Warehouse scheduling performance analysis considering LHRL

Wen-Qiang Yang · Li Deng · Qun Niu ·
Min-Rui Fei

Received: 3 April 2013 / Accepted: 7 April 2013 / Published online: 11 May 2013
© Shanghai University and Springer-Verlag Berlin Heidelberg 2013

Abstract Warehouse scheduling efficiency has to do with the length-height ratio of location (LHRL) to some extent, which hasn't been well investigated until now. In this paper a mathematic model is built by analyzing the relation between the travel time of the stacker and LHRL. Meanwhile, warehouse scheduling strategy is studied combining with the project on the automatic production line of an enterprise, and a warehouse scheduling strategy is proposed based on index of quality (IoQ) parameters. Besides, the process of getting the value of IoQ is also simplified with the idea of sparse matrix. Finally, the IoQ scheduling strategy is compared with random strategy and First Come First Out strategy in different LHRLs. The simulation results show that the IoQ scheduling strategy not only improves the quality of the product effectively, but also improves the efficiency of the scheduling substantially.

Keywords Warehouse scheduling · Performance analysis · Index of quality (IoQ) · Length-height ratio of location (LHRL) · Sparse matrix

1 Introduction

With the increasing competition of market economies, many companies are pursuing higher levels of production automation in manufacturing industry. For example, the automated warehouses are employed and play a significant

role in the field of manufacturing and processing [1, 2]. Thus it is meaningful to investigate the automated warehouses scheduling. However, it has not been considered so far that how the LHRL affects the warehouses scheduling efficiency [3].

This problem is discussed with one practical application in a certain enterprise. As the storage time of the products of the enterprise is required strictly for a certain time in the process, it can be classified as the production model of the just in time (JIT). This requires a reasonable scheduling for the semi-finished products in the warehouse in order to ensure its quality, which plays a very important role in reducing the cost.

According to the related literature, this type of scheduling can be summarized into two categories [4]. One is the stochastic scheduling, which has no model. That is to say, it chooses an object randomly and there is no optimization at all. Therefore, there has little research value on it. The other scheduling is based on model, which produces the optimal solution using the optimization algorithm. Generally, the characteristics of the optimal solution show the shortest path, the least time, the highest utilization of equipments and so on. There has been a great number of literatures on the scheduling of the workshop warehouse up to now. However, some researchers just use intelligent algorithms [5–12], such as the genetic algorithm, the particle swarm algorithm, the niche algorithm, and the neural network, to optimize the path. The path denotes getting out and going into the storage of the products. Some combine the advantage of genetic algorithm and the advantage of simulated annealing to study the lot-sizing scheduling problems, which enhances the utilization of warehouse. Some study the forklift dispatch problem by statistical learning. The statistic information is collected by means of the sensor nodes, and the action of the forklift can be

W.-Q. Yang · L. Deng (✉) · Q. Niu · M.-R. Fei
Shanghai Key Laboratory of Power Station Automation
Technology, School of Mechatronics Engineering and
Automation, Shanghai University, Shanghai 200072,
People's Republic of China
e-mail: dengli@shu.edu.cn

determined ahead of time. Hence it can improve the production efficiency.

However, there are few literatures about the strict control of the storage time of product in the warehouse at present. Sometimes, it may produce many task requests which are about getting out of warehouse at a time in view of the randomness of the production process. Thus, how the semi-finished products are scheduled reasonably without affecting the quality makes it a great challenge. Because the traditional static scheduling methods have been unable to meet such requirements, a warehouse scheduling system is modeled based on the weight timeout value of the products which marks the influence degree of the product quality.

2 Illustration of warehouse layout and LHRL

The warehouse system of the enterprise is composed of six parts [13]. They are six rows racks, a stacker, two round rails, a pipeline, three input buffers, and three output buffers respectively. Its working principle is that the stacker makes the corresponding action based on the instruction from scheduling system, which may be described that the stacker goes to the input buffer along the rail and takes the semi-products and put them into the appointed locations, then may pick semi-products required and put them into the specified output buffer. The warehouse system layout is shown in Fig. 1.

