Computers & Industrial Engineering 81 (2015) 1–13

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A storage assignment heuristic method based on genetic algorithm for a pick-and-pass warehousing system ☆

Jason Chao-Hsien Pan [a], Po-Hsun Shih [b], Ming-Hung Wu [a], Jyh-Horng Lin [c,*]

[a] Department of Business Administration, Takming University of Science and Technology, No. 56 Huanshan Road, Section 1, Taipei 11451, Taiwan, ROC
[b] Department of Information Management, Vanung University, No. 1 Van-Nung Road, Chung-Li, Tao-Yuan 320, Taiwan, ROC
[c] Department of digital media design, Cardinal Tien College of Healthcare & Management, No. 407, Sec 2, Jian Fu Road, Sanshing Township, Yilan County 26646, Taiwan, ROC

ARTICLE INFO

Article history:
Received 9 August 2013
Received in revised form 20 November 2014
Accepted 8 December 2014
Available online 17 December 2014

Keywords:
Storage assignment policy
Order picking
Warehouse management
Pick-and-pass system

ABSTRACT

An order storage assignment problem (SAP) is to find an effective way to locate products in a warehouse in order to improve the operational efficiency of order picking. Since SAP is an NP-hard problem, many heuristic algorithms have been proposed. Most of previous researches focused on picker-to-parts warehousing systems or automated storage and retrieval systems. However, pick-and-pass systems play an important role for the faster delivery of small and frequent orders of inventory with the rise of e-commerce and e-business in the global supply chain. Two factors lead to idle time of pickers in a pick-and-pass system: picking line imbalance and shortage replenishment of products. This paper develops a genetic based heuristic method to solve SAP for a pick-and-pass system with multiple pickers to determine the appropriate storage space for each product and balance the workload of each picking zone so that the performance of the system can be improved. A simulation model based on FlexSim is used to implement the proposed heuristic algorithm and compare the throughput for different storage assignment methods as well. The results indicate that the proposed heuristic policy outperforms existing assignment methods in a pick-and-pass system.

1. Introduction

A pick-and-pass system, also called a progressive zoning system, is a commonly used approach for small-to-medium size stock-keeping units (SKUs) such as household, health and beauty, office or food products where SKUs can be stored in relatively small and accessible pick locations along the picking line (Maloney, 2000). Since the rise of e-commerce and e-business, global supply chain management has been focusing on faster delivery of small and frequent orders of inventory at a lower total cost (Chen & Wu, 2005; Petersen & Gerald, 2002). Thus, the operations of the pick-and-pass system have a major impact on the efficiency of the supply chain management.

A storage assignment problem attempts to find an effective way of locating products in a warehouse in order to improve the operational efficiency of order picking. It influences almost all key performance indicators of a warehouse such as order picking time and cost (Muppani & Adil, 2008a), and has been extensively investigated. Since SAP is in the class of NP-hard problems (Frazelle & Sharp, 1989), many heuristic methods for solving SAP have been proposed. The major storage assignment policies discussed in literatures are random storage, frequency-based storage, class-based storage and picking frequency class-based storage policies (Petersen & Gerald, 2004).

Random policy is the simplest one among various storage policies and is applied widely in practice since it often requires less space than other methods, and results in a better level of utilization of all picking aisles (Petersen & Gerald, 2004). Consequently, random storage policy is often used as a benchmark to evaluate the performance of other policies. On the other hand, the frequency-based policy assigns the SKU with the highest picking frequency to the storage location closest to the Input/Out point and the SKU with next highest frequency to the next closest location and so on. In between, the class-based policy is a compromise between the accuracy of the frequency-based and the easiness of the random storage policies (Benjamin, Bart, & Denis, 2003). The picking frequency class-based storage policy is the most popular one adopted in warehouses in which SKUs are classified into classes according to their picking frequencies and randomly stored in respective classes. Petersen and Gerald (2004) analyzed the relation between the number of product classes and the pickers' travel time in picker-to-parts picking systems by simulation. Muppani and Adil (2008a) developed a

☆ This manuscript was processed by Area Editor Manoj Tiwari.

* Corresponding author. Tel.: +886 2 2219 1131x7114; fax: +886 2 2219 8074.
E-mail address: linjh@ctcn.edu.tw (J.-H. Lin).

http://dx.doi.org/10.1016/j.cie.2014.12.010
0360-8352/

simulated annealing algorithm to solve an integer programming model for class formation and storage assignment by considering all possible product combinations, storage-space cost and order-picking cost. Muppani and Adil (2008b) also proposed a nonlinear integer programming model and a branch-and-bound algorithm for the formation of storage classes, and demonstrated that there are significant savings in using class-based storage policy. Park and Seo (2009, 2010) formulated the planar storage location assignment problem (PSLAP) by a mathematical programming model and developed an efficient genetic algorithm (GA) for a large size real life problem and developed dynamic heuristic procedures (2010) accordingly. Chan and Chan (2011) indicated that the key to effective implementation of a storage assignment system is to match the types of warehouse storage system and the variety of SKUs in the customer orders. Grosse, Glock, and Jaber (2013) developed a mathematical model to investigate the effects of worker learning and forgetting on storage assignment strategies in a typical order picking system. Kovàks (2011) proposed a mixed integer program model for finding a class-based storage policy that minimizes the order cycle time, the average picking effort, or a linear combination of these two criteria in a warehouse characterized by multi-command picking and served by milkrun logistics. Glock and Grosse (2012) proposed different storage location assignment policies for an order-picking system where the warehouse is divided into zones with shelves being arranged in the shape of a U in each zone. Chuang, Lee, and Laih (2012) presented a two-stage clustering-assignment problem model to improve the performance of the frequency-based assignment method for a customized-orders picking problem. Bottani, Cecconi, Vignali, and Montanari (2012) used a genetic algorithm to optimize item allocation in a warehouse for reducing the travel time of pickers.

While most warehousing storage related research focused on automatic storage and retrieval systems (AS/RS) or picker-to-parts systems, few considered the pick-and-pass systems. In a pick-and-pass system, a picker finishes picking one part of an order and then hands to the picker in the next zone until all the SKUs are picked in that order. De Koster (1994) approximated the picking operation by means of a Jackson network model and assumed that the service time at each pick station is exponentially distributed and customer orders arrive according to a Poisson process. Malmborg (1995) studied the problem of assigning products to locations with zoning constraints. Petersen and Gerald (2002) showed that storage policy has a significant effect on the average travel distance within a zone. Jewkes, Lee, and Vickson (2004) developed an efficient dynamic programming algorithm to determine the optimal SKUs allocation and picker locations for an order picking line with multiple pickers. Jane and Laih (2005) presented several heuristic algorithms to balance the work load among order pickers in a picking line. Gagliardi, Ruiz, and Renaud (2008) proposed and analyzed different product location and replenishment strategies for a distribution center that uses a pick-and-pass system for order fulfillment.

Two important issues have to be taken into account in solving a SAP for a pick-and-pass system. The first one is the shortage of SKUs in the zone. A pick-and-pass system can be divided into the forward area and the reserve area (Rouwenhorst et al., 2000). Pickers fulfill orders in the forward area, i.e., the picking lines. Once an SKU to be picked is out of stock, a picker must stop until replenishment is made using the stored SKUs from the reserve area. A picker is unproductive while he/she calls for an emergency replenishment that impacts supply chain performance (Gagliardi et al., 2008). Hence, the storage space of each of the SKUs should be properly allocated based on the market demand in order to avoid as much as possible the shortage of SKUs in the racks of the picking lines. Consequently, an optimal assignment policy has to consider simultaneously both the picking line balance and the space allocation under such circumstance. Second, the balance of the workloads among all pickers has a great impact on the efficiency of the system. Similar to a manufacturing flow line, imbalance can cause serious deterioration of order throughput time as multiple pickers are working in a pick-and-pass system (Brynzer & Johansson, 1995; Pan & Wu, 2009). Since pickers need to wait for the preceding one to finish picking and pass on the container that carries the SKUs picked, the picking times between two consecutive zones in a line may vary and the congestion of the containers in the line inevitably occurs. The purpose of his paper is to develop a genetic algorithm based storage assignment heuristic algorithm in a pick-and-pass system with the consideration of the line balance and the space allocation to improve the throughput of the system. The performance of the algorithm is validated and compared with the results generated by simulation modes.

2. Description of picking operations and storage assignment problem

2.1. The picking operation for a pick-and-pass system

A pick-and-pass system is composed of I picking lines each having a roller conveyor connecting its J stations (zones) located along the lines, as illustrated in Fig. 1. Each zone has only one picker who picks the SKUs of an order located in the vertical shelves in his/her zone. An order may be fulfilled by several picking lines and then all parts of the order in these picking lines are combined to complete the picking.

