# Improvement of Modi-InDeep Storage in Flow-rack Automated Storage and Retrieval Systems

Min WANG [a], Zhu-Xi CHEN [b,*]

College of Information Engineering, Yangzhou University, 225000, Yangzhou, China

[a]mz160519@yzu.edu.cn, [b]zxchen@yzu.edu.cn

* Corresponding author

**Keywords:** Flow-rack AS/RS, Storage assignment, Heuristic.

**Abstract.** Flow-rack automated storage and retrieval system (AS/RS) has very high storage density, in which each storage unit (bin) can simultaneously store multiple unit-loads in depth. Based on the turnover information, Modi-InDeep heuristic can appropriately organize unit-loads in each bin of flow-rack to reduce blocking unit-loads. In this paper, two modifications are proposed to improve the performance of Modi-InDeep heuristic. Experimental results demonstrate that the designed modifications can save the total travel time of the storage machine and reduce blocking unit-loads.

## Introduction

Automated storage and retrieval systems (AS/RSs) [1] play a very important role in advanced manufacturing systems [2-3]. Recently, the increasing land cost promotes the implementation of high density AS/RSs, such as 3D-compact system [4], flow-rack AS/RS [5], mobile rack system [6], double-deep system [7], et al. In these systems, the flow-rack AS/RS has great practical application prospect since its low construction and operation costs.

In Fig.1, a flow-rack AS/RS consists of *L* columns and *H* rows of slope bins is demonstrated. In each bin, at most *M* unit-loads can be simultaneously stored (*M* layers). Since the decks of bins replaced by wheels or rollers, unit-loads are driven by the gravity to follow the first-in-first-out (FIFO) mode in each bin. A storage machine performs storage operations on the storage face and a retrieval machine conducts retrieval operations on the retrieval face. Occasionally, requested unit-load may be blocked by blocking unit-loads, which must be removed to the restoring conveyor. Blocking unit-loads slide to the storage face and are restored by the storage machine. Apparently, storage and retrieval machines spend extra operation costs for removing and restoring blocking unit-loads. Therefore, turnover information of unit-loads must be carefully considered to design effective method for the generation of storage and retrieval operations in a flow-rack AS/RS.

