Send Orders for Reprints to reprints@benthamscience.ae

The Open Cybernetics & Systemics Journal, 2015, *9*, 1833-1839 1833

Open Access

Travel Time Model of the Storage/Retrieval Machine for Multi-Deep AS/RS Based on Flexsim

Wang Fan[1,*], Guo Dongfen[1], Tang Shujuan[2], He Hong[3] and Cui Xiao[4]

[1]*Department of Express Delivery and Logistics, Shijiazhuang Posts and Telecommunications Technical College, Shijiazhuang, China;* [2]*Personnel Department, Shijiazhuang Tiedao University, Shijiazhuang, China;* [3]*Beijing CSN Technology Inc, Beijing, China;* [4]*Accounting institute, ShiJiaZhuang Vocational College of Finance & Economics, Shijiazhuang, China*

Abstract: The travel time of the S/R machine in multi-deep AS/RS was researched in this paper based on simulation methods. A flexible model of multi-deep AS/RS with 3600 kinds of plans was built and simulated aiming at the S/R machines travel time. A large number of simulation data was thus obtained, which was then used in the multiple regression analysis. Ultimately, the fitting formulas of the S/R machines travel time were built. The fitting formula of the time of queuing shows that the time of queuing has positive correlations with the number of floors and the number of bays, while negative correlations with the number of levels. Furthermore, the value of the 2860 sets of valid experimental data is far more than this. Using these data got by the simulation model, more valuable conclusions can be obtained in further data mining.

Keywords: Multi-deep AS/RS, multiple regression analysis, S/R machine, simulation, travel time.

1. INTRODUCTION

Multi-deep AS/RS is a kind of AS/RS with multi-deep racks. Multi-deep rack has a depth of several unit loads, which needs fewer aisles between racks so that a significant amount of floor space will be saved. Moreover, a warehouse with such a layout, which is also called compact or super high-density storage systems, has relative short travel times to store and retrieve unit loads.

Multi-deep rack essentially consists of several single-deep storage racks (SR) placed one by one, and so the transport unit loads (TUL) are stored in one of the storage lanes of the SR. Each storage lane of the SR is independently accessible, and so any TUL can be stored in any storage lane at any level of the SR. A typical multi-deep AS/RS has the special design of the storage/retrieval (S/R) machine, which moves the TUL to or from the storage lane of the SR on the either side of the picking aisle.

It needs to be pointed out that when a TUL W is going to be retrieved by a S/R machine, if there are other TUL in front of (the side near the cross tunnel) W, it will be necessary to move them to another proper storage lane first, and then move W to the outside of the AS/RS. The movement of other TUL here is called RELOCATION.

2. LITERATURE REVIEW

Up to now, researches in multi-deep AS/RS concentrate in storage strategies and travel time of the S/R machine. Yu & DeKoster (2009) studied the optimal storage zone boundaries of compact 3D storage system using class-based storage. The results showed that the S/R machine travel time is significantly influenced by the zone dimensions, zone sizes and ABC curve skewness [1]. Lerher (2010) studied the mean cycle time for single and dual command cycles. A simulation model of the selected double-deep AS/RS had been developed to compare the performances of the proposed analytical travel time models [2]. Tompkins (2003) said that in a large warehouse with both fewer types of goods and greater liquidity, double-deep rack has higher efficiency than single-deep rack [3]. Zhang Qian (2012) established travel time models of S/R machine in both double-deep rack and single-deep rack, based on random storage assignment, class-based storage assignment of the picking face and the class-based storage assignment of deep-lane direction [4]. However, the restrictions in the model are too relaxed to represents reality correctly. The restrictions include, for example, “considering the entrance and the re-store points as the same point”, “the classification strategy only considering the case of 2 categories”. For double-deep rack, Huang Wei (2013) studied the expected travel time of the dual-shuttle crane using the slide platform to relocate [5]. But the results are not applicable for multi-deep rack. In addition, the study did not consider the influence of shelf size on travel time. Meanwhile, the model had a lower scalability; it has to be rebuilt to fit additional restrictions.