The length l and the height h of warehouse location are shown in Fig. 2, and the LHRL R is expressed as follows,

$$R = \frac{l}{h}. \quad (1)$$

Equation (1) is used to indicate the shape of the location. Supposing the horizontal velocity and the vertical velocity of the stacker are v_x and v_y respectively and the location perimeter L is fixed. If there is a semi-product which will be picked at the position P and the P coordinate is (m, n) in

Fig. 1 The warehouse system layout

Fig. 2 The structure of rack

Fig. 2. Then the time t which the stacker spend on the task is as follows [14],

$$t = 2\max\{((R/(1+R))Lm)/v_x, (1/(1+R))Ln/v_y\}. \quad (2)$$

Equation (2) illustrates that the LHRL change causes the time of stacker spending change for the same tasks. Consequently, the warehouse scheduling efficiency is also influenced by it.

3 Description of warehouse scheduling problem and structure of model

There exists some semi-finished products which have different processes on automation production line, thus they each have different storage time in the warehouse. At the same time, the random of the production is regarded. Therefore, it may cause that those semi-finished products which come out of warehouse at a time compete for the stacker, and then the storage time of the semi-finished products is longer than the normal time to some degree. This problem is even more serious in the case that there are a large number of semi-finished products [15–17].

In order to quantize the problem, the definitions are as follows.

Definition 1 The unit timeout value which affects the degree of the product quality is represented as weight. Depending on the different storage time of product in the warehouse, the weight is defined as follows

$$w = \begin{cases} 1/2 & \text{Storage time is 2 d,} \\ 1/3 & \text{Storage time is 4 d,} \\ 1/5 & \text{Storage time is 5 d,} \\ 1/8 & \text{Storage time is 10 d.} \end{cases} \quad (3)$$

Definition 2 According to the locations of the semi-finished products and their storage time in the warehouse, a buffer is opened up in the RAM, in which the storage time matrix $S_{m \times n}$ and the weight matrix $W_{m \times n}$ are generated

dynamically corresponding to the semi-finished product in the warehouse, they are defined as

$$S_{m \times n} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{bmatrix}, \quad (4)$$

$$W_{m \times n} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix}. \quad (5)$$

Definition 3 The past time matrix $P_{m \times n}$ indicates the time pass by the semi-finished products since they have been stored in the warehouse, which is defined as

$$P_{m \times n} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix}. \quad (6)$$

The scheduling model is described as,

$$I_i = \max\{(p_i - s_i)w_i\}, \quad i \in (1, 2, 3, \dots, m \times n), \quad (7)$$

The scheduling will be operated in descending order of I_i , the scheduling sequence is showed as,

$$I_{il}, I_{jl}, I_{kl}, \dots, I_{zl} \quad (8)$$

where $il, jl, kl, \dots, zl \in (1, 2, 3, \dots, m \times n)$.

As it always takes the stacker a certain amount of time to carry the semi-finished product out of the warehouse, this scheduling sequence (8) shows a local optimum rather than a global optimum, which is proved as follows.

Supposing

$$I_i > I_j, \quad (9)$$

the inequality (9) is also substituted as

$$(p_i - s_i)w_i > (p_j - s_j)w_j. \quad (10)$$

Then Δt , I_i and I_j can be rewritten as

$$I_i = ((p_i - s_i) + \Delta t)w_i, \quad (11)$$

$$I_j = ((p_j - s_j) + \Delta t)w_j. \quad (12)$$

The relationship between I_i and I_j is

$$\begin{cases} I_i > I_j, & \text{when } w_i > w_j, \\ I_i > I_j \text{ or } I_i < I_j \text{ or } I_i = I_j, & \text{when } w_i < w_j. \end{cases} \quad (13)$$

Therefore, scheduling sequence (8) is not a global optimum.