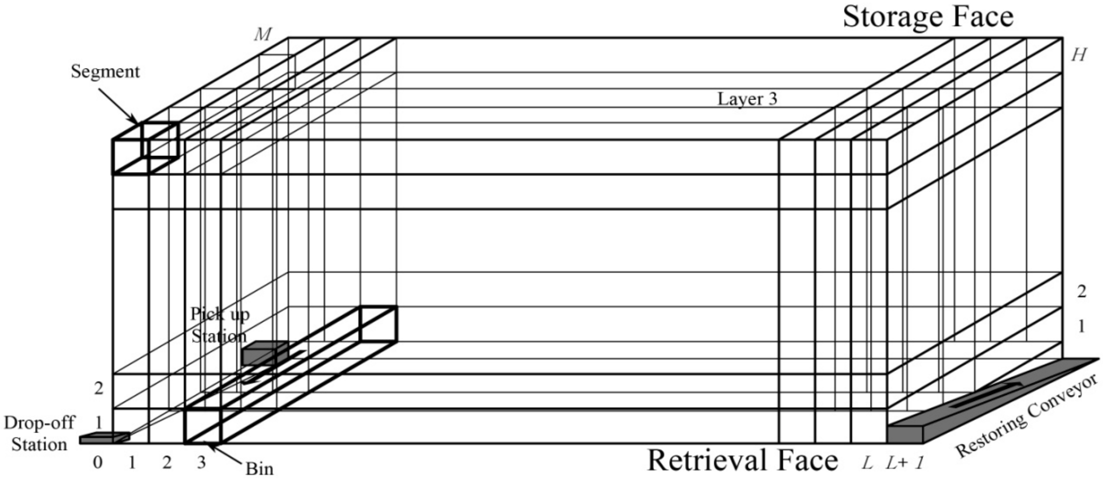

Fig.1 A flow-rack AS/RS with L column and H rows of bins and M layers

In the duration-of-stay (DOS) storage, the arrival and departure time of all items in a planning cycle are known in advance. Based on arrivals and departures, FIFO unit-loads can share easy accessible bins in a flow-rack AS/RS to not only save the travel times of storage and retrieval operations but also reduce blocking unit-loads [8]. For class-based storage, in in-deep storage stores fast-turnover unit-loads to front layers and slow-turnover unit-loads to behind layers of flow-rack to minimize blocking unit-loads [9]. However, in-deep storage assumes that 20% of fast-turnover unit-loads contribute 80% of turnover requests, which may constrain its application scale. Based on the impacts of class limit, a Modi-InDeep heuristic has been proposed [10], which assigns all unit-loads into the same class and stores as many unit-loads as possible to bins near the pick-up/drop-off station for saving storage/retrieval costs. Still, there is improvable room for the Modi-InDeep heuristic. In this paper, two improvements are proposed and the effectiveness of the two improvements are analyzed by simulation experiments.

The rest of this paper is as follow. The in-deep storage and Modi-InDeep heuristic are detailed in Section 2. The two proposed modifications are introduced in Section 3. Performance evaluation is demonstrated in Section 4 and followed by discussions in Section 5.

## In-deep Storage and Modi-InDeep heuristic

In a flow-rack AS/RS, the storage machine picks up a unit-load from the pick-up station, moves to a bin and stores the unit-load within a storage operation. As well, the retrieval machine removes blocking unit-loads to the restoring conveyor, retrieve an outgoing unit-load and return it to the drop-off station within a retrieval operation. Blocking unit-loads are restored by the storage machine. Apparently, both machines spend extra costs for removing/re-storing blocking unit-loads. Therefore, minimize blocking unit-loads is very important in a flow-rack AS/RS.

The class-based storage policy is widely used in AS/RS. As described in [11], a total of $N$ types of unit-loads with the references of 1, 2,..., $N$ are stored to and retrieved from the flow-rack. The demand of type $i$ unit-loads in a unit of time (turnover rate) is fixed and known, which is denoted as $D(i)$. It assumes that $D(1) >= D(2) >= ... >= D(N)$. Let $A = D(1) + D(2) +... + D(N)$ be the annual demand of the warehouse. According to the ABC-demand Curve, $G(i) = (i / N)^s$ $(0< s <=1)$ and $D(i) = A(G(i) - G(i\text{-}1))$. Thus, the turnover rates of different types of unit-loads can be estimated. In in-deep storage, the following principles apply for storage/retrieval operations using turnover information: 1) allocate the unit-load to bins based on the turnover rate, 2) the fast-turnover unit-loads should be stored in front layers to reduce the probability of blocking and 3) choose outgoing unit-loads closest to retrieval face.

Modi-Indeep heuristic is modified from the in-deep storage [10]. Modi-InDeep heuristic stores unit-loads to bins closest to the drop-off station if there is no other types of unit-loads before them. The unit-loads will be stored in bins closest to the restoring conveyor if there are other types of unit-loads stored in bins. In retrieval, Modi-Indeep heuristic selects unit-loads from bins closest to the drop-off station if there is no blocking unit-load. Otherwise, unit-loads from bins closest to the restoring conveyor if there are blocking unit-loads. Based on above mechanisms, the storage/retrieval cost can be reduced compared to in-deep storage.

## Two Improvements of Modi-Indeep

Although Modi-InDeep effectively reduces the quantity of blocking unit-loads, there is still room for improvement. In Modi-InDeep, the storage machine will return to the middle point of the storage face after finishing a storage operation [10]. This mechanism causes the storage machine has much higher workload than the retrieval machine does. Consequently, the first improvement, i.e., IMP1, is to let the storage machine stay in the position where the unit-load has been stored until it receives another storage or re-storage operation.