The existing researches in multi-deep AS/RS are mainly qualitative or establishing models using mathematical methods for quantitative research, few studies using simulation methods can be found. Factors considered in these researches are limited; there are fewer considerations on factors like acceleration, deceleration, handling time, etc. When researches in logistics system are not very complicated or the system is simplified to reduce the complexity, mathematical methods can be used. But multi-deep AS/RS has numerous variables as well as complex causations, which necessarily bring disturbance factors. Therefore, its quantitative formula should be fitting formula based on experimental data rather than mathematical derivation formula. However, the experimental data is very difficult to get in reality, thus it can be gotten through the establishment of the simulation model. Moreover, the mechanism of random numbers in simulation can ensure that the results of the simulation model can adequately represent the reality. Consequently, the simulation method should be the preferred method of researches in multi-deep AS/RS.

3. THE MODELING PREPARATIONS

The S/R machine travel time in multi-deep AS/RS was researched in this paper based on Flexsim — a famous simulation software. A flexible model of multi-deep AS/RS with 3600 kinds of plans was built and simulated aiming at the S/R machines travel time. A large number of simulation data was thus obtained, which was then used in the multiple regression analysis. Ultimately, the fitting formulas of the S/R machines travel time were built. The modeling preparations are as follow.

3.1. Concept Definition

Floor: like the floor of buildings, multi-deep AS/RS may also have a plurality of floors. Each floor in the model is 2m high, including an S/R machine, two symmetrical multi-deep racks (Rack for short), an unloading platform, a delivery platform, and a cross tunnel for the S/R machine walking between the Racks which is 2m wide (see Fig. **1**).

Bay: A column of storage location which is perpendicular to the cross tunnel (see the dotted line in Fig. **1**). When an S/R machine works, it travels along the cross tunnel to the appointed bay first, and then travels along the bay to the appointed storage location.

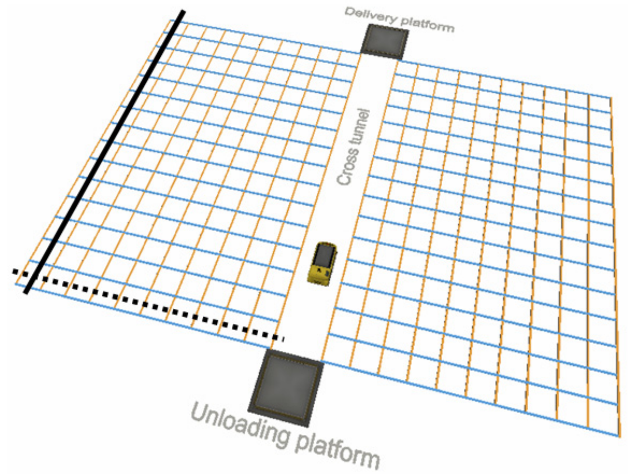

Fig. (1). Sketch map of single floor structure.

Level: A row of storage location which is parallel to the cross tunnel (see the solid line in Fig. **1**).

3.2. Processes

In the model shown in Fig. (**2**), the desired objectives are as follows. After the arrival of a TUL to be stored, it should be transported by the elevator for inflow to the unloading platform of the appointed floor. At the same time, the S/R machine on the right floor will immediately travel to the unloading platform and bring the TUL to the appointed storage location unless it is processing other tasks. In the case of retrieval process, the relocation should first be done if necessary, and then until not blocked, the S/R machine on the same floor will directly transport the TUL to the delivery platform, which will be then carried away by the elevator for outflow.

