Enterprise Information Systems

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/teis20

Using a heuristic multi-objective genetic algorithm to solve the storage assignment problem for CPS-based pick-and-pass system

Mengru Tu , Ming-Feng Yang , Sheng-Long Kao , Feng-Cheng Lin , Ming-Hung Wu & Cheng-Kuan Lin

To cite this article: Mengru Tu , Ming-Feng Yang , Sheng-Long Kao , Feng-Cheng Lin , Ming-Hung Wu & Cheng-Kuan Lin (2020): Using a heuristic multi-objective genetic algorithm to solve the storage assignment problem for CPS-based pick-and-pass system, Enterprise Information Systems, DOI: 10.1080/17517575.2020.1811388

To link to this article: https://doi.org/10.1080/17517575.2020.1811388

Published online: 21 Sep 2020.

ENTERPRISE INFORMATION SYSTEMS
https://doi.org/10.1080/17517575.2020.1811388

Using a heuristic multi-objective genetic algorithm to solve the storage assignment problem for CPS-based pick-and-pass system

Mengru Tu[a,b], Ming-Feng Yang[a,b], Sheng-Long Kao[a,b], Feng-Cheng Lin[c], Ming-Hung Wu[d] and Cheng-Kuan Lin[a]

[a]Department of Transportation Science, National Taiwan Ocean University, Keelung, Taiwan; [b]Center of Excellence for Ocean Engineering, National Taiwan Ocean University, Keelung, Taiwan; [c]Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan; [d]National Audit Office, Taipei, Taiwan

ABSTRACT
With the advancement of AI and the widespread application of IoT and cloud computing, the pick-and-pass system (PKPS) can potentially be transformed into a cyber-physical system (CPS) based intelligent warehouse picking system. This paper proposes a CPS-based PKPS with a heuristic multi-objective genetic algorithm to solve the NP-hard storage assignment problem (SAP) for order picking operations in an e-commerce-based warehouse. The proposed algorithm considers both the workload balance between picking lines and emergency replenishment during picking operation. Finally, the study shows that the proposed algorithm is effective in improving the efficiency of picking operations based on software simulation.

ARTICLE HISTORY
Received 9 November 2019
Accepted 13 August 2020

KEYWORDS
Cyber-physical system; storage assignment problem; pick-and-pass system; multi-objective genetic algorithm; computer-aided picking systems

1. Introduction

Due to the convenience of internet accessibility and inexpensive logistics services, customers can order all kinds of favourite products anytime, anywhere from the internet using personal computers or mobile devices. Such changes have led to the rapid rise of e-commerce and the substantial growth of online retailers (Leung et al. 2018). In such an environment, the vast number of orders with a different amount of product types and small lot sizes of each order exerts a significant impact on warehousing operators. To tackle the challenges of e-commerce logistics operations, many e-commerce firms improved or upgraded their warehouse system for e-commerce distribution centres, from manual and paper-and-pencil operations to the use of computer-aided picking systems (CAPS) and the fully automated picking operations. According to De Koster et al. (2007), the order picking system can be divided into five types: picker-to-part, put-to-light system, parts-to-picker, automated picking, and picking robots. The pick-and-pass system (PKPS) is a variant of picker-to-part systems. With the development of e-business,

CONTACT Mengru Tu tuarthur@email.ntou.edu.tw Department of Transportation Science, National Taiwan Ocean University, Keelung, Taiwan

the PKPS is vital for the fast distribution of recurrent small orders of stock-keeping units (SKUs) (Pan et al. 2015).