As proved above, this scheduling model is time-varying. Therefore, the scheduling sequence needs to be updated dynamically all the time to achieve a global optimum.

4 Utilization of sparse matrix in warehouse scheduling

The matrix $D_{m \times n}$ is defined as the difference between $P_{m \times n}$ and $S_{m \times n}$, in which the element d_i indicates the timeout corresponding to the location in the warehouse. That's to say, it can be represented as

$$D_{m \times n} = P_{m \times n} - S_{m \times n}. \quad (14)$$

Equation (7) can also be expressed as

$$I_i = \max\{d_i w_i\}, \quad i \in (1, 2, 3, \dots, m \times n). \quad (15)$$

According to the objective function (15), the semi-finished product in the warehouse can be found. If the $m \times n$ elements which are gone through in a matrix are calculated and sorted, the number of comparisons is expressed as

$$T(m \times n) = \begin{cases} 0, & m \times n = 1, \\ 2T(\frac{m \times n}{2}) + ((m \times n) - 1), & m \times n \neq 1. \end{cases} \quad (16)$$

The time complexity can be expressed as $O((m \times n) \times \log(m \times n))$ from the above formula.

The above method has little effect on the search efficiency when the size of a warehouse is small. Otherwise, it will have a huge influence on the search efficiency. In view of this situation, the rules are defined as follows.

Given a situation that the number of the semi-finished products whose storage time expires is a small part of the whole product in the warehouse in a short period of time, based on which the matrix $D_{m \times n}$ is transformed as follows.

If there are some positions in which there are no semi-finished products or the storage time of semi-finished products don't expire or semi-finished products has been placed the scheduling sequence, the elements of the matrix $D_{m \times n}$ are assigned to zero corresponding to such positions.

The number of zero elements is much larger than the number of non-zero elements by means of such transformation in the matrix $D_{m \times n}$, so it becomes a sparse matrix, in which only non-zero elements of the matrix are calculated. Then it will improve the calculation and scheduling efficiency greatly.

5 Design of scheduling algorithm based on IoQ strategy

The principle of the scheduling algorithm based on the IoQ parameter is that the IoQs are computed for semi-finished products whose storage time has expired, and the one whose IoQ is the largest scheduled first. With an eye to the different weights and the changing timeout value for such semi-finished products, the IoQs of the rest semi-finished products of the warehouse also keep varying. It is needed to calculate the IoQs to achieve the optimal scheduling on

line all the time. The specific scheduling steps are as follows.

Step 1 The mapping two-dimension array $\mathbf{a}$ is maintained in RAM according to the size of the warehouse in the actual production. The elements of the table are structure variables which are associated with the semi-finished product. Those variables include the following members:

- (i) the warehouse location number of the semi-finished product,
- (ii) the storage time of the semi-finished product,
- (iii) the weight of the semi-finished product corresponding to the current storage time,
- (iv) the start time of the semi-finished product in the warehouse,
- (v) the flag which marks whether the semi-finished product has been in the scheduling sequence.

Step 2 The $\mathbf{S}_{m \times n}$, $\mathbf{W}_{m \times n}$ and $\mathbf{P}_{m \times n}$ are generated from the mapping $\mathbf{a}$ at the current time. If the elements of $\mathbf{W}_{m \times n}$ are all zero, then goes to Step 1.

Step 3 The $\mathbf{D}_{m \times n}$ is worked out by Eq. (14), meanwhile, the matrix $\mathbf{D}_{m \times n}$ is transformed to the sparse matrix by the rules of this paper. Then the two-dimension array $\mathbf{b}$ is generated whose elements correspond to the non-zero in the sparse matrix $\mathbf{D}_{m \times n}$.

Step 4 The IoQs of the semi-finished products are calculated in $\mathbf{b}$ according to Eq. (15) and the maximum value among IoQs is placed on the tail of the scheduling sequence. The flag of the semi-finished product, which identify whether it has been in the scheduling sequence, is set as the number one.