3.3. Model Parameters

The bay-randomized storage assignment rule is used in storage process, *i.e.* the target storage location is in random bay, and in a level farthest from the cross tunnel. On the con-

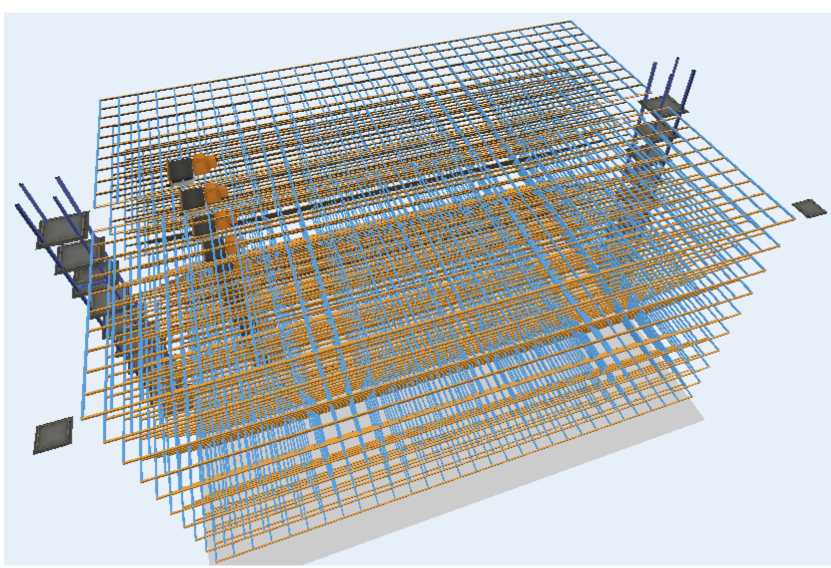

Fig. (2). Visual rendering of the model.

trary, in retrieval process a TUL from random location is to be chosen as the target.

The speed and extension speed of the S/R machines are both constant 2m/s, load time and unload time are both 3s. The speed of the elevators is constant 3m/s. the acceleration and deceleration are not considered but will be easily set in the model if necessary.

Storage location size: 1.5m(H)×1.2m(W)×1.2m(D)

The AS/RS in the model contains n_1 floors, and each has a height of 2m.

Each Rack has n_2 bays and n_3 levels.

The initial occupancy rate of a Rack is $n_4\%$.

The arrivals of storage and retrieval tasks obey the same Poisson distribution n_5 per hour.

Among them, n_1 to n_5 are the independent variables of the model, their ranges are as follows.

$$n_1 \in \{5, 6, 7, 8, 9, 10\}$$

$$n_2 \in \{10, 15, 20, 25, 30\}$$

$$n_3 \in \{5, 6, 7, 8, 9, 10\}$$

$$n_4 \in \{60, 70, 80, 85, 90\}$$

$$n_5 \in \{50, 70, 90, 100\}$$

3.4. Evaluation Indexes

Some indexes should be checked after the stop of the model, which can be the dependent variables of the fitting formulas. These indexes can be divided into two categories: the utilization rate and the expected value of time. For details, see 5.1.

3.5. The Purpose of the Simulation

The ultimate goal of the simulation is to establish the multiple regression equations with the indexes above as dependent variables. According to the ranges of the 5 dependent variables, after the full array, 3600 kinds of plans can be made. And all of them should be calculated through the Experimenter in Flexsim. According to the results of the experiments, then the data analysis as well as the multiple regression analysis can be conducted.

3.6. General Principles and Solutions

Friendly interactive interface should be designed to ensure the setting of the independent variables quick and easy, with which some kind of plan can be directly simulated by setting the independent variables.

The simulation model can have greater flexibility, for instance, factors like acceleration can be arbitrarily adjusted. The extensibility and reusability of the model should be ensured as far as possible. The extensibility means that the model can accommodate more storage strategies, tasks distributions, etc. The reusability means that the modules in the model can be directly copied to meet the demand of model building.

Seeing the big picture, the global table can be used to set the independent variables and record some of the simulation data. Therefore, two global tables "parameters" and "timerecorder" were built. The GUI of the model was designed with cold link to "parameters", by which the values of n_1 to n_5 can be set.