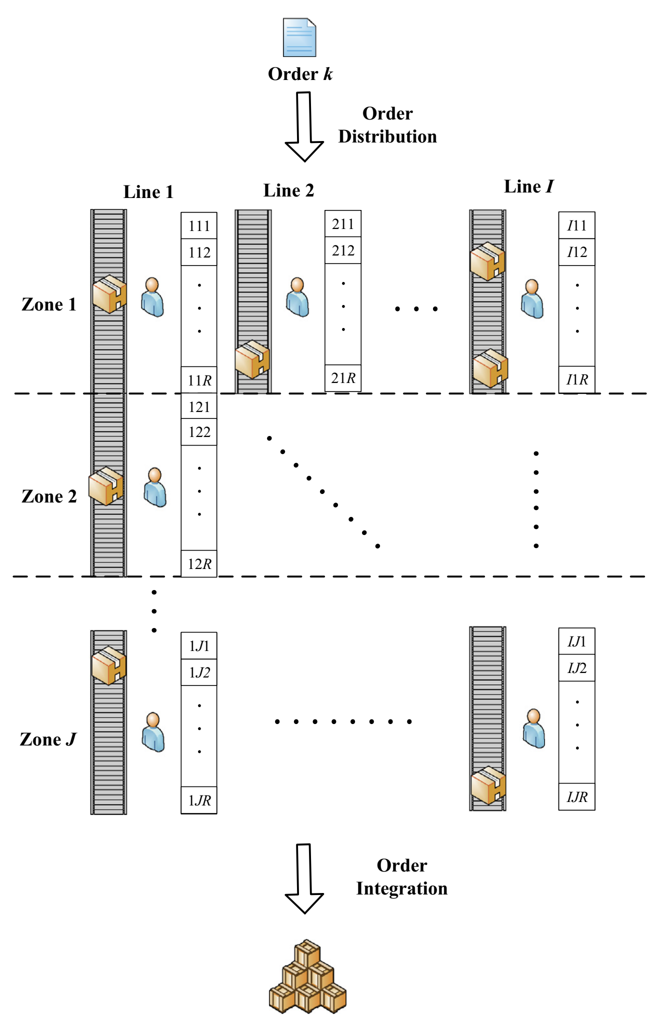


Fig. 1. A pick-and-pass system with multiple synchronized picking zones.

In practice, most of pick-and-pass systems often use a computer aided picking system (CAPS) to assist the activities of the order picking. The CAPS effectively improves the picking productivity by 50% or more, and reduces the picking task error (Jane & Laih, 2005). In addition, CAPS further simplifies the training for the pickers, thus cuts down the operational cost.

Under the use of CAPS, light indicator modules are installed on all racks to guide pickers to the picking locations, and show the exact amount of each SKU to be picked in the zone. The picker retrieves the SKU quantity displayed on a rack and then confirms the pick by pressing the lighted button. After the completion of the pickings of one order in the assigned zone, the CAPS sends the data of next order to all light indicator modules in that zone.

For a picking operation, it is reasonable to assume that the seek time of a picker can be omitted. Moreover, the travel time of pickers can also be neglected because each picker moves within a small area. Thus the workload is measured by the picking time rather than by the travel time (Jane & Laih, 2005).

The following assumptions on the pick-and-pass system and the picking operations under study are made in this paper:

(1) All SKUs of all the orders have the same size and weight.
(2) The time to pick an SKU from a rack is constant.
(3) Each SKU is independent of the others in an order.
(4) Each SKU is allocated in one zone only, i.e., an SKU can only be picked by one picker.
(5) Each rack stores only one SKU.
(6) The storage racks are narrow and low, i.e., the travel time is negligible.
(7) All the picking zones have the same number of racks.
(8) Once a shortage of SKU is detected, an emergency replenishment is carried out and the time to replenish one SKU is constant.

The notation used in this paper is defined as follows:

i	picking line index, $i = 1, 2, \ldots, I$
j	picking zone index, $j = 1, 2, \ldots, J$
r	rack index, $r = 1, 2, \ldots, R$
k	SKU index, $k = 1, 2, \ldots, K$
n_k	mean number of SKU k to be picked
p_k	demand rate of SKU k to be picked
c	capacity of a rack
t	picking time
L	mean workload

2.2. A formulation model for storage assignment problem

The idle time may exist for some pickers in a pick-and-pass system due to the line imbalance or the emergency replenishment of SKUs. When SKUs can be located evenly in each zone based on their demands, the idle time of pickers can be reduced and the performance of the system can thus be improved. Hence, this paper formulates a mathematical model for the storage assignment problem in a pick-and-pass system with multiple picking lines under study as follows:

$$\text{Minimize} \quad Z = w_1 \sum_{k=1}^{K} \left| c \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{r=1}^{R} x_{ijrk} - n_k p_k \right| + w_2 \sum_{i=1}^{I}\sum_{j=1}^{J} \left| \sum_{r=1}^{R}\sum_{k=1}^{K} n_k p_k x_{ijrk} t - L \right|. \quad (1)$$

Subject to

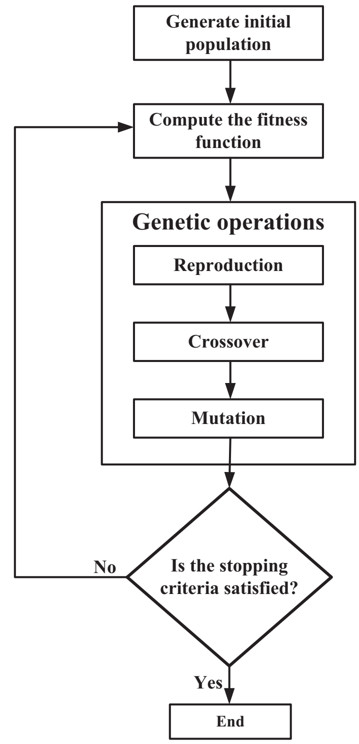


Fig. 2. The typical procedure of GAs.

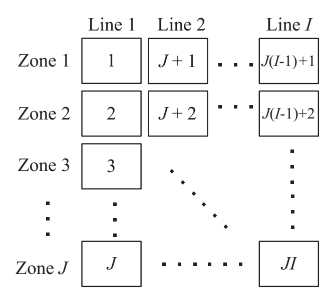


Fig. 3. The numbering of zones in a pick-and-pass system with multiple picking lines.

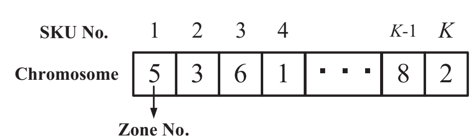


Fig. 4. A chromosome for storage assignment in a pick-and-pass system.

$$\sum_{k=1}^{K} x_{ijrk} = 1 \text{ for } i = 1, \ldots I; \quad j = 1, \ldots, J, \text{ and } r = 1 \ldots, R. \quad (2)$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{r=1}^{R} x_{ijrk} - \sum_{r=1}^{R} x_{ijrk} \leq 0 \text{ for } i = 1, \ldots, I; \; j = 1, \ldots, J, \text{and } k = 1, \ldots, K. \quad (3)$$

$$x_{ijkr} = 0 \text{ or } 1 \text{ for } i = 1, \ldots, I; \quad j = 1, \ldots, J; \; k = 1, \ldots, K, \text{ and } r = 1, \ldots, R. \quad (4)$$

where $x_{ijrk} = 1$, if SKU k is assigned to rack r of zone j in picking line i; and 0, otherwise. The constraint set (2) ensures that one rack only stores an SKU. The constraint set (3) is used to restrict that an SKU is assigned to a single zone only. The term $\left|c\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{r=1}^{R}x_{ijrk} - n_k p_k\right|$ in object function (1) depicts the absolute difference between the number of a certain SKU in the system and the expected number of that SKU to be picked. To avoid SKUs replenishment, the space size, or equivalently the number of racks in a zone, in which an SKU is located should approach its expected number to be picked, $n_k p_k$, as much as possible. The term $\sum_{i=1}^{I}\sum_{j=1}^{J}\left|\sum_{r=1}^{R}\sum_{k=1}^{k} n_k p_k x_{ijrk} t - L\right|$ in object function (1), the summation of absolute deviations (SAD) of workloads of all pickers, is to measure the level of workload balance of the pick-and-pass system. Since the number of zones is fixed in a line, minimizing SAD is equivalent to minimizing the mean absolute deviation (MAD). If all SKUs can be more fairly allocated to zones, then each picker's picking time would be closer to the mean workload of all the pickers in the system. As a result, the SAD value can be reduced. w_i is a constant positive weight. For the picking operation, it is reasonable to assume that the picker's idle time caused by an emergency replenishment is much greater than that caused by an imbalance picking line, because a picker must stop until SKUs are transported to his/her picking zone from the reserve area and even the other pickers need to wait for that picker. Thus, we assume $w_2 \gg w_1$.