The second improvement, i.e., IMP2, mainly acts on the rules for selecting outgoing unit-loads. As shown in Fig.2, IMP2 modifies the Step 5, 9 and 11 in the retrieval algorithm of Modi-Indeep. IMP2 selects unit-loads with references smaller than the target unit-loads instead of the larger ones (Shown in gray boxes in Fig.2). Since unit-loads with small reference have higher turnover rate than unit-loads with large reference, a unit-load with small reference has higher probability to be retrieved before a unit-load with large reference. Therefore, retrieve unit-load from bin contains the most unit-loads with smaller reference (Step 9 and 11) can obviously reduce the blocking probability. As well, outgoing unit-load selected by Step 5 implies the target blocked by blocking unit-loads. In this case, restoring fast-turnover unit-loads as blocking unit-loads may let them be stored together and reduce blocking unit-loads in future retrieval operations.

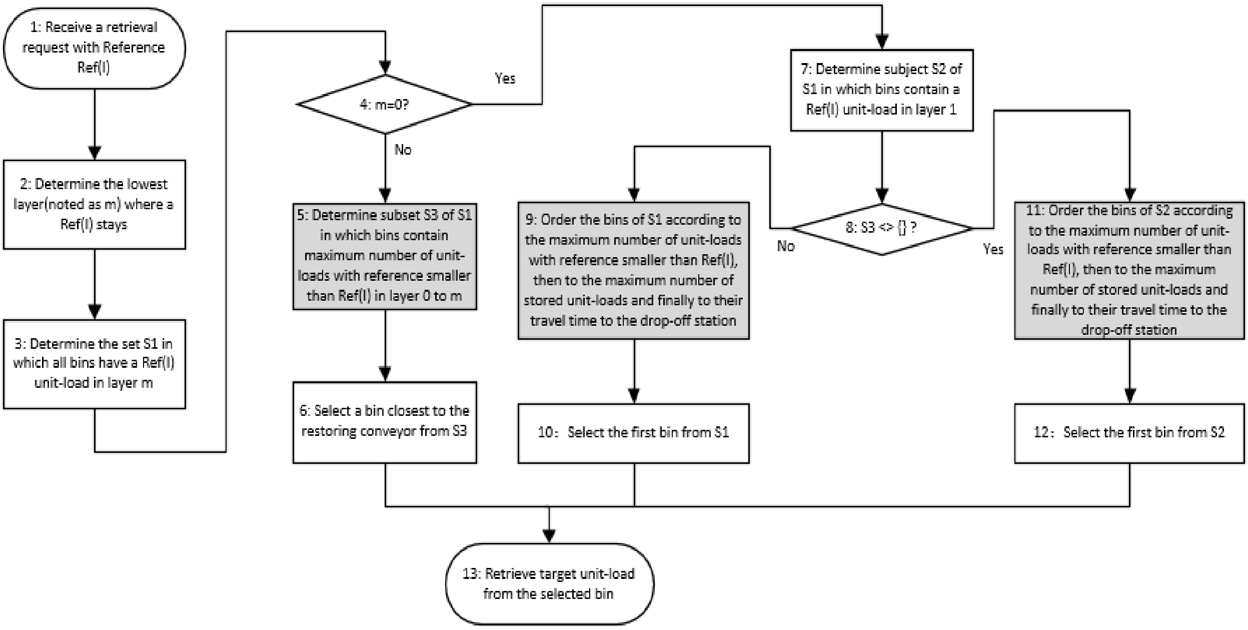
1: Receive a retrieval request with Reference Ref(I)
2: Determine the lowest layer(noted as m) where a Ref(I) stays
3: Determine the set S1 in which all bins have a Ref(I) unit-load in layer m
4: m=0?
Yes
No
5: Determine subset S3 of S1 in which bins contain maximum number of unit-loads with reference smaller than Ref(I) in layer 0 to m
6: Select a bin closest to the restoring conveyor from S3
7: Determine subject S2 of S1 in which bins contain a Ref(I) unit-load in layer 1
8: S3 <> {} ?
No
Yes
9: Order the bins of S1 according to the maximum number of unit-loads with reference smaller than Ref(I), then to the maximum number of stored unit-loads and finally to their travel time to the drop-off station
10: Select the first bin from S1
11: Order the bins of S2 according to the maximum number of unit-loads with reference smaller than Ref(I), then to the maximum number of stored unit-loads and finally to their travel time to the drop-off station
12: Select the first bin from S2
13: Retrieve target unit-load from the selected bin