To enhance the automation of PKPS, many firms adopted IoT and cloud technologies in their warehouse systems. The technology advancement of cloud computing and machine learning makes it possible to efficiently handle big data generated from IoT devices and multiple data sources (Chen et al. 2015; Gladence, Karthi, and Anu 2015; Yang et al. 2018b; Yang et al. 2019; Tu et al. 2018a; Chen 2019), secure critical data (Yang et al. 2017; Yang et al. 2018a), and support applications with dynamic computing resource requirements (Chen et al. 2019, 2020; Lu et al. 2020). Many applications using heuristic algorithms were also implemented on the cloud to solve large-scale optimisation problems (Lin et al. 2016; Lin et al. 2016; Guo and Shen 2017; Singh and Kumar 2019; Chiang et al. 2019; Maleki 2019). Integrating technologies of IoT and cloud computing, a cyber-physical system (CPS) can bridge the divide between the physical space (IoT system) and digital space (cloud platform) for PKPS. CPS utilises IoT sensors and actuators to collect real-time operational data of the physical world and conduct intelligent control to adjust its system behaviour in response to changing conditions and environment (Lin, Sedigh, and Miller 2010; Verl, Lechler, and Schlechtendahl 2012; Tu et al. 2018b). In other words, the physical part of CPS contains computing hardware such as sensors and actuators, while software components are implemented in the computational part of CPS (Verl, Lechler, and Schlechtendahl 2012). Hence, based on the multichannel communications between lower-level devices (sensors/actuators) and higher-level decision-making systems, a CPS can assist context-aware and situation-aware controlling model (Wang, Törngren, and Onori 2015). Thus, CPS can transform traditional PKPS into an industry 4.0 based intelligent PKPS in a warehouse environment.

A PKPS is often adopted for handling small-to-medium size SKUs and is also termed a progressive zoning system. In a progressive zoning system, a picker usually picks one part of a customer order in one zone and hands over the container with picked items to the next picking operator in the next zone until that order is fulfilled (Pan et al. 2015; Pan and Wu 2009). Thus the PKPS is ideal for warehouse operations under e-commerce business models.

In order to make the PKPS more efficiently, the optimisation of storage assignments in a warehouse is a critical and nontrivial issue (Muppani and Adil 2008). A storage assignment policy aims to solve the storage assignment problem (SAP) and provide a method for allocating SKUs in an e-commerce warehouse. An excellent storage assignment policy can make the picking operation more efficient and also reduce the warehouse storage waste, which can simultaneously decrease order picking time and reduce the total operation cost. The mathematical model of SAP has been proven to be an NP-hard problem (Frazelle and Sharp, 1989). Thus, a vast number of methods based on heuristic algorithms have been proposed to search for the optimal solution for SAP.

Also, different kind of storage policy has been proposed and widely used in the warehouse. (Petersen and Gerald 2002) showed that the storage assignment policy could significantly affect the average travel distance within a picking zone. Random and frequency-based policies are two commonly used storage assignment policies adopted by many warehouse systems (Pan et al. 2015). The random storage assignment policy is frequently used as a benchmark to appraise the performance of alternative policies. The frequency-based storage assignment policy assigns picking frequency to SKUs where an SKU with higher picking frequency will be assigned closer to a storage location near the Input/Output point.

The Genetic algorithm (GA) has the advantage of completely automatic computing process and benefit of aiding in local minima avoidance; thus, many research use GA to obtain an approximate best solution instead of other conventional search methods (Mary Gladence and Ravi 2013; Shakshuki, Malik, and Sheltami 2014). GA technique has also been used for tackling SAP (Pan et al. 2015; Pan and Wu 2009) and optimising order picking and replenishment operations in the warehouse (Leung et al. 2016; Poon et al. 2011; Mendes, Gonalves, and Resende 2009). Bottani et al. (2012) explored the use of a GA to optimise the allocation of SKUs in a warehouse, with the eventual goal of decreasing the travel time of pickers and streamlining order picking jobs. (Cai et al. 2016) proposed a storage assignment policy base on storage frequency with consideration of workload balance. Pan et al. (2015) developed a heuristic GA for SAP in a PKPS considering workload.