Because the proposed mathematical model is a classical SAP which has been proven a NP-hard problem (Abdel-Hamid & Borndörfer, 1994; Frazelle & Sharp, 1989), this paper presents a method for the space requirement and uses GA to deal with the location assignment of the SKUs.

3. A heuristic algorithm for the SAP in a pick-and-pass system

3.1. The storage space allocation

The number of racks an SKU allocated in a zone should be determined before the decision of location assignment is made, so that all of SKUs can be placed into the warehousing system and the shortage of SKUs can be reduced. Thus, an optimal upper bound of the space requirement of an SKU to be located is proposed. The sum of the squared difference between the expected numbers of all SKUs to be picked and the numbers of storage racks of all times in the system can be expressed as follows:

$$f = \left(c\sum_{k=1}^{K} Y_k - n_k p_k\right)^2, \tag{5}$$

where $Y_k = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{r=1}^{R}x_{ijrk}$ is the number of space for SKU k to be stored.

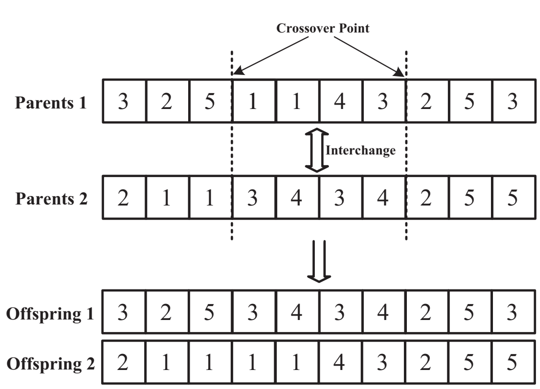


Fig. 5. An example of two-point crossover.

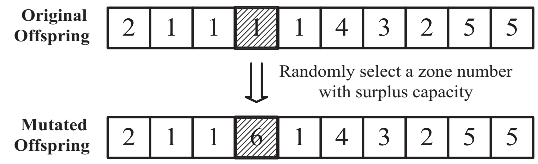


Fig. 6. An example of a mutation operator.

To find the minimum value of f, take the partial derivative of f with respect to Y_k as follows:

$$\frac{\partial f}{\partial Y_k} = 2(cY_k - n_k p_k) \text{ for } k = 1, 2, \ldots, K. \tag{6}$$

Setting Eq. (6) to zero and solve for Y_k, it follows that

$$Y_k = n_k p_k / c, \quad \text{for } k = 1, 2, \ldots, K. \tag{7}$$

The number of storage space in whole system is $I \times J \times R$; hence,

$$\sum_{a=1}^{K} Y_a \leq IJR. \tag{8}$$

And,

$$\frac{\sum_{a=1}^{K} Y_a}{Y_k} \leq \frac{IJR}{Y_k} \quad \text{for } k = 1, 2, \ldots, K. \tag{9}$$

Using Eq. (7), we have

$$\frac{\sum_{a=1}^{N} n_a p_a / c}{n_k p_k / c} \leq \frac{IJR}{Y_k} \quad \text{for } k = 1, 2, \ldots K. \tag{10}$$

Then, the optimal upper bound of space requirement of SKU k to be located is

$$Y_k \leq \frac{n_k p_k IJR}{\sum_{a=1}^{N} n_a p_a} \quad \text{for } k = 1, 2, \ldots, K. \tag{11}$$

3.2. A GA based assignment storage location algorithm

After the spaces of all SKUs are determined, these SKUs can be distributed to picking zones. A heuristic algorithm based on GA is proposed to locate each SKU into a warehouse in order to balance the workload of the pickers. GA (Holland, 1975) is one of the most widely used approaches to solve optimization problems which can be described with chromosome encoding. Because GAs practically do not need the knowledge of mathematics and only use a fitness function to evaluate the quality of different solutions, even specific large-scale problems can be solved very effectively with acceptable good solutions. Hsu, Chen, and Chen (2005) and Bottani et al. (2012) advocated that GA approach is beneficial to the development of order picking polices.

In the application of GAs, a solution is transformed into a chromosome which is composed of several codes (genes). The optimal solution can be obtained by the operations of genes of chromosomes as shown in Fig. 2. Before a GA can be implemented, a suitable coding and fitness function for the SAP must be developed.

3.2.1. Encoding the SAP

Following the convention of a straightforward encoding approach with a string of integers used by classic GA, picking zones are numbered sequentially in the pick-and-pass system as shown in Fig. 3. Consequently, the encoding chromosome of a solution can be expressed as shown in Fig. 4. If K SKUs are to be located, a chromosome with K genes is used where the first gene (code) denotes the zone of the first SKU to be located, the second gene denotes the zone of the second SKU to be located and so on.

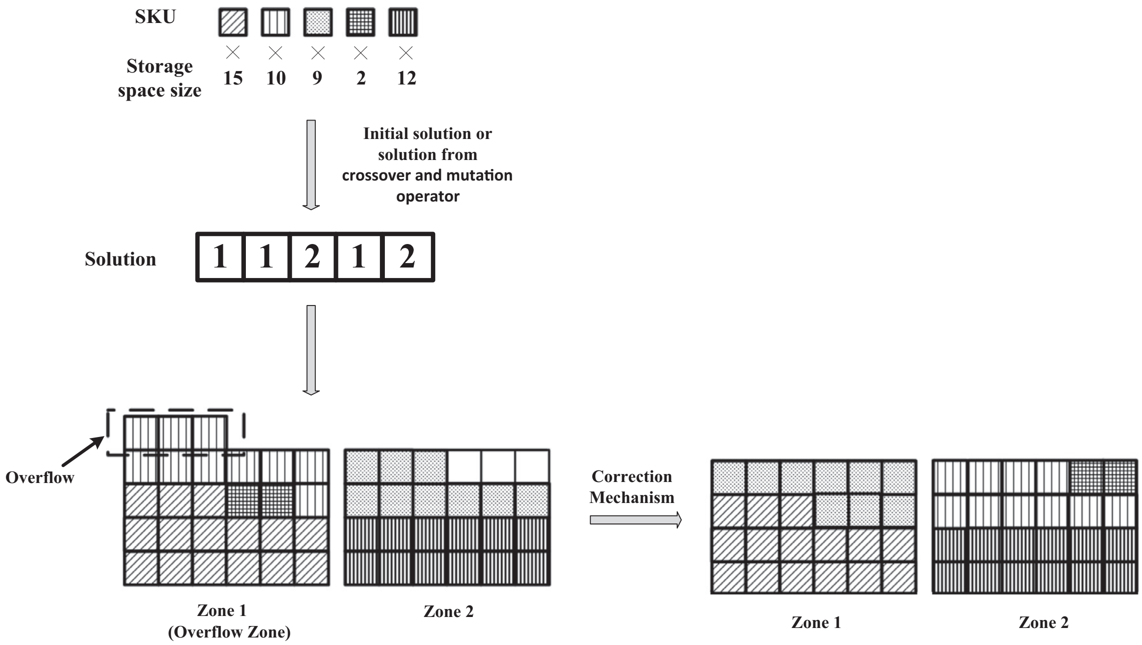


Fig. 7. An example of overflow zone.

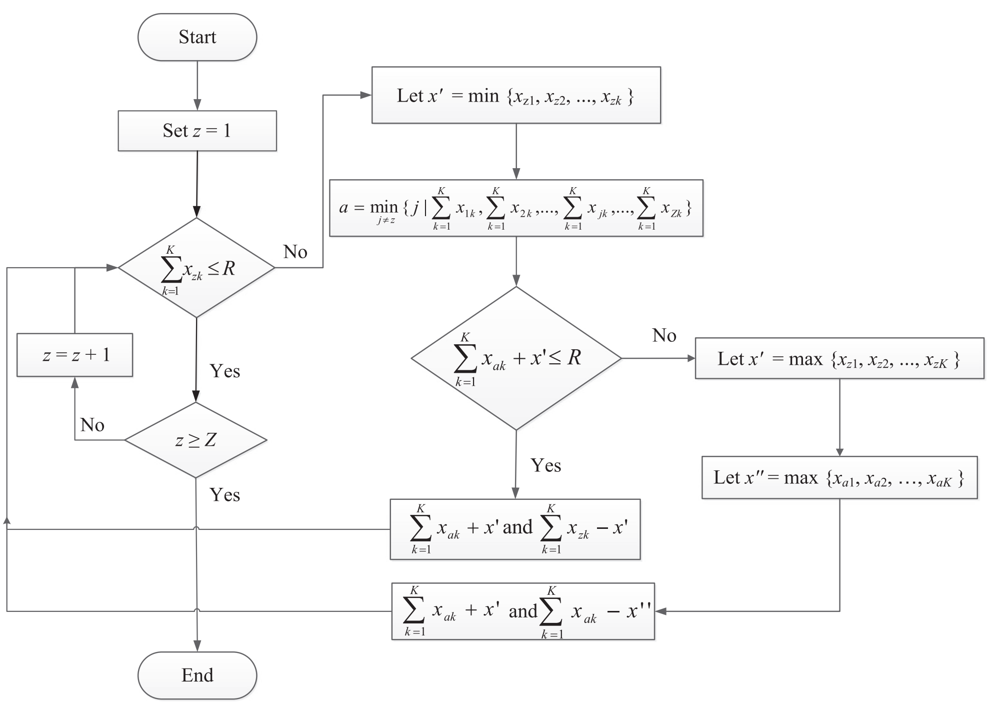


Fig. 8. The procedure of the proposed correction mechanism.