Fig.2 The IMP2 to the retrieval of Modi-Indeep

## Experiments and Performance

In simulation experiments, $L$ = 20, $H$ = 10 and $M$ = 10. Let $t_h$ = 1.0s and $t_v$ = 2.0s be the travel times between two horizontally and vertically adjacent bins, respectively. As well, $N$ = 100 and $s$ takes values of 0.065,0.139,0.222,0.317 and 0.431 to make the first 20% fast-turnover unit-loads contribute 90%, 80%, 70%, 60% and 50% turnover requests, respectively.

In each instance, 1600 warm-up unit-loads are stored firstly. Then, a sequence of requests contains 2000 storage requests and 2000 retrieval requests is generated. Three modes are denoted as RAN (storage/retrieval requests are randomly generated), 1By1 (a type *i* outgoing unit-load followed by a type *i* incoming unit-load in the sequence of requests), and 10By10 (ten retrieval requests followed by ten storage requests with the same types in the sequence of requests). The turnover rates of warmup unit-loads and requests follow the ABC-demand curve [11]. For each experiment configuration, the average experimental results are achieved by 100 generated instances. Let $s_{total}$ be the total travel time of the storage machine and $r_{total}$ be the total travel time of the retrieval machine. Let ST = $s_{total}$/2000) and RT = $r_{total}$/2000 be the performance measures of storage and retrieval. ARPD (Average Relative Percentage Deviation) shows the effectiveness.

## Performance of IMP1

Tab.1 demonstrates the values of ST, ARPD and 95% confident intervals in RAN, 1By1 and 10By10 modes obtained by the IMP1. As shown in Tab.1, IMP1 can significantly reduce the value of ST since the extra travel time for returning to the middle point of the storage face is saved by IMP1. The experiment results, i.e., values of ARPD, illustrate that the total travel time of storage machine can be reduced 13%~50% by IMP1. With the increase of $s$, the randomness of unit-load types increase, which leads the reduction of the effectiveness of IMP1.

Tab.1 The effectiveness of IMP1

| | Random | | | | 1by1 | | | | 10by10 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | original | | IMP1 | | original | | IMP1 | | original | | IMP1 | |
| s | ST | ARPD | ST | ARPD | ST | ARPD | ST | ARPD | ST | ARPD | ST | ARPD |
| 0.065 | 23.06 | 102.81 | 11.37 | 0.00 | 22.33 | 149.61 | 8.91 | 0.00 | 23.26 | 80.59 | 12.88 | 0.00 |
| | [22.97,23.15] | | [11.12,11.62] | | [22.27,22.39] | | [8.78,9.03] | | [23.19,23.34] | | [12.75,13.01] | |
| 0.139 | 24.81 | 56.93 | 15.81 | 0.00 | 24.22 | 69.96 | 14.25 | 0.00 | 24.85 | 47.48 | 16.85 | 0.00 |
| | [24.68,24.94] | | [15.63,16.00] | | [24.17,24.27] | | [14.12,14.38] | | [24.78,24.91] | | [16.74,16.97] | |
| 0.222 | 26.09 | 35.11 | 19.31 | 0.00 | 25.71 | 48.91 | 18.25 | 0.00 | 26.03 | 33.42 | 19.51 | 0.00 |
| | [25.98,26.19] | | [19.07,19.56] | | [25.63,25.78] | | [18.08,18.42] | | [25.97,26.09] | | [19.35,19.66] | |
| 0.317 | 27.21 | 27.54 | 21.57 | 0.00 | 26.98 | | 21.08 | 0.00 | 27.16 | 25.28 | 21.74 | 0.00 |
| | [27.13,27.28] | | [21.41,21.73] | | [26.91,27.05] | | [20.94,21.21] | | [27.08,27.25] | | [21.60,21.88] | |
| 0.431 | 28.04 | 20.50 | 23.27 | 0.00 | 27.85 | 27.99 | 22.99 | 0.00 | 27.97 | 20.72 | 23.17 | 0.00 |
| | [27.98,28.10] | | [23.16,23.37] | | [27.79,27.90] | | [22.89,23.08] | | [27.92,28.01] | | [23.08,23.27] | |

## 4.2 Performance of IMP2

Because of the effectiveness of IMP1, the storage machine will not return to the middle point of the storage face in the following experiments.