Most previous studies assumed that the SKUs would not run out during picking operations while developing storage assignment policies. However, in reality, some SKU often runs out of stock, and the emergency replenishment occurs, which delays the total picking time. In this paper, we consider the emergency replenishment as a vital factor. Once the SKUs runs out in a picking zone, the PKPS must start the emergency replenishment processes and stop the picking operation until the replenishment is done. Another critical factor often overlooked in studying picking strategies is the balance of workload among picking lines or picking zone. The difference in workload could affect the picking operation. When the workload of a picking line is too large, it will cause congestion on the picking line, delay the total picking time, and increase the cost of picking. Jane (2000) developed algorithms to allot ordered items to the picking zone and to balance the job loading among picking zones, which is referred to as a relay picking system. Jewkes, Lee, and Vickson (2004) presented an efficient dynamic programming algorithm to determine the optimal SKUs allocation for a picking line with multiple pickers. Table 1 summarises the key criteria for this and previous research studies regarding SAP. The characteristic of this paper is similar to that of Pan et al. (2015), where both study the pick-and-pass warehouse system and consider the workload balance and replenishment problems. Pan et al. (2015) assumed that the quantity of SKU stocks in the warehouse is constant and equal to the upper limit calculated by the proposed formula. Assuming the constant SKU stock quantity is unrealistic in practice. Thus, we relax the above assumption and calculates the quantity of SKU stocks based on estimated market demand, which is closer to the real situation in a PKPS.

This paper aims to design a CPS-based PKPS framework using a proposed heuristic multi-objective GA (MOGA) with a random coefficient to solve SAP concerning picking line balance and emergency replenishment in an e-commerce based warehouse. The random coefficient in fitness function can let the solutions spread widely in Pareto optimal solution space. Moreover, to preserve the uniqueness of those solutions found in the solution space, this paper also develops an elite preservation mechanism in the algorithm to keep those solutions. The proposed storage assignment algorithm of the

Table 1. Comparison of criteria for research studies regarding SAP.

Criteria		Jewkes, Lee, and Vickson (2004)	Bottani et al. (2012)	Jane (2000)	Pan et al. (2015)	This paper
Warehouse Type	Pick-and-pass			●	●	●
	Picker-to-part	●	●			
Performance Measurement	Workload_balance			●	●	●
	Travel_time	●	●			
	Replenishment				●	●
Problem Solving Method	Heuristic	●		●		
	GA		●		●	
	MOGA					●

CPS-based PKPS will collect data from CAPS regularly to reflect the environment changes in the warehouse picking area.

The salient contributions of this paper are twofold: (i) We propose a novel CPS-based pick-and-pass system (PKPS) framework to enhance the intelligence of e-commerce warehouse picking operations. (ii) Under the architecture of CPS and considering the workload balance between picking lines and emergency replenishment, we developed a heuristic random weight multi-objective genetic algorithm to solve the storage assignment problem (SAP) and improve the efficiency of picking operation.

The rest of this paper is organised as follows. Section 2 presents the system model of the proposed CPS-based pick-and-pass system with its underlying assumptions. In Section 3, we design and develop a GA-based storage assignment algorithm for a CPS-based pick-and-pass system. Experimental design and simulation results are presented and discussed in Section 4. Finally, Section 5 concludes the paper.

2. CPS-based pick-and-pass system

2.1. *Enterprise system architecture for CPS-based pick-and-pass system*

In this paper, the warehouse with a PKPS is divided into two separate but adjacent areas as the reserve area and forward area. There are several picking lines in the forward area, and all picking operations are performed in this area. Each picking line is divided into multiple small zones connected by a roller conveyor. This paper calls these zones in the picking lines as 'working cells.' It may be called 'picking cell' in practice, but we use the term 'working cell' because it can entail both picking and packing operation in a pick-and-pack warehouse system. This paper assumes that each working cell store different SKUs, but one SKU can only be stored in one working cell, and each working cell has only one picker. After receiving the order from the computer-aided picking system (CAPS), operators in each working cell start to pick corresponding SKU products related to the order into a transport plate and deliver to the next working cell to fulfill the customer order. In practice, CAPS devices are installed in the warehouse for each working cell. CAPS can improve picking productivity and decrease picking errors. CAPS will help a picker find the SKUs in a working cell by light up the light indicator modules installed on all (storage) racks and display the precise amount of each SKU to be picked in any working cell. With the rise of AI and wild application of and cloud and IoT technologies in the logistics industry, the trend has become evident to transform a traditional PKPS into an intelligent warehouse through a CPS-based architecture.