Fig. 4 illustrates that SKUs 1, 2, 3 and 4 are located into zone 5, 3, 6 and 1, respectively.

3.2.2. Fitness function

The performance of an individual chromosome must be measured via a fitness function which represents the opportunity of the chromosome to be selected to reproduce the next generation of offspring. In a pick-and-pass system, the objective of the SAP is to reduce the idle time of pickers through workload balancing and thus can be incorporated in the fitness function of the problem. Since the first term of object function in Eq. (1) describes the level of workload balance, it is modified to be the fitness function as follows:

$$f = \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \left| \sum_{k=1}^{K} \sum_{r=1}^{R} n_r p_r x_{ijrk} - \sum_{k=1}^{K} p_k n_k / IJ \right| \right)^{-1}, \quad (12)$$

where $\sum_{k=1}^{K} p_k n_k / IJ$ is the expected workload of an individual zone. Such a fitness function implies that a chromosome with better performance indicated by a higher value of f would have a higher probability to be selected.

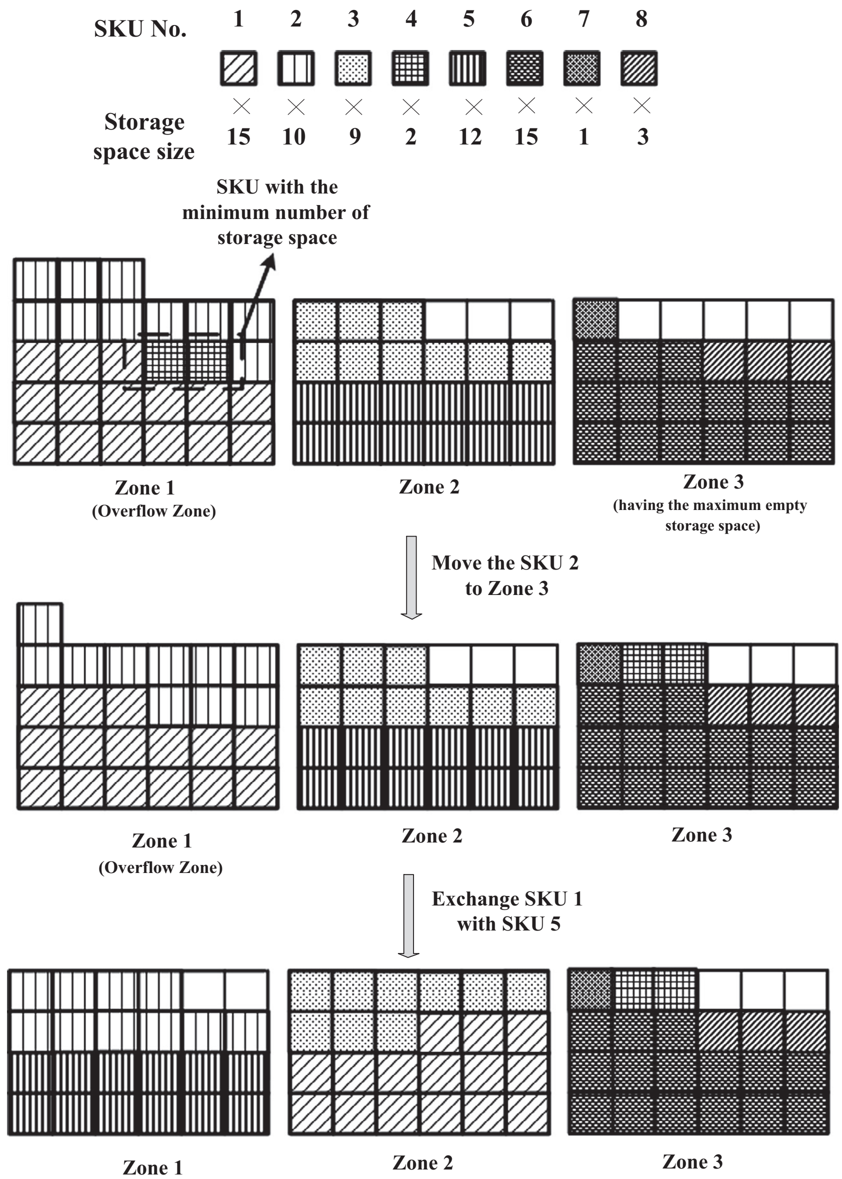


Fig. 9. An example of the correction mechanism.

3.2.3. Selection

Selection is a GA operator that selects chromosomes from current generation to be copied into next generation. Many types of selection method are used in GA and the linear ranking selection (Falkenauer, 1998) is adopted to determine the probability of the chromosomes to be selected. For linear ranking selection, the chromosomes are sorted according to their fitness values, in which rank M is assigned to the best chromosome and rank 1 to the worst chromosome. The fitness of chromosome v is normalized by

$$f'_v = \frac{f_{\max} - f_{\min}}{M}(\text{Rank } v). \tag{13}$$

And, the probability of chromosome v to be selected is

$$r_v = \frac{f'_v}{\sum_{v=1}^{M} f'_v}. \tag{14}$$

3.2.4. Crossover

Crossover is a genetic operator which selects genes from parent chromosomes and generates a new offspring. A two-point crossover operator is to select randomly two crossover points within two parent chromosomes and then interchange the genes of the paired chromosomes between those points to produce two new offspring as shown in Fig. 5.

3.2.5. Mutation

Mutation operator is to alter randomly one or more gene codes in a chromosome from its initial state with a certain probability. In order to meet the limited space constraints, the gene codes selected are replaced with a candidate code. Fig. 6 shows an example of a mutation operator.

3.2.6. Correction mechanism and the replenishment rule

Initial solutions that are randomly generated or the solutions produced from the proposed crossover and mutation operators may have overflow zones in which the numbers of SKUs to be stored are over the capacity of the zones. Fig. 7 illustrates an example of an overflow zone. In the example, five SKUs need to be stored in two picking zones each having 24 racks. The storage space of each SKU is also listed in Fig. 7. After randomly generating an initial