Other principles require constant attention in the modeling process, for example, to avoid using the absolute path when getting a node is to ensure the reusability of the model.

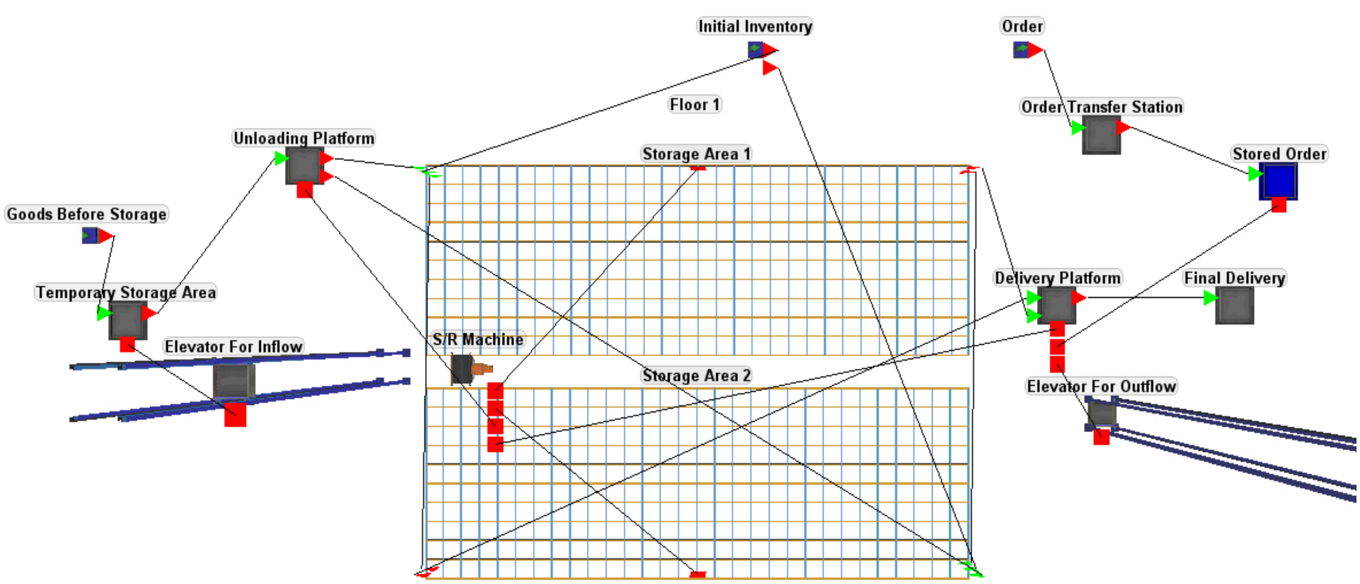

Fig. (3). Single-floor model structures.

4. THE ESTABLISHMENT OF MULTI-DEEP AS/RS SIMULATION MODEL

The establishment of the model can be divided into 3 strides. First, the single floor model was built with most of the logics in the model. After the full establishment of the single floor model, it can be rapidly transformed to the multi-floor model by copying and adjusting. Finally the independent variables were set and called in the model so that, in the model, different plans of the independent variables could be simulated.

4.1. The Single-Floor Model

The structures of the single floor model are shown in Fig. (**3**) which only shows the fixed resources (FR) and their links in the single floor model rather than their locations. The settings of the independent variables will be introduced later.

[Initial Inventory] ([*] is on behalf of an FR named *) was used to produce initial inventories at time 0 sending directly to random Racks.

In storage process, to achieve the collaborative work of the elevator and the S/R machine, the calling relationships among FR were designed using the central ports, and then the coordinated task sequences were created in [Temporary Storage Area].