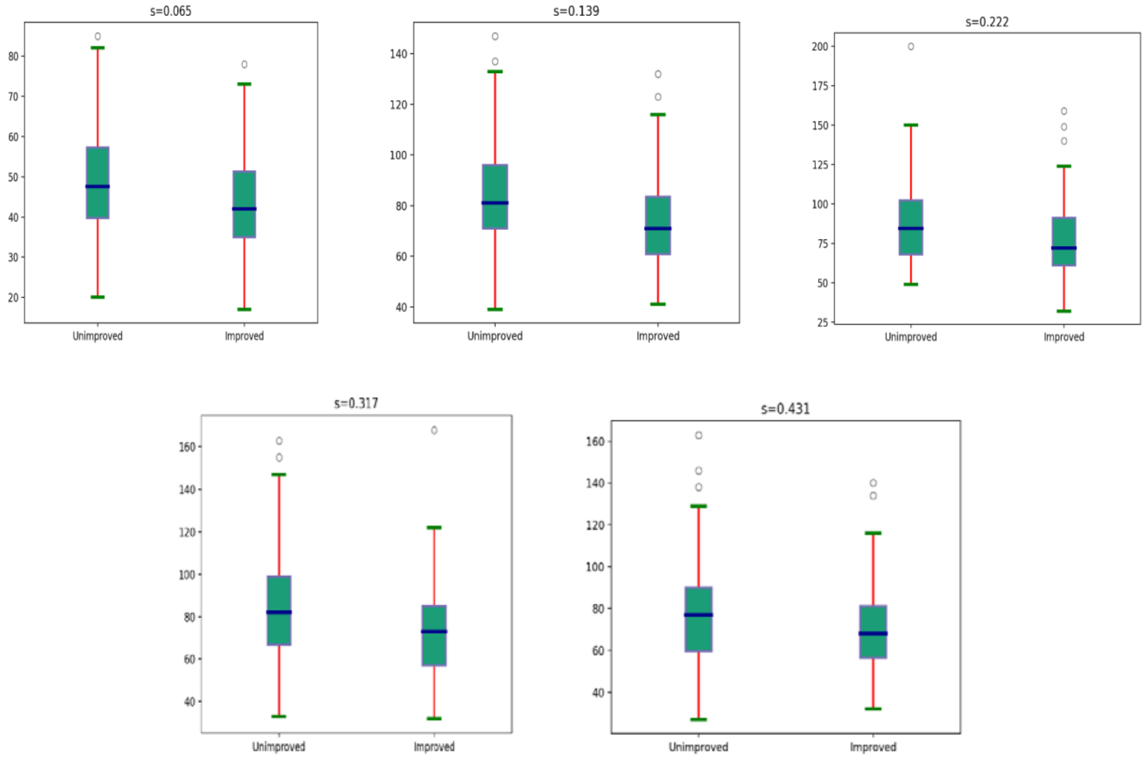

Fig.3 Numbers of blocking unit-loads obtained by Unimproved and IMP2

Tab.2 Values of *ST* obtained by Unimproved and IMP2

| | Random | | | | 1by1 | | | | 10by10 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Unimproved | | IMP2 | | Unimproved | | IMP2 | | Unimproved | | IMP2 | |
| s | ST | ARPD | ST | ARPD | ST | ARPD | ST | ARPD | ST | ARPD | ST | ARPD |
| 0.065 | 11.37 | 4.03 | 10.93 | 0.00 | 8.91 | 6.96 | 8.33 | 0.00 | 12.88 | 5.66 | 12.19 | 0.00 |
| | [11.12,11.62] | | [10.59,11.26] | | [22.27,22.39] | | [8.20,8.45] | | [12.75,13.01] | | [12.08,12.30] | |
| 0.139 | 15.81 | 5.61 | 14.97 | 0.00 | 14.25 | 7.22 | 13.29 | 0.00 | 16.85 | 7.19 | 15.72 | 0.00 |
| | [15.63,16.00] | | [14.65,15.29] | | [24.17,24.27] | | [13.14,13.43] | | [16.74,16.97] | | [15.59,15.84] | |
| 0.222 | 19.31 | 6.92 | 18.06 | 0.00 | 18.25 | 6.54 | 17.13 | 0.00 | 19.51 | 5.75 | 18.45 | 0.00 |
| | [19.07,19.56] | | [17.79,18.34] | | [25.63,25.78] | | [16.93,17.34] | | [19.35,19.66] | | [18.31,18.59] | |
| 0.317 | 21.57 | 5.84 | 20.38 | 0.00 | 21.08 | 5.61 | 19.96 | 0.00 | 21.74 | 5.43 | 20.62 | 0.00 |
| | [21.41,21.73] | | [20.19,20.56] | | [26.91,27.05] | | [19.81,20.12] | | [21.60,21.88] | | [20.25,20.78] | |
| 0.431 | 23.27 | 4.77 | 22.21 | 0.00 | 22.99 | 4.41 | 22.02 | 0.00 | 23.17 | 4.84 | 22.10 | 0.00 |