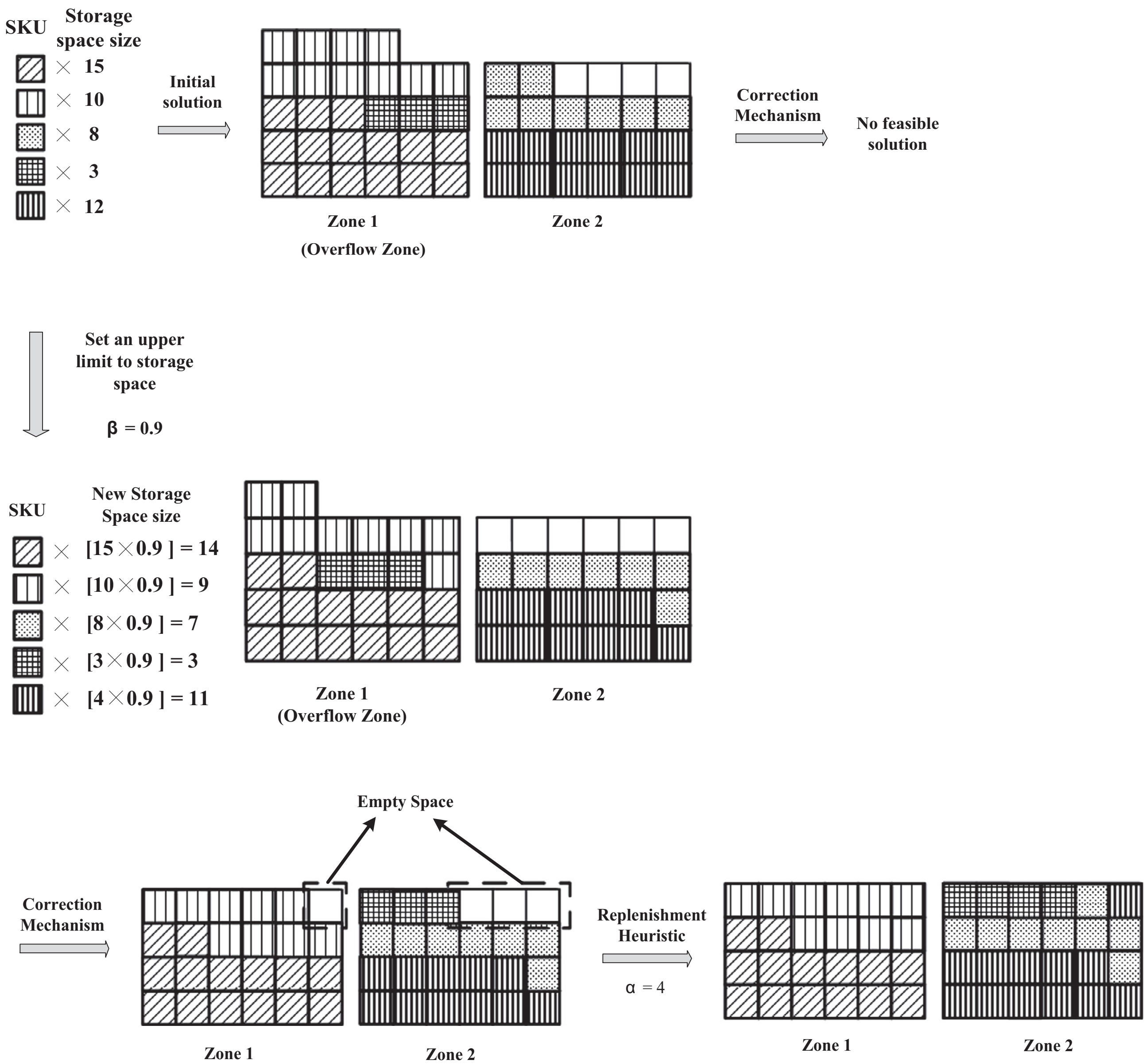


Fig. 10. An example of an infeasible solution and the replenishment heuristic.

Table 1
Summary of the experimental scenarios.

	Scenario no.							
	1	2	3	4	5	6	7	8
No. of type of SKUs (N)	50	100	100	150	200	250	300	300
Distribution of no. of a SKU to be picked	U[5,10]	U[1,10]	U[5,10]	U[1,15]	U[5,15]	U[1,15]	U[1,15]	U[5,15]
No. of zones (J)	5	5	10	5	10	10	5	10
No. of lines	2	2	2	3	3	3	4	4

solution of (1,1,2,1,2), the three SKUs in zone 1 require a total of 27 storage racks which overflows the capacity of the zone. Therefore, a correction mechanism has to be established to ensure that the resulting number of the racks required does not exceed the maximum storage space allowed in a zone by moving some of the SKUs in overflow zones to those with surplus capacities until no overload exists in any zone.

Fig. 8 illustrates the procedure of the proposed correction mechanism where x_{zk} denotes the number of SKU k in zone z and Z be the number of zones in the system. When zone z has excessive SKUs to be stored, SKU k which uses the minimum number of storage racks in that zone will be moved to zone j which has the maximum empty storage space until no overflow zone exists in the line. If no other zones can store SKU k, then the SKU occupying the maximum number of storage racks in zone i will be exchanged with the SKU using the maximum number of storage racks in zone j. Fig. 9 illustrates such an example. In Fig. 9, zone 1 is an overflow area. SKU 4 in zone 1 requires only two racks and will be moved to

Table 2
Comparisons of the number of stockouts for different GA parameters for scenario 1.

GA parameters		Values		
Population size		10	20	30
	Mean	390.74	363.40	350.00*
	S.D.	4.76	3.70	3.53
Crossover rate		0.4	0.6	0.8
	Mean	390.40	350.00*	362.76
	S.D.	4.65	3.53	4.25
Mutation rate		0.01	0.02	0.03
	Mean	350.00*	361.42	370.26
	S.D.	3.53	4.34	4.05

* The best value among the tests (*P*-value < 0.01, the number of samples is 50), and S.D. stands for ‘standard deviation’.

zone 3 which has the maximum empty storage space. After the first correction, zone 1 is still in the overflow situation and the other two zones do not have enough space to accommodate SKU 2. Consequently, SKU 5, the one using the maximum number of storage racks in zone 2, will be selected and exchanged with SKU 1 which uses the maximum number of storage racks in the overflow zone.

However, not all infeasible solutions can be resolved through the proposed correction mechanism. When an initial allocation space size is too large, the overflow zone may not be able or difficult to move or interchange SKUs with other zones which have surplus capacities. Hence, this paper proposes a parameter β to determine the upper limit of the storage space available to ensure the flexibility of the adjustment of the storage positions, where $0 < \beta \leqslant 1$. Let n_k^U be the upper limit on the number of SKU k to be picked and $n_k^U = [n_k \times \beta]$. The upper limit of the storage space available in a pick-and-pass system can be determined by $\sum_{k=1}^{K} n_k^U$. For example, there are two picking zones having 24 racks each in a warehouse and the allocation space of five SKUs are determined as shown in Fig. 10. Each combination of storage location is not a feasible solution. If $\beta = 0.9$, the storage space of each SKU will be relatively reduced and a feasible solution can be obtained accordingly.

Fig. 10 also illustrates that several empty storage spaces may exist in some zones after the completion of the storage assignment and initial space allocation. Filling up these spaces can reduce the possibility of out of stock of SKUs. This paper proposes a replenishment heuristic based on the ratio of the

Table 4
Mean completion time (unit time) for the eight scenarios on 500 orders in the experiment.

Scenario	RG	FG	HG
1	11,205	11,807	10,393
2	19,188	18,695	14,972
3	11,094	11,653	10,217
4	25,803	28,878	21,601
5	24,968	25,978	19,081
6	27,090	27,254	20,087
7	45,695	43,384	34,889
8	26,595	26,235	21,572

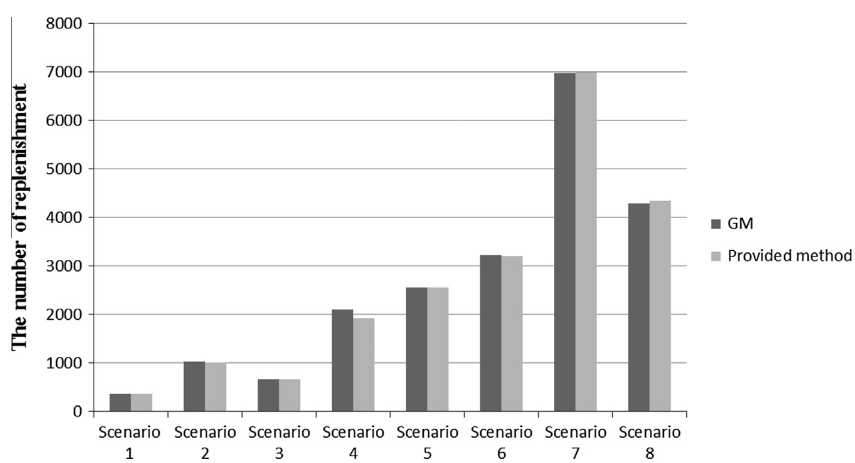


Fig. 11. A comparison of the number of replenishment between GM and the proposed method.