Step 5 Under field conditions, the time which the stacker spends to carry the semi-finished product out of the warehouse passes, then go to Step 1. The following semi-finished product in the scheduling sequence is obtained.

6 Experimental verification and analysis

The number of the semi-finished product and their timeout values are assumed in the warehouse, and the following agreement is done.

The perimeter of the location is 3 m. v_x and v_y are 1 m/s, 0.5 m/s respectively. The products quality or scheduling performance is marked as SoP, which is the sum of the product of the timeout and the weight for every task. The semi-finished products whose timeout value is 10 s is marked as A; in the same way, the semi-finished products whose timeout value is 20 s is marked as B; the semi-finished products whose timeout value is 30 s is marked as C; the semi-finished products whose timeout value is 40 s is marked as D; the semi-finished products whose timeout value is 50 s is marked as E.

The proposed strategy is verified and compared with FCFO [18] strategy and random strategy respectively. On the condition whose platform is Windows XP, frequency is 1.49 GHz, memory size is 1.99 GB, development environment is VC++ 6.0.

Table 1 shows that the experiments have six tasks, and each task includes four subtasks. And the X_Y indicates that the number of timeout category which belongs to Y is X . At the same time, the coordinates of six tasks in the warehouse are shown in Fig. 3, of which Xyz means the product belongs to X , the task and the subtask of which are y and z respectively. Besides, different tasks are labeled with different colors. The numbers at the left side and at the top side of Fig. 3 are represented as layer numbers, column numbers respectively.

Table 2 states that the IoQ strategy is compared with the other two strategies with six tasks respectively under different LHRLs.

From Table 2, it can be seen that the varied LHRL has a great influence on scheduling performance for scheduling strategies. At the same time, the simulation results reveal that the LHRLs corresponding to minimum SoP of different scheduling strategies are different. For example, for the Task 1, the LHRLs corresponding to the minimum SoPs are 1, 0.75 and 0.5 for IoQ, FCFO and RANDOM respectively. Therefore, to get better scheduling performance, it is essential to find a suitable LHRL for different scheduling strategies. Furthermore, it can be observed that the IoQ strategy outperforms the other two scheduling strategies considering all the SoP values, and FCFO is more or less than RANDOM. In other words, the IoQ strategy compared with the others has the smallest impact on the quality of the products.

In order to highlight the advantage of the IoQ relative to the others. The evaluation metric is defined as

$$Q(a, b) = \frac{b - a}{b} \times 100\%, \quad (17)$$

where a and b represent the SoP values of different scheduling strategies. The $Q(a, b)$ indicates a is the improvement degree of product quality relative to b . To get

Table 1 Tasks of the warehouse on different cases

Task No.	Subtask 1	Subtask 2	Subtask 3	Subtask 4
1	(3 _A , 2 _B , 3 _C , 4 _D)	(3 _A , 3 _C , 5 _D)	(3 _A , 1 _B , 4 _D)	(3 _B , 3 _C , 1 _D)
2	(4 _A , 2 _C , 3 _D)	(3 _B , 2 _D , 2 _E)	(5 _A , 3 _B , 2 _C)	(3 _A , 1 _C , 3 _E)
3	(2 _A , 3 _C , 1 _D)	(1 _B , 4 _D , 3 _E)	(2 _A , 2 _B , 5 _C , 2 _D)	(4 _B , 3 _E)
4	(4 _B , 2 _C , 1 _E)	(3 _A , 3 _C , 2 _D)	(1 _C , 4 _D , 2 _E)	(4 _A , 4 _B , 2 _E)
5	(4 _D , 4 _E)	(2 _A , 3 _B , 3 _C , 3 _D)	(1 _C , 4 _D , 2 _E)	(2 _A , 2 _B , 2 _C , 5 _D , 3 _E)
6	(5 _A , 2 _B , 3 _E)	(2 _B , 3 _D)	(3 _A , 1 _B , 3 _C)	(5 _C , 2 _D , 4 _E)