The enterprise system architecture for the proposed CPS-based PKPS is described in Figure 1. In the proposed architecture, the CAPS is assumed to be the existing enterprise system installed on server 1 with many CAPS devices deployed on storages racks within each working cell. The CPS-based PKPS is a new information system decoupled from CAPS and installed on server 2. It first receives the data feedback from CAPS as well as predicted data, such as estimated SKU demand data, from the business data centre. After computing by the storage assignment software module, the CPS finds a near-optimal solution for the SAP in the CPS-based PKPS and gives instructions to the CAPS. The cycle of computation depends on the business needs of a company. After the instructions are given to the CAPS form CPS, the picker in the warehouse can retrieve items from a rack based on the SKU quantity information revealed on a rack and then confirm the picking job by pressing the lighted button. As the picking job for processing one order in the designated picking cell is concluded, the CAPS then directs the picking instruction of the next order to all light indicator modules in the same cell.

CAPS can simplify picking tasks, and this paper assumes that the picking time for each kind of SKUs is constant. Since the picking time for each SKU is constant, high picking frequency in the same working cell will make the workload too large for the cell and cause the congestion. The congestion will delay the total picking time and increase the cost of picking. To prevent the congestion caused by picking operation in a single working cell, the workload of all working cells has to distribute equitably. On the other hand, the capacity of racks in the working cell is limited, and the extra SKUs will storage in the

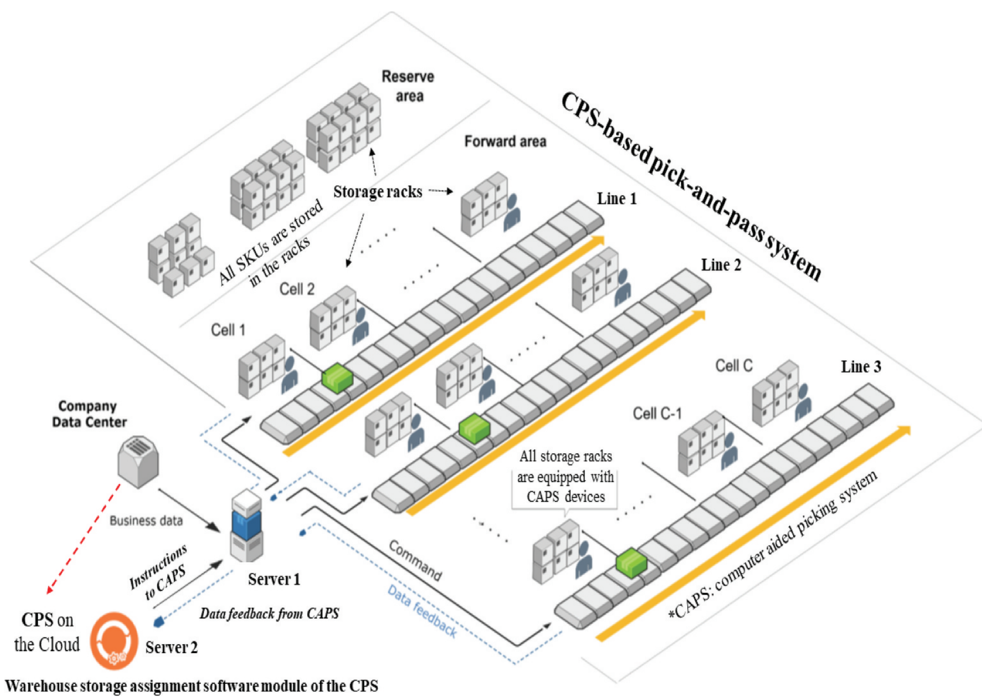


Figure 1. The enterprise system architecture for CPS-based PKPS.