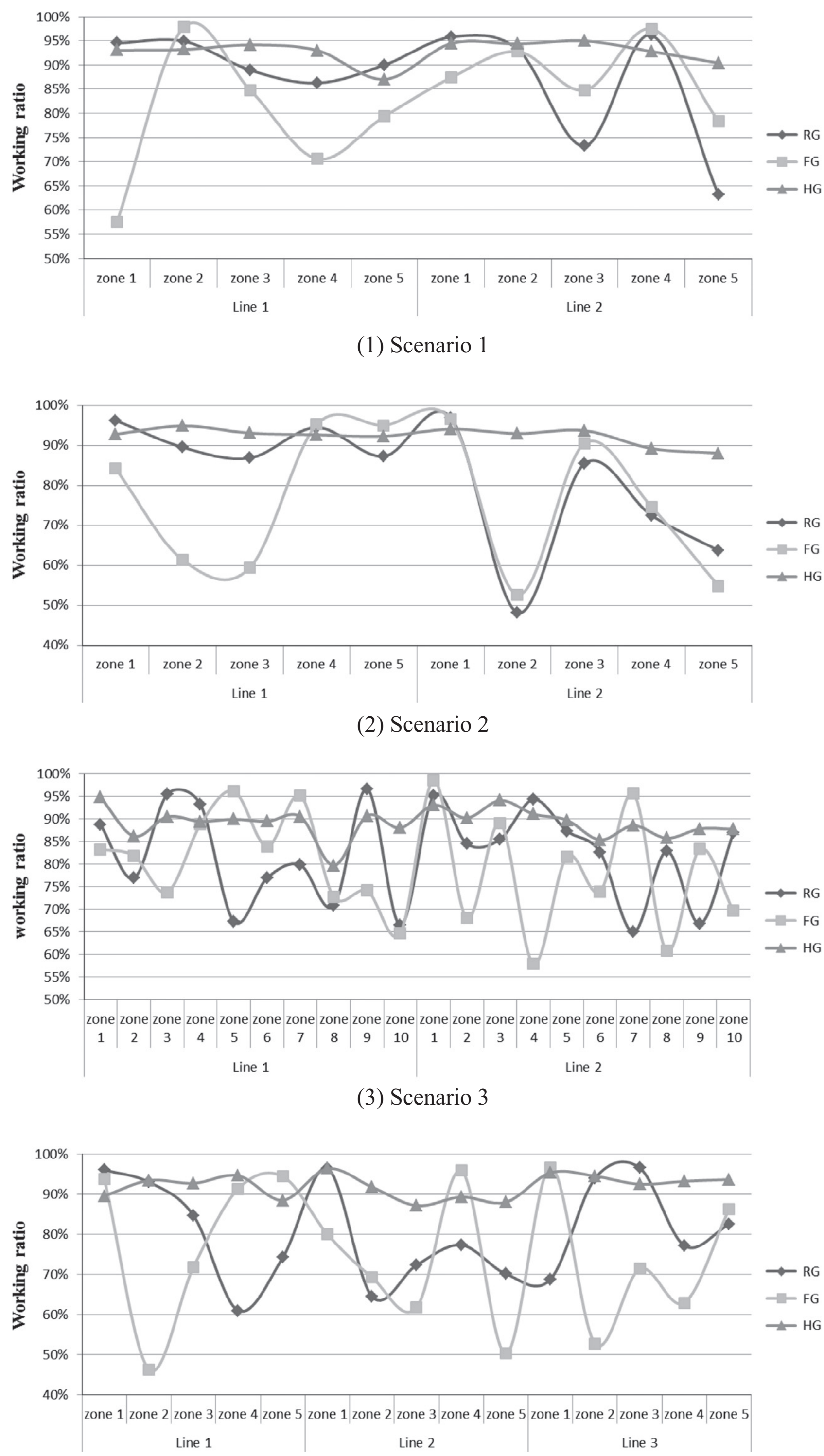
Table 3
Comparisons of the number of stockouts among storage policies in the experiment.

	Scenario no.							
	1	2	3	4	5	6	7	8
RG								
Mean	361.44	1043.24	662.34	2234.74	2684.20	3368.36	7484.66	4498.40
S.D.	3.20	8.31	5.33	13.77	12.39	13.73	26.99	16.50
FG								
Mean	363.14	1044.40	676.30	2204.74	2664.44	3378.08	7439.62	4580.54
S.D.	3.53	7.14	5.03	10.63	9.24	12.79	23.96	18.30
HG								
Mean	350.00*	1010.62*	651.22*	2119.68	2548.28*	3195.56*	6977.64*	4334.20*
S.D.	3.53	6.55	4.55	11.06	17.16	13.90	28.63	13.23

* Significant improvements (*P*-value < 0.01, the number of samples is 50), and S.D. stands for ‘standard deviation’.

Table 5
Blocking ratio statistics for the eight scenarios in the experiment (%).

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario8
RG	12.38	17.05	12.38	19.50	26.12	25.66	23.05	21.18
FG	16.92	15.35	16.92	25.08	24.73	25.05	21.72	24.55
HG	7.26	6.61	7.26	8.05	11.48	10.59	9.87	10.25



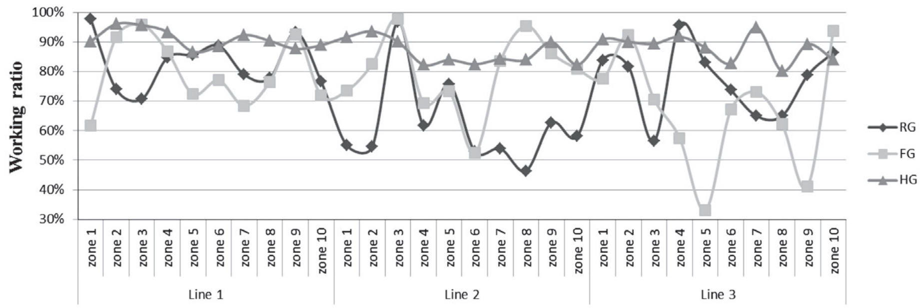
(1) Scenario 1

(2) Scenario 2

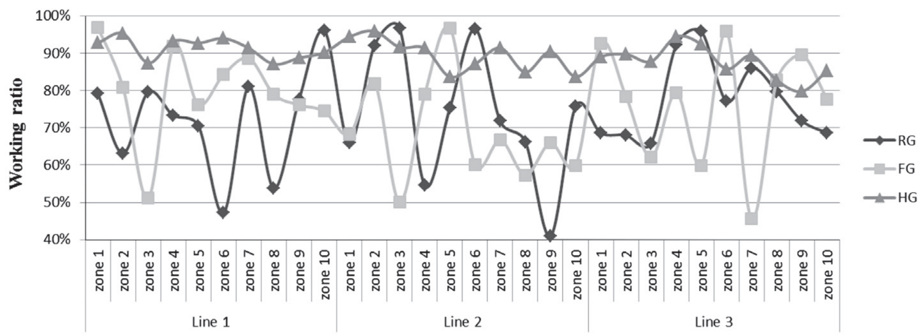
(3) Scenario 3

(4) Scenario 4

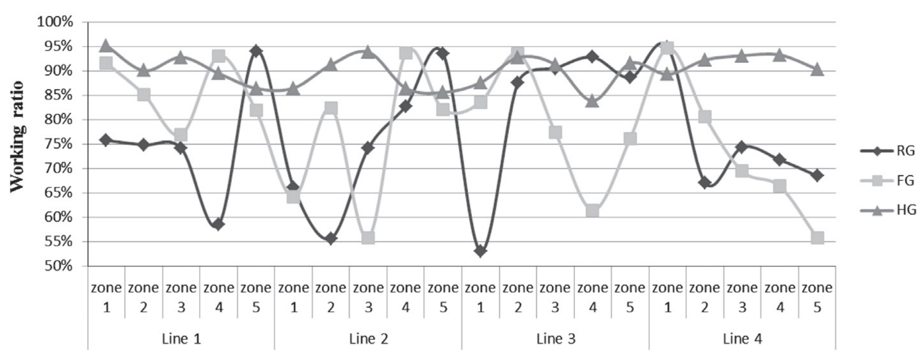
Fig. 12. Utilization statistics for each zone with eight scenarios in the experiment.



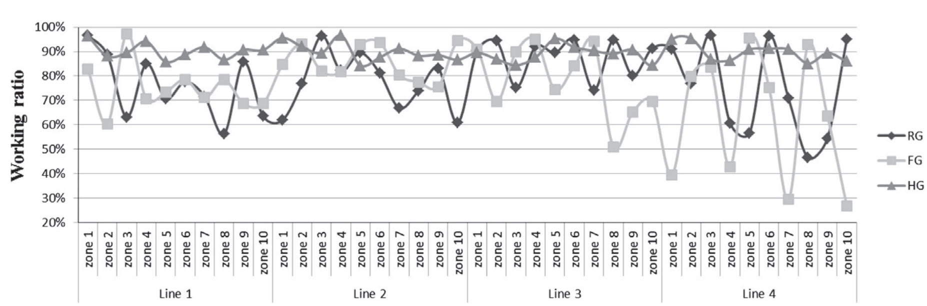
(5) Scenario 5



(6) Scenario 6



(7) Scenario 7



(8) Scenario 8

Fig. 12 (*continued*)

expected numbers of an SKU to be picked and the current number of racks the SKU in the zone, $\tau_{ijk} = \sum_{r=1}^{R} x_{ijrk}/n_k r_k$. Gagliardi et al. (2008) found that some products in a small amount of racks lead to too many stockouts. Thus, this study also designs a parameter α to limit all SKU using the minimum number of racks in a zone. For example, $\alpha = 2$ denote that each SKU in a zone has at least two racks. The replenishment heuristic can be described as follows:

Step 1: Set the lower bound on the number of storage space of a SKU in a zone, α.
Step 2: Find SKU a, the one uses the minimum number of racks in zone z.
Step 3: If $\sum_{r=1}^{R} x_{izra} < \alpha$, then go to Step 5; otherwise, go to step 4.
Step 4: Find SKU a, which has the minimum ratio τ_{izk} in zone z.
Step 5: Add SKU a to zone z.