In retrieval process, the arrivals of outbound tasks were simulated by [Order], and the transportation tasks were achieved by the task sequences set in [Delivery Platform]. Taking the statistics into account, the items from [Order] entered [Stored Order] instead of [Delivery Platform], and the model performed the task sequences code mentioned above by message triggers. The logics of the task sequences were shown in Fig. (**4**). First, the target outbound TUL W* should be defined, then the blocking TUL should be relocated until W* was not blocked. When relocating, the S/R machine picked up the blocking TUL nearest to the cross tunnel and then traveled to a random Rack S* and unloaded the TUL if S* was available. Otherwise, if S* was unavailable, the S/R machine would then travel to the opposite Rack S** to see if available. Similarly, unloaded the TUL if S** was available, or if not, stopped the model.

How to stop the model? The code was written in the storage strategies set in each Rack. The logic of the code in “Place in Level” was easy. It could be achieved by using the option of “First Available Level” provided by Flexsim combined with rotating the Rack to let the first level farther away from the cross tunnel. The code in “Place in Bay” was an improved code based on the “Random Bay if Available” option. Storage only needed to consider the random available column; while relocation needed to consider the situation that the TUL was moved to another bay in the original Rack. When relocating, the S/R machine performed only one task at a time, so the solution may be like this. Set a label of the Rack the bay number x of the TUL on its exit, otherwise the value of the label was 0. When considering which bay to choose, a random available bay different from bay x should be the choice. If no bay could be chosen a number greater than the total bay number of the Rack should be returned such as 100. This could both meet the storage strategies, and ensure the termination of the model at the storage policy failure. Combined with the outbound task sequence, it follows that when both Racks in any floor were unavailable the model would be terminated.

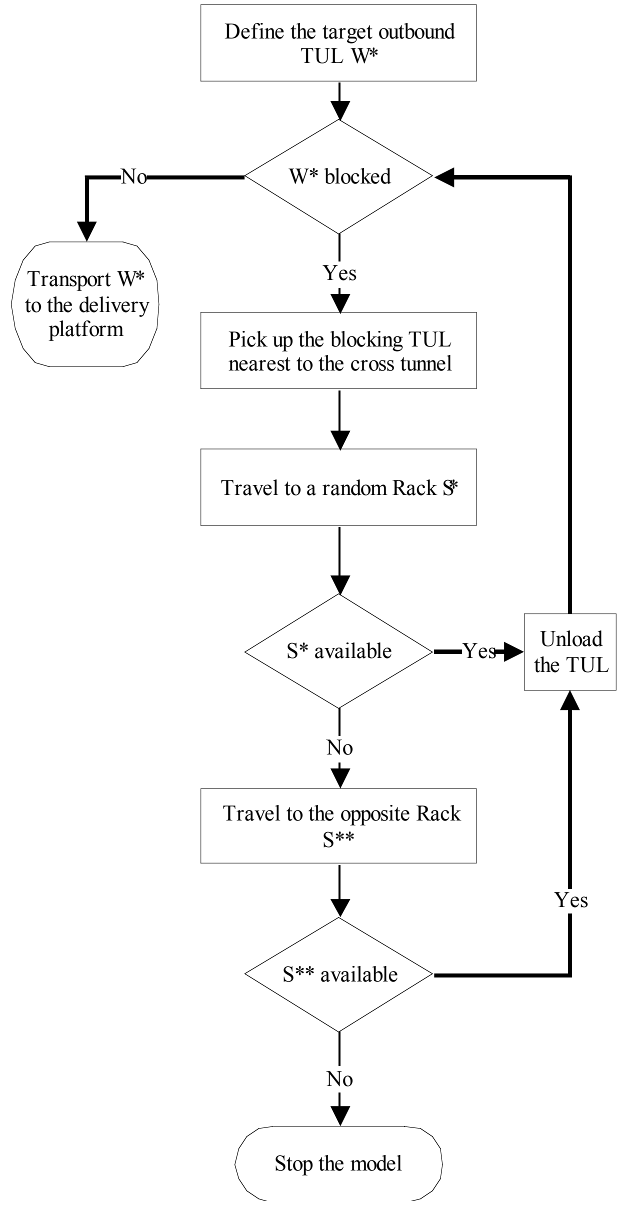

Fig. (4). Flow chart of the outbound task sequence.