reserve area. Once the stocks in working cells are insufficient during picking operation, the picking operation stops and starts the emergency replenishment. This paper also assumes that the time of the emergency replenishment is constant for all SKUs, and the whole picking operation will keep freezing until the emergency replenishment is done. Hence, the total number of stock needs to approach customer demand as much as possible if we want to prevent replenishment operation. The challenges described above are related to SAP, and the warehouse storage assignment software module in the proposed CPS-based PKPS architecture (as shown in Figure 1) will be used to solve the SAP and help improve logistics efficiency for the e-commerce based warehouse. The following sections will discuss the mathematical formulation of SAP and the design of a heuristic multi-objective GA to solve SAP.

2.2. Formulation of the SAP

SAP is an NP-hard problem. To solve the SAP for the CPS-based PKPS warehouse system, we proposed a mathematical model to describe the SAP. The following sections discuss the formulation of the mathematical model.

2.2.1. Assumption and notations

This paper adopts the following assumptions which are made by Pan et al. (2015) in the CPS-based PKPS warehouse system under study:

(1) The size and weights for all SKUs of all the orders are assumed to be the same.

(2) The time is assumed to be constant for picking an SKU from a rack.

(3) Each SKU is not dependent on other SKUs in an order.

(4) Each SKU is allotted in one picking cell only, i.e., an SKU can be handpicked by one picker only.

(5) Each rack stores only one SKU.

(6) The travel time in a picking cell is considered negligible as the storage racks are usually thin and low.

(7) The same number of racks is equally installed in all picking cells.

(8) When CPS detects any shortage of SKU, the task of emergency replenishment for that SKU is immediately carried out and the time of SKU replenishment is a constant.

(9) Each item is independent of the other items within an order (Pan and Wu 2009; Jewkes, Lee, and Vickson 2004; Jarvis and McDowell 1991).

The notations used in this paper are defined in Table 2.

2.2.2. A mathematical model to formulate the SAP

In a pick-and-pass system, idle time may arise due to the cell imbalance and the emergency replenishment of ordered items. When ordered items are equally allocated in each picking cell based on real-time customer demand data, we can significantly decrease the idle time of picking tasks and improve the system performance. Thus, the overarching goal of the proposed CPS-based PKPS is to minimise the waste of wait time among pick-and-pass operations and thus improve the order picking productivity and

Table 2. The notations used in this paper.

I	Maximum number of picking lines;
i	Picking line index, i = 1, 2, ….., I;
C	The maximum number of working cells;
c	Working cell index, c = 1, 2, … ., C;
R	The maximum number of racks;
r	Rack index, r = 1, 2, …, R;
K	Maximum number of SKUs;
k	SKU index, k = 1, 2, … … ., K;
n_k	Mean number of SKU k to be picked;
p_k	Demand rate of SKU k to be picked;
A	The capacity of a rack;
T_P	Time to finish a picking operation;
T_r	Time to finish an emergency replenishment;
W	Mean workload;
f_v	fitness value
w_1, w_2	the random weight coefficients generated from a uniform distribution between 0 and 1.

efficiency of the entire system. Hence, this paper formulates a mathematical model for SAP in a CPS-based PKPS with multiple working cells as follows:

Minimise

$$Z = w_1 T_r \sum_{k=1}^{K} \left| A \sum_{c=1}^{C} \sum_{r=1}^{R} X_{crk} - n_k p_k \right| + w_2 \sum_{c=1}^{C} \left| \sum_{r=1}^{R} \sum_{k=1}^{K} n_k p_k X_{crk} T_p - W \right| \quad (1)$$