Fig. 3 Coordinates of six tasks in the warehouse

Table 2 SoP values comparison of the warehouse scheduling strategies under different R

R	Strategy	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
0.25	IoQ	5,339.49	3,715.45	3,534.84	4,897.63	6,323.70	3,446.13
	RANDOM	7,457.86	5,036.54	4,898.19	5,512.27	7,353.74	4,963.57
	FCFO	7,148.14	4,855.77	5,013.99	5,870.57	10,132.90	4,367.88
0.5	IoQ	4,686.89	3,477.96	3,616.93	4,278.97	5,641.45	3,133.47
	RANDOM	6,714.64	4,726.68	5,134.24	5,009.32	6,673.65	4,577.09
	FCFO	6,457.29	4,515.08	5,052.83	5,502.74	8,970.72	4,060.63
0.75	IoQ	4,431.15	3,578.27	4,103.25	4,115.88	5,527.32	3,046.68
	RANDOM	6,851.14	4,888.66	5,662.85	4,870.09	6,559.60	4,457.97
	FCFO	6,425.61	4,568.35	5,440.13	5,454.45	8,375.91	3,992.22
1	IoQ	4,369.05	3,759.44	4,638.74	4,173.19	5,625.57	3,135.06
	RANDOM	7,136.62	5,087.00	6,287.79	4,906.56	6,638.72	4,485.77
	FCFO	6,541.90	4,714.18	5,996.13	5,559.38	8,119.87	4,098.27
1.25	IoQ	4,399.07	3,951.80	5,091.13	4,329.70	5,830.00	3,312.09
	RANDOM	7,541.12	5,259.84	6,787.70	5,049.99	6,833.64	4,611.28
	FCFO	6,796.21	4,843.47	6,461.01	5,755.84	8,126.75	4,283.14
1.5	IoQ	4,442.43	4,110.35	5,443.60	4,521.58	6,072.75	3,517.65
	RANDOM	7,891.44	5,417.92	7,191.00	5,223.33	7,080.57	4,802.33
	FCFO	7,017.90	4,961.58	6,851.71	5,982.68	8,214.86	4,531.47
1.75	IoQ	4,537.48	4,275.35	5,735.38	4,701.23	6,294.07	3,694.73
	RANDOM	8,229.22	5,569.72	7,522.49	5,373.89	7,300.59	4,996.92
	FCFO	7,259.61	5,075.06	7,179.92	6,187.16	8,324.92	4,749.50
2	IoQ	4,649.88	4,403.07	5,977.33	4,872.15	6,484.42	3,845.95
	RANDOM	8,522.77	5,708.23	7,798.73	5,502.10	7,493.30	5,167.68
	FCFO	7,489.00	5,182.12	7,453.43	6,370.82	8,431.57	4,941.55

clear results of the evaluation, the results can be shown in Fig. 4.

Figure 4 illustrates that the IoQ strategy improve obviously the scheduling performance compared with the other two strategies under different R . Furthermore, the improvement degree varies as the R changes, and the best

improvement ratio even reaches 45.4 % and 37.9 % respectively relative to RANDOM and FCFO. Meanwhile, the LHRL values R are different when the metric $Q(a, b)$ gets the peak value for every tasks, and the reason of which is that the timeout weights are various for every subtasks in every tasks, which implies that there is a suitable R for every task and every

Fig. 4 Product quality improve ratio for IoQ relative to FCFO and random strategies in six tasks

scheduling strategy. What's more, it can be deduced that the IoQ strategy is better than the other two no matter what the R is. Therefore, it is also proved that the proposed IoQ strategy is more effective for improving scheduling performance considering the changes of R . In a word, this paper provides a sound theoretical base for the design of the automated warehouse.