4.2. The Multi-Floor Model

When the single-floor model finished, it could be rapidly copied by VisualTool to establish the multi-floor model. Dragged a VisualTool into the model, adjusted the positions, and then moved all of the FR shown in Fig. (**3**) into the VisualTool except [Goods Before Storage], [Temporary Storage Area], [Order], [Order Transfer Station], [Final Delivery], [Elevator For Inflow], [Elevator For Outflow]. And then the VisualTool was on behalf of one floor. Copied and pasted the VisualTool to generate other floors, adjusted the spatial locations of each floor, linked each floor to the outside, and then hided the 3D shape of other floors. As a result the preliminary multi-floor model was formed (consult Fig. **2**).

4.3. The Settings of the Independent Variables

The independent variables were recorded in a global table, and could be called from this table. For ease of use, a GUI was designed (see Fig. **5**)

n_1, as the floor number, would be frequently called in the statistics of the model. So it can be assigned to a global variable. Only n_1 floors participate in the running of the model, the inbound TUL and the orders as well as the initial inventories would not be sent to other floors. And this could be achieved by closing the ports on reset.

n_2 was the bay number, and n_3 was the level number. The size of a rack in Flexsim cannot be set by simply adjusting the numbers of bay and level in the code [6]. Another idea came to solve this. In the model, set the bay number and the level number both to the max values of their ranges. When the numbers of bay and level were ascertained, set the appropriate storage logics in both "Place in Bay" and "Place in Level" of each Rack by calling the global table to ensure the TUL were located in the specified range (within n_2 bays and n_3 levels). And this is equivalent to modifying the size of the Rack.

n_4 was the initial inventories which could be used to calculate the initial number of TUL in each floor. It might be called in [Initial Inventory] of each floor at time 0.

n_5 was the arrival distribution, which could be called in [Goods Before Storage] and [Order]. Note that Poisson (n_5) PC/H equaled to exponential (0, 3600/n_5, 0) S/PC.

5. EXPERIMENTER DESIGN

5.1. The Dependent Variables' Setting

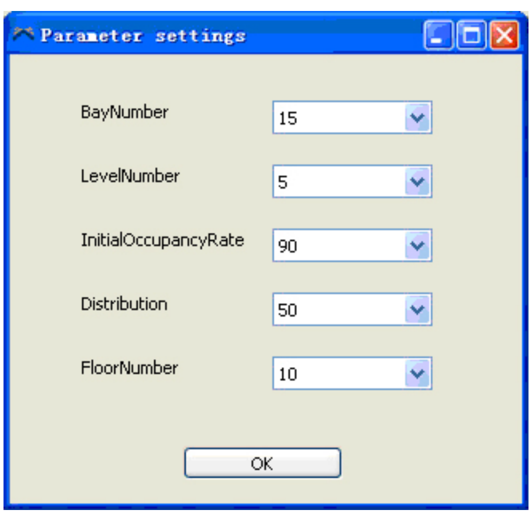

Fig. (5). GUI.

The dependent variables were the indexes checked after the stop of the model, which should be set in the experimenter. They include the utilization rate of the elevator for inflow m_1, the utilization rate of the elevator for outflow m_2 the expected value of utilization rate of the S/R machines m_3, expected value of inbound time m_4, expected value of outbound time m_5, expected value of time of queuing m_6.

Among them, the expected value can be understood as the average value. Queuing means that if the elevator or the S/R machine was occupied thus was unable to perform the current task, then this task needs to wait in line until the elevator or the S/R machine was released.