Subject to

$$\sum_{k=1}^{K} X_{ckr} = 1 \mathrm{for} c = 1, \ \ldots \ C; , \mathrm{and} r = 1, \ \ldots \ , R. \quad (2)$$

$$\sum_{c=1}^{C} \sum_{r=1}^{R} X_{crk} - \sum_{r=1}^{R} X_{crk} \leqq 0 \mathrm{for} c = 1, \ldots C; \ k = 1, \ \ldots \ , \ K. \quad (3)$$

$$X_{ckr} = 0 \mathrm{or} 1 \mathrm{for} c = 1, \ \ldots \ , C; \ k = \ 1, \ \ldots \ , K, \ and r = 1, \ \ldots \ , R \quad (4)$$

$$w_1 + w_2 = 1; 0 \le w_1 \le 1 \mathrm{and} 0 \le w_2 \le 1 \quad (5)$$

In formulation model, where X_{ckr} = 1, means SKU k is assigned to rack r of cell c; and 0, otherwise. The objective function (1) described two subgoals in the model under the overarching goal of minimising the waste of wait time among pick-and-pass operations resulting from emergency replenishment and imbalance of workload among picking (working) cells. The two sub-goals are described below. The former part (subgoal 1) means the absolute difference between storage spaces for an SKU in the system and expected picking-demand. To decrease replenishment, the storage space for an SKU should close to the expected picking demand (customer needs) as much as possible because of the limited storage racks available in the warehouse.

This paper uses T_r to enable the first part of Equation (1) to have the same unit in terms of a real-time unit (seconds). The latter part (subgoal 2) depicts the summing the absolute deviations from measuring the workloads from all pickers. Since the imbalance of workload in each picking zone would cause severe delay on order throughput time (Pan and Wu 2009), thus, fairly allocating of SKUs in each picking zone is another objective in this paper. Therefore, the objective function of Equation (1) serves as the performance index for evaluating the operational efficiency of the proposed CPS-based pick-and-pass system. The smaller the value of Z in Equation (1) indicates a better system performance. Equation (2) makes sure that a single rack stores only one kind of SKU. Equation (3) ensures that an SKU will not be assigned to multiple picking zones in the system. Moreover, w_1 and w_2 in Equation (1) are two weight coefficients associated with the balance of workload and the number of replenishment, respectively. The solution of Equation (1) is Pareto-optimal if all of the weights are positive (Miettinen 1999).

Besides, to formulate the objective function and its constraints as described above, the storage space layout for the number of racks to store SKUs in a pick-and-pass operation working cell must be determined before making the storage location assignment decision. To decrease the occurrence of replenishment operations, the numbers of all SKUs are expected to match the expected picking demand. (Pan et al. 2015) defined the optimal upper bound of storage space for SKU k in the pick-and-pass warehouse system. We thus adopt the formulation of the storage space requirement proposed by (Pan et al. 2015) in equation 5 shown below:

$$Y_k \leq \frac{n_k p_k CR}{\sum_{k=1}^{K} n_k p_k} \; for\, k = \; 1, \; 2, \; \ldots \; , K \tag{6}$$

Where $Y_k = \sum_{i=1}^{I} \sum_{r=1}^{R} X_{crk}$ denotes the total storage space to store SKUs in a CPS-based PKPS.

Since the mathematical model to solve SAP of a PKPS is a classical SAP and has been proven by many research to be an NP-hard problem (Abdel-Hamid and Borndorfer, 1994; Frazelle and Sharp, 1989), we thus proposed a heuristic storage assignment algorithm based on random weight multi-objective GA to improve the solution quality.