7 Conclusions

This paper investigated how the LHRL affects the scheduling performance. Meanwhile, the IoQ mathematical model is built, which is based on the idea that the extent of the unit timeout value which has influence on the product quality. Besides, the scheduling strategy with IoQ is designed. The experiment is performed by scheduling the semi-finished products with different storage time in the warehouse. Comparing with warehouse scheduling strategies, the results show that R has different effect on scheduling performance for different scheduling strategies, and there is a proper R for the best scheduling performance of each scheduling strategy. Additionally, the proposed strategy greatly improves the scheduling performance compared with the other two R changing conditions.

Acknowledgments This work is supported by the National Natural Science Foundation of China (Grant Nos. 61074032 and 61273040), the Project of Science and Technology Commission of Shanghai Municipality (Grant No. 10JC1405000), and the Shanghai Rising-Star Program (Grant No. 12QA1401100).

References

- Hachemi K, Sari Z, Ghoulali N (2012) A step-by-step dual cycle sequencing method for unit-load automated storage and retrieval systems. *Comput Ind Eng* 63(4):980–984
- Ekren BY, Heragu SS, Krishnamurthy A et al (2013) An approximate solution for semi-open queueing network model of an autonomous vehicle storage and retrieval system. *IEEE Trans Autom Sci Eng* 10(1):205–215
- Gagliardi JP, Renaud J, Ruiz A (2012) Models for automated storage and retrieval systems: a literature review. *Int J Prod Res* 50(24):7110–7125
- Hu G, Kise H, Xu Y (2005) Optimization for input/output scheduling for automated warehouses. *Trans Inst Syst Control Inf Eng* 18(4):156–163
- Chan FTS, Kumar V (2009) Hybrid TSSA algorithm-based approach to solve warehouse-scheduling problems. *Int J Prod Res* 47(4):919–940
- de René K, Tho LD, Kees JR (2007) Design and control of warehouse order picking: a literature review. *Eur J Oper Res* 182(2):481–501
- Basile F, Chiacchio P, Coppola J (2012) A hybrid model of complex automated warehouse systems-part i: modeling and simulation. *IEEE Trans Autom Sci Eng* 9(4):640–653
- Estanjini RM, Lin YW, Li KY et al (2011) Optimizing warehouse forklift dispatching using a sensor network and stochastic learning. *IEEE Trans Ind Inf* 7(3):476–486
- Luo J, Wu CQ, Li B et al (2011) Modeling and optimization of RGV system based on improved QPSO. *Comput Integr Manuf Syst* 17(2):321–328
- Yu MF, Koster RD (2010) Enhancing performance in order picking processes by dynamic storage systems. *Int J Prod Res* 48:4785–4806
- Liu J, Wu CQ, Li MJ, Chen XB (2009) Order picking problem based on ant colony algorithm. *Syst Eng Theory Pract* 29(3):179–185

12. Li MJ, Chen XB, Zhang MF (2007) Path planning based on the swarm intelligence algorithm. *J Tsinghua Univ (Sci & Tech)* 47(z2):1770–1773
13. Chen L, Langevin A, Riopel D (2011) Dynamic relocation problem in an automated storage/retrieval system. *J Shanghai Jiaotong Univ* 45(1):115–119
14. Lerher T, Sraml M, Potrc I et al (2010) Travel time models for double-deep automated storage and retrieval systems. *Int J Prod Res* 48(11):3151–3172
15. Chang FL, Liu ZX, Xin Z et al (2007) Research on order picking optimization problem of automated warehouse. *Syst Eng Theory Pract* 2:139–143
16. Shi N (2010) K constrained shortest path problem. *IEEE Trans Autom Sci Eng* 7(1):15–23
17. Liu SN, Ke YL, Li JX et al (2006) Optimization for automated warehouse based on scheduling policy. *Comput Integr Manuf Syst* 12(9):1438–1443
18. Vandeveld G, Gademann N, Vandeveld S (2005) Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE Trans* 37(1):63–75