The calculations of m_1, m_2 and m_3 were simple. They can be obtained by simple modifications based on the "State percentage" option in the experimenter calculating "1- idle rate". Note that when calculating m_3, the floor number n_1 should be considered. In the statistics, the inactive FR should be excluded; otherwise the idle rate would increase.

On the contrary, the calculations of m_4, m_5, m_6 were complicated. Flexsim is unable to distinguish the time of inbound and of outbound [6], hence it needs manually set some logics to distinguish them.

As to m_5, because the outbound time begins in the message trigger of [Delivery Platform] and ends in the entry trigger of [Final Delivery], so in the message trigger of [Delivery Platform] a label could be added to the target TUL to record the beginning time; while in the entry trigger of [Final Delivery] the single outbound time would be calculated by subtracting the label time from the entry time. Add up each outbound time and task sequence number in the global table "timerecorder", which could be called in statistics. There is still a problem. Upon termination of the model, there might still be some outbound TUL in the model whose processing time should be added to "timerecorder". The solution was to set model stop trigger. Write code to walk through all items in the model. If the item was the outbound TUL, its processing time would be calculated by subtracting the label time from the model termination time and then be added to "timerecorder".

As to the time, the total time equals the sum of inbound time, outbound time and idle time, and the queuing time was included in both inbound time and outbound time. So m_4 could be calculated by a simple subtraction while the calculation of m_6 was not so simple. First, it needed to understand the storage location of task sequences in task executers. In the tree of the model the nodes "activetasksequence" and "tasksequencequeue" would be found under the path ">variables" of each S/R machine. The former recorded the task sequence under processing while the latter recorded the task sequences in queue to be processed. That is, when a task sequence was dispatched to an S/R machine, it would enter the "activetasksequence" node and immediately execute if the node was empty. Otherwise, it would enter the

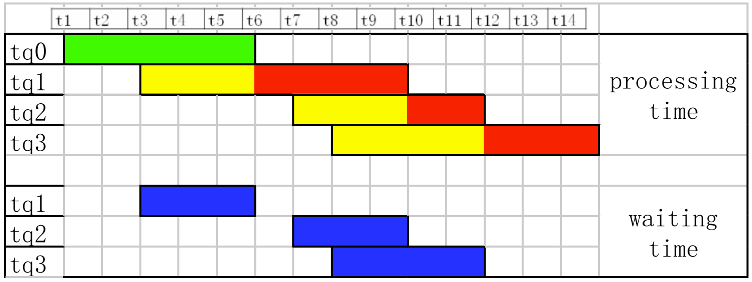

Fig. (6). Method of calculating the queuing time.

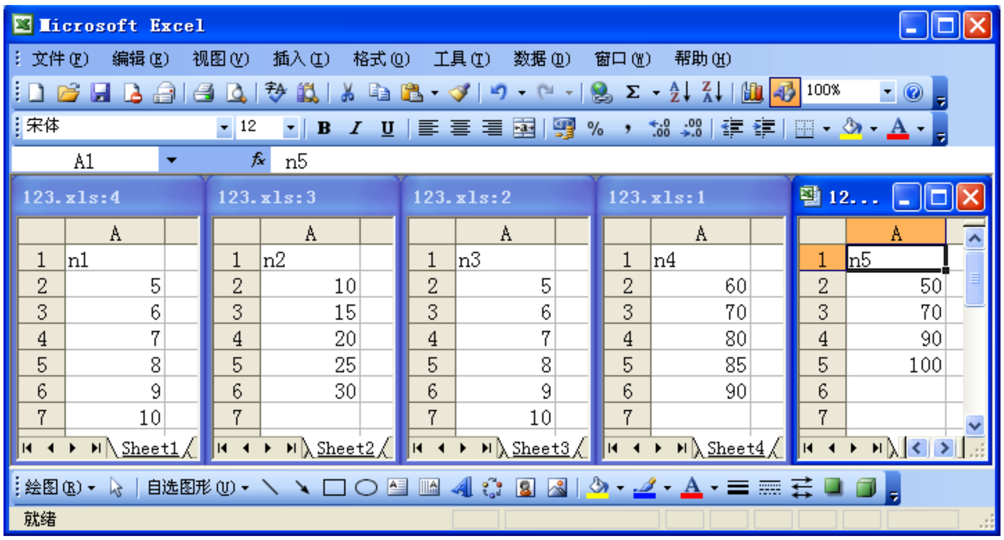

Fig. (7). Excel file to generate experimental plans.