| | [23.16,23.37] | | [22.09,22.32] | | [27.79,27.90] | | [21.89,22.15] | | [23.08,23.27] | | [21.99,22.21] | |

Tab.3 Values of *RT* obtained by Unimproved and IMP2

| | Random | | | | 1by1 | | | | 10by10 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Unimproved | | IMP2 | | Unimproved | | IMP2 | | Unimproved | | IMP2 | |
| s | RT | ARPD | RT | ARPD | RT | ARPD | RT | ARPD | RT | ARPD | RT | ARPD |
| 0.065 | 11.03 | 3.57 | 10.64 | 0.00 | 8.46 | 5.88 | 7.99 | 0.00 | 12.50 | 5.21 | 11.88 | 0.00 |
| | [10.81,11.27] | | [10.29,10.98] | | [8.33,8.59] | | [7.86,8.11] | | [12.36,12.65] | | [11.77,11.99] | |
| 0.139 | 15.58 | 6.35 | 14.65 | 0.00 | 13.66 | 6.22 | 12.86 | 0.00 | 16.30 | 6.33 | 15.33 | 0.00 |
| | [15.26,15.90] | | [14.31,14.98] | | [13.55,13.78] | | [12.71,13.01] | | [16.17,16.42] | | [15.21,15.45] | |
| 0.222 | 18.52 | 6.07 | 17.46 | 0.00 | 17.44 | 5.51 | 16.53 | 0.00 | 18.73 | 4.87 | 17.86 | 0.00 |
| | [18.30,18.74] | | [17.17,17.74] | | [17.29,17.60] | | [16.32,16.73] | | [18.62,18.84] | | [17.71,18.00] | |
| 0.317 | 20.48 | 5.95 | 19.33 | 0.00 | 19.93 | 5.38 | 18.93 | 0.00 | 20.55 | 5.01 | 19.57 | 0.00 |
| | [20.32,20.64] | | [19.15,19.51] | | [19.77,20.09] | | [18.77,19.08] | | [20.39,20.70] | | [19.41,19.74] | |
| 0.431 | 21.70 | 4.63 | 20.74 | 0.00 | 21.36 | 3.89 | 20.56 | 0.00 | 21.56 | 4.41 | 20.65 | 0.00 |
| | [21.58,21.83] | | [20.62,20.85] | | [21.25,21.46] | | [20.44,20.68] | | [21.47,21.65] | | [20.54,20.76] | |

