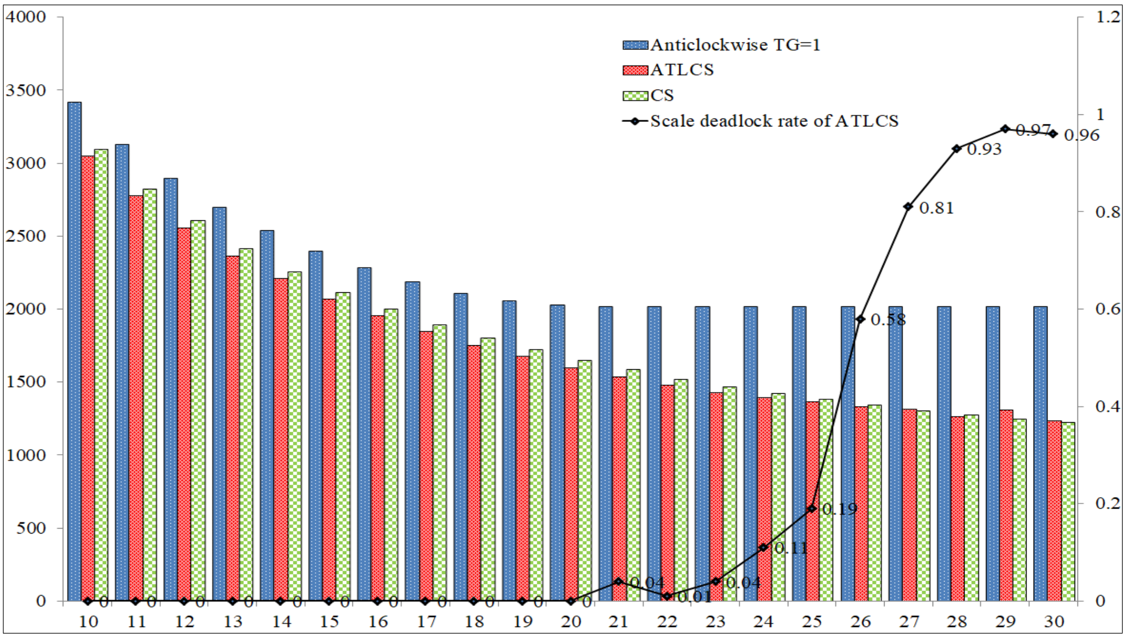


Figure 6: *TSU* and scale deadlock rate with the increase of robots' number under different scenarios.

5. CONCLUDING REMARKS

This paper proposes a new warehouse simulation model (SFRMFSCA) and uses it to investigate the optimal location of picking stations, the optimal number of robots, and the effect of traffic control in a warehouse of about 1400 m^2. The model provides several characteristics compared with the existing warehouse simulation models: 1) Robots do not crash when the cellular automaton model is used for robot navigation in warehouses. 2) We introduce traffic lights to allocate the right-of-way, which is more intuitive than other warehouse simulation models or software. 3) The robot speed is described in more detail, and we consider the acceleration and deceleration of robots in bidirectional aisles and cross-aisles road based on cellular automata. 4) This article clarified the optimization effect of traffic control on *OPE* and designed a comprehensive strategy to verify the effectiveness of this strategy. The main idea behind the SFRMFSCA model is that it emphasizes the effectiveness of traffic control for robots on *OPE* improvement based on cellular automata. In summary, the SFRMFSCA provides a more detailed description of robots' movement on bidirectional warehouse aisles and cross-aisles and designed a comprehensive strategy to optimize the *OPE*.

Several areas are worth exploring in future research. For example, robot vehicles can move more freely to adapt to different storage picking environments, design more efficient traffic control rules to optimize the *OPE* of RMFS, etc. In addition, the comparison between different RMFS simulation environments is also a valuable research area in the future.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 71572090 and 71771130), the National Key R&D Program of China (No. 2018YFE0105100), the Shenzhen Basic Research Programs (Grants No. JCYJ20180306174223343 and No. JCYJ20190813172201684) and a grant from the Institute for Guo Qiang, Tsinghua University.

REFERENCES

[1] Guo, H.; Yang, P.; Xu, T.; Zhang, C. (2019). Local return routing strategy in a flow-picking system, *2019 IEEE 6th International Conference on Industrial Engineering and Applications*, 903-907, doi:10.1109/IEA.2019.8715019

[2] Lamballais, T.; Roy, D.; de Koster, M. B. M. (2017). Estimating performance in a robotic mobile fulfillment system, *European Journal of Operational Research*, Vol. 256, No. 3, 976-990, doi:10.1016/j.ejor.2016.06.063

[3] Tompking, J. A.; White, J. A.; Bozer, Y. A.; Tanchoco, J. M. A. (2010). *Facilities Planning*, 4th edition, John Wiley & Sons, Hoboken

[4] Kovac, M.; Djurdjevic, D. (2020). Optimization of order-picking systems through tactical and operational decision making, *International Journal of Simulation Modelling*, Vol. 19, No. 1, 89-99, doi:10.2507/ijsimm19-1-505

[5] Boysen, N.; Briskorn, D.; Emde, S. (2017). Parts-to-picker based order processing in a rack-moving mobile robots environment, *European Journal of Operational Research*, Vol. 262, No. 2, 550-562, doi:10.1016/j.ejor.2017.03.053