"tasksequencequeue" node and wait in the queue. Second, the method of calculating the queuing time should be considered. As shown in Fig. (**6**), tasksequence0 (tq0) is under processing. The queuing time of tasksequence1 is the time represented by the yellow area in the same row and the red area represents processing time. In order to be more striking, the blue areas are listed separately representing the queuing time. It is not difficult to see that the total queuing time $T_w = (t_6 + t_{10} + t_{12}) - (t_3 + t_7 + t_8)$. So when a task sequence begins, if the "activetasksequence" node is not empty then add up system time to "start time"; when a task sequence ends, if the "activetasksequence" node is not empty then add up system time to "end time"; in the end, upon termination of the model the queuing time equals to "end time" minus "start time".

Based on the above analysis, if the "activetasksequence" node is not empty, "start time" should be added up on receiving task sequence, while "end time" on unloading. Ultimately the queuing time could be calculated.

Now, all the dependent variables are set, which could be obtained from global tables by calling and calculating.

5.2. Generate Experimental Plans

As already mentioned, 3600 plans could be generated in the model, which were impossible by manual methods. However, Microsoft Query can help. As shown in Fig. (**7**), create an excel file first, and record the parameters in different sheets. Run Microsoft Query; choose the right tables and columns. Then 3600 kinds of programs can be seen after pressing the "OK" button.

All the dependent and independent variables were input to the experimenter, and 2860 sets of valid results were got after calculation.

Here, take m_6 as an example. A multivariate linear regression was performed using SPSS, and an available and very significant regression model was then obtained.

$$m_6 = 20.232 + 0.656n_1 + 0.509n_2 - 0.353n_3 - 0.071n_4 - 0.045n_5$$

It can be seen that the time of queuing has positive correlations with the number of floors and the number of bays, and negative correlations with the others. n_4 and n_5 played a small role in the model. Notably, the time of queuing decreased when the number of levels increased. Interested scholars can verify the result in practice.

2860 sets of experimental data have very high value in use. The multivariate linear regression is just a simple try, and more valuable conclusions can be obtained in further data mining using these data.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] Y. Yu, and R. De Koster, "Optimal zone boundaries for two-class-based compact three-dimensional automated storage and retrieval systems", *IIE Transactions (Institute of Industrial Engineers)*, vol. 41, no.3, pp. 194-208, 2009.

[2] T. Lerther, M. Sraml, I. Potrc, and T. Tollazzi, "Travel time models for double-deep automated storage and retrieval systems", *International Journal of Production Research*, vol. 48, no.11, pp. 3151-3172, 2010.

[3] J. A. Tompkins, J. A. White, and Y. A. Bozer. "Tanchoco Facilities Planning," 3rd ed., John Wiley & Sons. 2003.

[4] Q. Zhang, "Research on Travel Time Model of the Stacker for Deep-lane Rack on Class-based Storage Assignment," Huazhong University of Science & Technology, 2012.

[5] W. Huang, "Travel Time Model of the Deep-lane Rack with Dual-shuttle Crane," Huazhong University of Science & Technology, 2013.

[6] M. Beaverstock, A. Greenwood, E. Lavery, and W. Nordgren, "Applied Simulation Modeling and Analysis using Flex Sim," 3rd ed., 2012.

Received: June 10, 2015 Revised: July 29, 2015 Accepted: August 15, 2015