Fig.3 illustrates the box-plots of numbers of blocking unit-loads obtained by Modi-Indeep (unimproved) and IMP2. Compared with the original retrieval algorithm, IMP2 gets fewer blocking unit-loads. As a consequence, the retrieval machine needs lower operation cost of removing blocking unit-loads and the storage machine spends lower operation cost of re-storing blocking unit-loads. Tab.2 and Tab.3 record the values of ST, RT, ARPD and 95% confident intervals obtained by original and IMP2 with RAN, 1By1 and 10By10 modes. Tab.2 and Tab.3 indicate that the value of ST and RT are reduced by IMP2. The experimental results prove that to reduce blocking unit-loads can improve the work efficiency of storage and retrieval machines.

## Conclusion

In this paper, the effectiveness of two improvements for the Modi-Indeep is analyzed by simulation experiments. Experimental results show that the first improvement can avoid unnecessary movement of the storage machine and the second improvement can reduce blocking unit-loads to improve the work efficiency of storage/retrieval machines. Both improvements can reduce the operational cost. However, the improvements proposed in this paper are very simple. Therefore, the future work is to design more sophisicated improvement for the Modi-InDeep heuristic. As well, the impacts of selection rules to the performance must be studied for designing more appropriate selection rules for in-deep storage and Modi-InDeep heuristic.

## Acknowledgements

This work was supported by National Natural Science Foundation for Young Scientists of China [71602169] and Natural Science Foundation of the Higher Education Institutions of Jiangsu Province, China [16KJB520045].

## References

[1] Bhaba R. Sarker, P. Sobhan Babu. Travel time models in automated storage/retrieval systems: a critical review [J]. International Journal of Production Economics, 1995, 40:173-184.

[2] Bing-bing LIU, Li-bo SUN, Yu-gang YU. Recent advances on researches of warehousing logistics and supply chain management [J]. Journal of University of Science and Technology of China, 2017, 47 (2) :176-187, (in Chinese) .

[3] Kees Jan Roodbergen, Iris F.A. Vis. A survey of literature on automated storage and retrieval systems [J]. European Journal of Operational Research, 2009, 194:343-362.

[4] De Koster, Rene B.M., Le-Duc Tho, et al. Optimal storage rack design for a 3-dimensional compact AS/RS [J]. International Journal of Production Research, 2010, 46 (6) :1495-1514.

[5] Bessenouci, Hakim Nadir, Zaki Sari, et al. Metaheuristic based control of a flow rack automated storage retrieval system[J]. Journal of Intelligent Manufacturing, 2012, 23 (4) :1157-1166.

[6] Chang, Tien-Hsiang, Hsin-Pin FU, et al. Innovative application of an integrated multi-level conveying device to a mobile storage system [J]. The International Journal of Advanced Manufacturing Technology ,2006,29 (9-10) : 962-968.

[7] Lerher, Tone, Matjaz Sraml, Iztok Potrc, et al. Travel time models for double-deep automated storage and retrieval systems. International Journal of Production Research,2010, 48(11) :3151-3172.

[8] Zhu-xi CHEN, Sarker, Xiao-ping LI, et al. Sequencing the storages and retrievals for flow- rack automated storage and retrieval systems with duration-of-stay storage policy [J]. International Journal of Production Research, 2015, 54 (4) :984-998

[9] Olivier Cardin, Pierre Castagna, Zaki Sari, et al. Performance evaluation of in-deep class storage for flow-rack AS/RS [J], International Journal of Production Research, 2012, 50 (23) :6775-6791.

[10] Zhu-xi CHEN, Jatinder N.D. Gupta. Improving the in-deep storage in flow-rack automated storage and retrieval systems [C]. In proc. 48th Annual Meeting of Decision Support Institution, Washington D.C., USA, Nov.18-20, 2017.

[11] Xiao-long GUO, Yu-gang YU, René B.M. De Koster. Impact of required storage space on storage policy performance in a unit-load warehouse [J]. International Journal of Production Research, 2016,54 (8) :2405-2418.