[6] Merschformann, M. (2018). Active repositioning of storage units in robotic mobile fulfillment systems, Kliewer, N.; Ehmke, J.; Borndörfer, R. (Eds.), *Operations Research Proceedings*, Springer, Cham, 379-385, doi:10.1007/978-3-319-89920-6_51

[7] Merschformann, M.; Lamballais, T.; de Koster, M. B. M.; Suhl, L. (2019). Decision rules for robotic mobile fulfillment systems, *Operations Research Perspectives*, Vol. 6, Paper 100128, 17 pages, doi:10.1016/j.orp.2019.100128

[8] Merschformann, M.; Xie, L.; Li, H. (2017). RAWSim-O: A simulation framework for robotic mobile fulfillment systems, *Journal Logistics Research*, Vol. 11, No. 1, Paper 8, 11 pages, doi:10.23773/2018_8

[9] Zou, B.; Xu, X.; Gong, Y.; de Koster, R. (2018). Evaluating battery charging and swapping strategies in a robotic mobile fulfillment system, *European Journal of Operational Research*, Vol. 267, No. 2, 733-753, doi:10.1016/j.ejor.2017.12.008

[10] Jiang, H. (2020). Solving multi-robot picking problem in warehouses: a simulation approach, *International Journal of Simulation Modelling*, Vol. 19, No. 4, 701-712, doi:10.2507/ijsimm19-4-co19

[11] Nigam, S.; Roy, D.; de Koster, R.; Adan, I. J. B. F. (2014). Analysis of class-based storage strategies for the mobile shelf-based order pick system, *13th IMHRC Proceedings*, Paper 18, 9 pages

[12] Azadeh, K.; de Koster, R.; Roy, D. (2019). Robotized and automated warehouse systems: review and recent developments, *Transportation Science*, Vol. 53, No. 4, 917-945, doi:10.1287/trsc.2018.0873

[13] Zou, B.; Gong, Y.; Xu, X.; Yuan, Z. (2017). Assignment rules in robotic mobile fulfilment systems for online retailers, *International Journal of Production Research*, Vol. 55, No. 20, 6175-6192, doi:10.1080/00207543.2017.1331050

[14] Yuan, Z.; Gong, Y. Y. (2017). Bot-in-time delivery for robotic mobile fulfillment systems, *IEEE Transactions on Engineering Management*, Vol. 64, No. 1, 83-93, doi:10.1109/TEM.2016.2634540

[15] Wolfram, S. (1986). *Theory and Applications of Cellular Automata: Selected Papers, 1983-1986 (Advanced Series in Complex Systems)*, World Scientific Publishing Co, Singapore

[16] Barlovic, R.; Santen, L.; Schadschneider, A.; Schreckenberg, M. (1998). Metastable states in cellular automata for traffic flow, *The European Physical Journal B – Condensed Matter and Complex Systems*, Vol. 5, No. 3, 793-800, doi:10.1007/s100510050504

[17] Fukui, M.; Ishibashi, Y. (1996). Traffic flow in 1D cellular automaton model including cars moving with high speed, *Journal of the Physical Society of Japan*, Vol. 65, No. 6, 1868-1870, doi:10.1143/jpsj.65.1868
[18] Guzmán, H. A.; Lárraga, M. E.; Alvarez-Icaza, L.; Carvajal, J. (2018). A cellular automata model for traffic flow based on kinetics theory, vehicles capabilities and driver reactions, *Physica A: Statistical Mechanics and its Applications*, Vol. 491, 528-548, doi:10.1016/j.physa.2017.09.094
[19] Li, M.; Ding, Z.-J.; Jiang, R.; Hu, M.-B.; Wang, B.-H. (2011). Traffic flow in a Manhattan-like urban system, *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2011, No. 12, Paper 12001, 13 pages, doi:10.1088/1742-5468/2011/12/p12001
[20] Qian, Y.-S.; Feng, X.; Zeng, J.-W. (2017). A cellular automata traffic flow model for three-phase theory, *Physica A: Statistical Mechanics and its Applications*, Vol. 479, 509-526, doi:10.1016/j.physa.2017.02.057
[21] Wolf, D. E. (1999). Cellular automata for traffic simulations, *Physica A: Statistical Mechanics and its Applications*, Vol. 263, No. 1-4, 438-451, doi:10.1016/S0378-4371(98)00536-6
[22] Wolfram, S. (1983). Statistical mechanics of cellular automata, *Reviews of Modern Physics*, Vol. 55, No. 3, 601-644, doi:10.1103/RevModPhys.55.601
[23] Zhao, H.-T.; Liu, X.-R.; Chen, X.-X.; Lu, J.-C. (2018). Cellular automata model for traffic flow at intersections in internet of vehicles, *Physica A: Statistical Mechanics and its Applications*, Vol. 494, 40-51, doi:10.1016/j.physa.2017.11.152
[24] Nagel, K.; Schreckenberg, M. (1992). A cellular automaton model for freeway traffic, *Journal de Physique I*, Vol. 2, No. 12, 2221-2229, doi:10.1051/jp1:1992277
[25] Ofer Biham, A.; Middleton, A.; Levine, D. (1992). Self-organization and a dynamical transition in traffic-flow models, *Physical Review A*, Vol. 46, No. 10, 6124-6127, doi:10.1103/PhysRevA.46.R6124
[26] Li, X.; Wu, Q.; Jiang, R. (2001). Cellular automaton model considering the velocity effect of a car on the successive car, *Physical Review E*, Vol. 64, No. 6, Paper 066128, 4 pages, doi:10.1103/physreve.64.066128