Table 6
Blocking ratios of bottleneck zones for all picking lines with eight scenarios in the experiment (%).

	Scenario 1		Scenario 2		Scenario 3		Scenario 4			Scenario 5			Scenario 6			Scenario 7				Scenario 8			
	Line 1	Line2	Line 1	Line2	Line 1	Line2	Line 1	Line2	Line 3	Line 1	Line2	Line3	Line 1	Line2	Line3	Line 1	Line2	Line 3	Line4	Line 1	Line2	Line3	Line4
RG	13.73	36.82	13.17	51.83	33.51	35.05	39.09	35.59	31.26	29.33	53.60	43.55	52.83	59.03	34.20	41.45	44.25	12.40	32.83	43.85	39.23	25.80	53.58
FG	42.42	21.60	40.66	47.31	35.24	42.09	53.75	38.25	47.35	38.38	47.59	67.02	48.95	49.94	54.43	23.06	44.13	38.46	44.20	39.79	24.46	49.11	73.42
HG	12.97	9.58	7.67	11.95	20.34	14.69	11.59	12.93	7.62	13.28	17.69	19.81	12.97	16.36	20.27	13.60	14.43	16.11	10.72	14.23	15.70	15.67	14.91

Step 6: If zone z is full, then go Step 7; otherwise, go to Step 2.
Step 7: Repeat Step 2 to Step 6 until all zones are full.

The overall procedure of the proposed storage assignment heuristic algorithm based on GA for the pick-and-pass system can be described as follows:

Step 0: Set the upper limit of the storage space available, β.
Step 1: Use Eq. (11) to determine the storage space of SKU k, $k = 1, 2, \ldots, K$.
Step 2: Set population size (M), no. of generations (T), crossover rate (Rc) and mutation rate (Rm), $m = 1$ and $g = 0$.
Step 3: Randomly generate chromosome m in generation g.
Step 4: Use the correction mechanism to correct all chromosomes.
Step 5: If $m = M$, go to Step 6; otherwise, set $m = m + 1$ and go to Step 3.
Step 6: Compute f_v in generation g using Eq. (11), $v = 1, 2, \ldots, M$.
Step 7: Select M chromosomes from generation g using Eq. (13) into next generation ($g + 1$).
Step 8: Select randomly $M \times Rc$ chromosomes from generation ($g + 1$) to be ($M \times Rc/2$) pairs of parents.
Step 9: Generate $M \times Rc$ offsprings by the crossover operator and the correction mechanism from those parents.
Step 9: Replace the parents with the offsprings.
Step 10: Select randomly $M \times Rm$ chromosome from generation ($g + 1$) and apply the mutation operator and the correction mechanism.
Step 11: If $g = T$, stop; otherwise, $g = g + 1$ and go to Step 6.

The proposed algorithm is basically an GA approach with a complexity of $O(\log K)$ (Rylander & Foster, 2001).

4. Experimental design and simulation model

4.1. Description of the warehouse configuration

As there are no test problems in the related literature, a pick-and-pass system with each zone having 60 racks and the capacity of a rack is 20 units is designed for numerical experiment. The picking time is one unit time per item. Eight scenarios with different number of picking zones and picking lines are designed to implement the storage assignment polices, as listed in Table 1. Each picking probability is generated randomly and the replenishment time per one SKU is five unit time. The picking time includes all item handling and administration activities. Moreover, β and α are assumed to be 0.9 and 2, respectively. Different values were tested for scenario 1 in order to find a better set of the GA parameter values. The results are listed in Table 2. Since population size = 30, crossover rate = 0.6 and mutation rate = 0.05 generate the lowest number of stockouts, this set of values is used to run the GA for 50 generations.

In order to measure the performance of the proposed method, a simulation model based on CAPS is implemented in Flexsim (2007). The FlexSim is a 3D object-oriented simulation environment for modeling discrete-event flow processes. The model was run 50 times with 500 orders for each picker in the line.

4.2. Computation results

This section compares the approach of Gagliardi et al. (2008), first-come-first-served (FCFS), random and the proposed heuristic assignment methods in an attempt to study their efficiencies using numerical data analysis with approximation method. So far, the research by Gagliardi et al. (2008) is the only one relevant to the storage assignment method for the pick-and-pass system with the consideration of replenishment. Gagliardi et al. (2008) used four different ratios to allocate storage SKUs space and found that the one based on the SKU demand frequency and the number of racks currently allocated to SKU is the best index among the four. Their tests are based on real data and show a 2% reduction on the number of stockouts. Since the space allocation heuristic with the demand frequency ratio has the best performance among the combinations Gagliardi et al. tested (2008), it is selected and denoted by GM. Fig. 11 indicates that no significant differences exist between these two methods on the number of replenishment at a confidence level of 99%; however, GM requires iterative simulation runs to obtain a better solution and that would consume computing time while the proposed method can achieve the same quality of solution without complex calculations. The main purpose of storage assignment policy is to assign SKUs to locations so that order picking time or cost can be reduced. But, in the Gagliardi's research (2008), the aim of the storage assignment policy is to determine the number of racks used by each SKU in order to reduce the number of stockouts without the consideration of the location of SKUS. Therefore, this paper proposes to incorporate the random and FCFS policies into GM, denoted by RG and FG, respectively, for the comparison with the proposed GA based algorithm denoted by HG. The procedures of RG and FG can be expressed by the following two steps:

Step 1: Use GM to determine the number of racks for each SKU.
Step 2: Use random (in RG)/FCFS (in FG) policy to determine the location for each SKU.

Table 3 indicates that the number of replenishment using HG is significantly lower than those of the other two polices at a confidence level of 99% for all scenarios except scenario 4. Table 4 reveals that the completion time of 500 orders to be picked using HG is the shortest under all scenarios tested. This outcome is reasonable because the proposed heuristic algorithm not only reduces the opportunity of shortage but also attempts to balance the workload among order pickers in all picking lines. Table 5 reveals that the proposed method provides a relatively balanced picking line for the pick-and-pass system in the experiment since the mean blocking ratio in general is less than 11.48%, where the mean blocking ratio, defined as (the mean blocking time of a picker/total simulation time) $\times$ 100%, includes the ratio of waiting time for the replenishment process and the handling of the next order.

Fig. 12 further illustrates the workload of each zone for all scenarios. Clearly, the curve generated by HG is relatively smooth

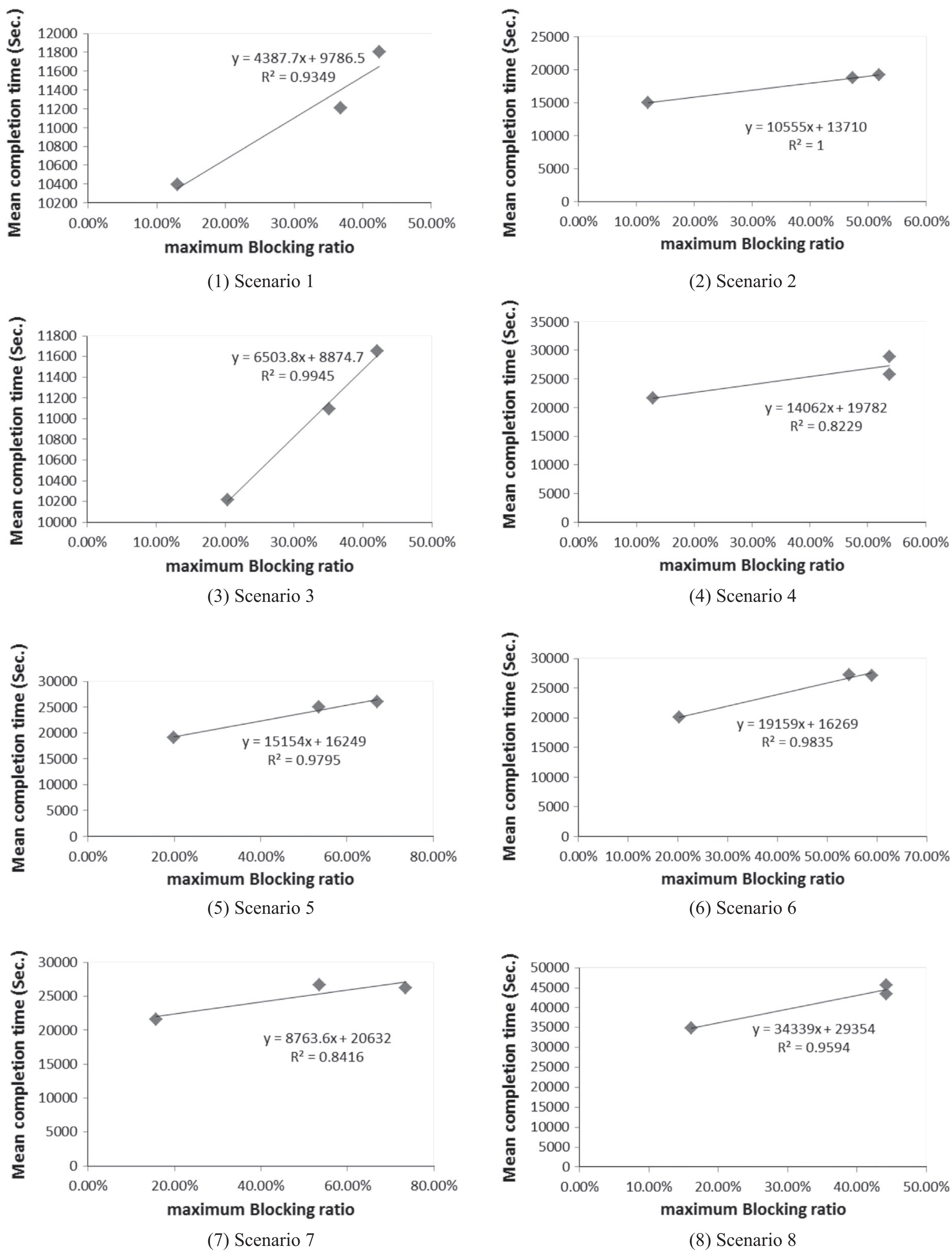


Fig. 13. Relationship between mean completion times and blocking ratios for all scenarios in the experiment.

because it strives to balance the workload of zones. In other words, the greater the amplitude is, the worse the performance becomes, as shown in both Fig. 12 and Table 4. As in a manufacturing process, the bottleneck zone in a picking line slows or limits the performance of a picking process. The blocking ratios of bottleneck zone of each picking line under all scenarios are listed in Table 6. For HG policy, the maximum blocking ratio is about 20% under scenario 6 while the FG policy reaches up to 73.42% in the greatest case. The RG, which is applied widely in practice, also yields 59.03% in Scenario 6. Fig. 13 shows the relationship between the mean completion time and the maximum blocking ratio in the system for all scenarios. The square of the correlation coefficient of each scenario being greater than 82% implies that bottleneck zones indeed affect the performance of order picking operation. Therefore, an optimal assignment policy has to consider the line balance for pick-and-pass system.

5. Conclusions

In a pick-and-pass system with multiple pickers, the line imbalance and the emergency replenishments could increase the possibility of idleness in the line. This paper presents a formulation for the determination of upper bound on the storage space size to avoid shortage of SKUs to be picked. Then, a storage assignment heuristic algorithm is developed based on the genetic algorithm to decide the location of SKUs with different storage space size and picking frequency. Consequently, the serial picking line can be leveled to significantly improve throughput of the system. The results of the numerical experiment indicate that the throughput of the proposed heuristic method is statistically better than those of the random policy and the policy of Gagliardi et al. (2008).

There are two possible extensions of the research developed in this paper. First, the assumption that all SKUs have the same size and weight can be relaxed to study the impact of other picking modes because a variety of sizes of SKUs exists in the real world. Secondly, the relation between storage assignment and other decision problems in the design and control of warehouse order picking process including layout design, routing and order batching (De Koster et al., 2013; Petersen & Gerald, 2004) could also be exploited in a manual order picking system.

References

Abdel-Hamid, A. A. A., & Borndörfer, R. (1994). *On the complexity of storage assignment problems*. Berlin: Konrad-Zuse-Zentrum für Informationstechnik.

Benjamin, P., Bart, V., & Denis, D. (2003). Dynamic ABC storage policy in erratic demand environments. *Jurnal Teknik Industri, 5*, 1–12.

Bottani, E., Cecconi, M., Vignali, G., & Montanari, R. (2012). Optimisation of storage allocation in order picking operations through a genetic algorithm. *International Journal of Logistics Research and Applications: A Leading Journal of Supply Chain Management, 15*(2), 127–146.

Brynzer, H., & Johansson, M. I. (1995). Design and performance of kitting and order picking systems. *International Journal of Production Economics, 41*, 115–125.

Chan, F. T. S., & Chan, H. K. (2011). Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. *Expert Systems with Applications, 38*, 2686–2700.

Chen, M. C., & Wu, H. P. (2005). An association based clustering approach to order batching considering customer demand patterns. *Omega, 33*, 333–343.

Chuang, Y. F., Lee, H. T., & Laih, Y. C. (2012). Item-associated cluster assignment model on storage allocation problems. *Computers & Industrial Engineering, 63*(4), 1171–1177.

De Koster, R. (1994). Performance approximation of pick-to-belt order picking systems. *European Journal of Operational Research, 72*, 558–573.

De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2013). Design and control of warehouse order pickimg: A literature review. *European Journal of Operation Research, 182*, 481–501.

Falkenauer, E. (1998). *Genetic algorithms and grouping problem*. New York, NY: John Wiley & Sons.

Flexsim simulation software user guide 4. (2007). Flexsim Software Products Inc.

Frazelle, E. H., & Sharp, G. P. (1989). Correlated assignment strategy can improve any order-picking operation. *Industrial Engineering, 21*, 33–37.

Gagliardi, J. P., Ruiz, A., & Renaud, J. (2008). Space allocation and stock replenishment synchronization in a distribution center. *International Journal of Production Economics, 115*, 19–27.

Glock, C. H., & Grosse, E. H. (2012). Storage policies and order picking strategies in U-shaped order-picking systems with a movable base. *International Journal of Production Research, 50*(16), 4344–4357.

Grosse, E. H., Glock, C. H., & Jaber, M. Y. (2013). The effect of worker learning and forgetting on storage reassignment. Decisions in order picking systems. *Computers & Industrial Engineering, 66*, 653–662.

Holland, J. (1975). *Adaptation In Natural and Artificial Systems*. Ann Arbour: The University of Michigan Press.

Hsu, C. M., Chen, K. Y., & Chen, M. C. (2005). Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Computers in Industry, 56*, 169–178.

Jane, C. C., & Laih, Y. W. (2005). A clustering algorithm for item assignment in a synchronized zone order picking system. *European Journal of Operational Research, 166*, 489–496.

Jewkes, E., Lee, C., & Vickson, R. (2004). Product location, allocation and server home base location for an order picking line with multiple servers. *Computers & Operations Research, 31*, 623–626.

Kovåks, A. (2011). Optimizing the storage assignment in a warehouse served by milkrun logistics. *International Journal of Production Economics, 133*(1), 312–318.

Malmborg, C. J. (1995). Optimization of cubic-per-order index layout with zoning constraints. *International Journal of Production Research, 33*, 465–482.

Maloney, D. (2000). Seeing the light. *Modern Materials Handling*, August, 49–53.

Muppani, V. R., & Adil, G. K. (2008a). A branch and bound algorithm for class based storage location assignment. *European Journal of Operational Research, 189*, 492–507.

Muppani, V. R., & Adil, G. K. (2008b). Efficient formation of storage classes for warehouse storage location assignment: Simulated annealing approach. *Omega, 36*, 609–618.

Pan, J. C. H., & Wu, M. H. (2009). A study of storage assignment problem for an order picking line in a pick-and-pass warehousing system. *Computer & Industrial Engineering, 57*, 261–268.

Park, C., & Seo, J. (2009). Mathematical modeling and solving procedure of the planar storage location assignment problem. *Computers & Industrial Engineering, 57*, 1062–1071.

Park, C., & Seo, J. (2010). Comparing heuristic algorithms of the planar storage location assignment problem. *Transportation Research Part E, 46*, 171–185.

Petersen, C. G., & Gerald, A. (2002). Considerations in order picking zone configuration. *International Journal of Operations & Production Management, 27*, 793–805.

Petersen, C. G., & Gerald, A. (2004). A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics, 92*, 11–19.

Rouwenhorst, B., Reuter, B., Stockrahm, V., van Houtum, G. J., Mantel, R. J., & Zijm, W. H. M. (2000). Warehouse design and control: Framework and literature review. *European Journal of Operational Research, 122*, 515–533.

Rylander, B., Foster, J., (2001). Computational complexity and genetic algorithms. *Proceedings of the World Science and Engineering Society's Conference on Soft Computing, Advances in Fuzzy Systems and Evolutionary Computation*, 248–253.