Figure 2 describes the high-level system framework of the proposed heuristic storage assignment algorithm with the input-output relationship of the algorithm. The system starts from taking the necessary parameters into the core algorithm, including warehouse facility(i, j, r), data of SKUs (k, n_k, p_k), and the parameters set for the MOGA. Data of SKUs is related to the picking operation and will be collected during the routine work daily by CAPS and recorded in the server. The algorithm would retrieve the data related to SKUs from the server. MOGA parameters like crossover rate, mutation rate can be adjusted by system engineers or system developers according to the need of a firm during system development or modification. After the calculation, the algorithm will generate a picking recommendation comprising an optimal quantity to store in a warehouse for each SKU in each working cell. Figure 2 illustrates an example of allocating the quantity of each SKU to its corresponding storage space (cell) in a warehouse.

3. Design and development of a GA-based storage assignment algorithm

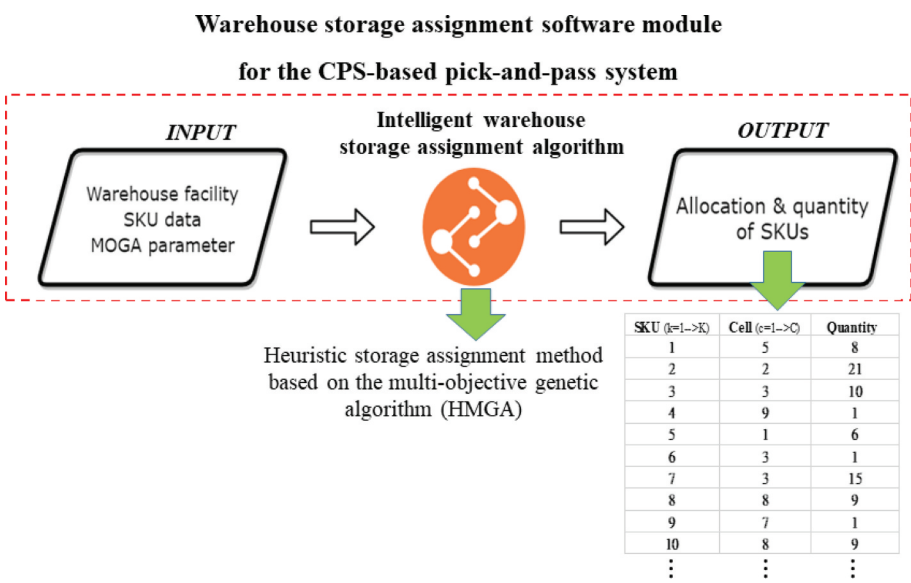
This section will design and develop a storage assignment algorithm based on heuristic random weight multi-objective GA for the proposed CPS-based pick-and-pass warehouse system.

3.1. Design of the storage assignment algorithm

This paper aims to balance the workload of each warehouse working cell and reduce the number of replenishment times for a CPS-based pick-and-pass warehouse system, instead of finding a single optimal solution. Thus, we present a heuristic storage assignment method for CPS-based PKPS based on a multi-objective GA to satisfy the two objectives described above. We term the proposed storage assignment method based on the heuristic multi-objective GA as HMGA in this paper. The algorithmic procedure of HMGA is presented in Figure 3. The following sections will discuss the major procedures of Figure 3 in detail.

3.1.1. Chromosome encoding

To alleviate the frequency of emergency replenishment during the picking operation and balance the workload among different picking lines and working cells in each picking line at the same time, this paper decodes the stock quantity and the allocation of SKUs into twin-chromosome to simultaneously fulfill the two objectives in the computation. In HMGA, each set of solutions is compiled into two chromosome α and β. The chromosomes will crossover and mutate to produce a new solution. We can repeat these procedures and select the best solution to search for Pareto-optimal solutions. The chromosome encoding design for SAP of this paper is described in Figure 4. If a total of



SKU (k=1->K)	Cell (c=1->C)	Quantity
1	5	8
2	2	21
3	3	10
4	9	1
5	1	6
6	3	1
7	3	15
8	8	9
9	7	1
10	8	9
⋮	⋮	⋮

Figure 2. The input-output description of our proposed storage assignment algorithm.

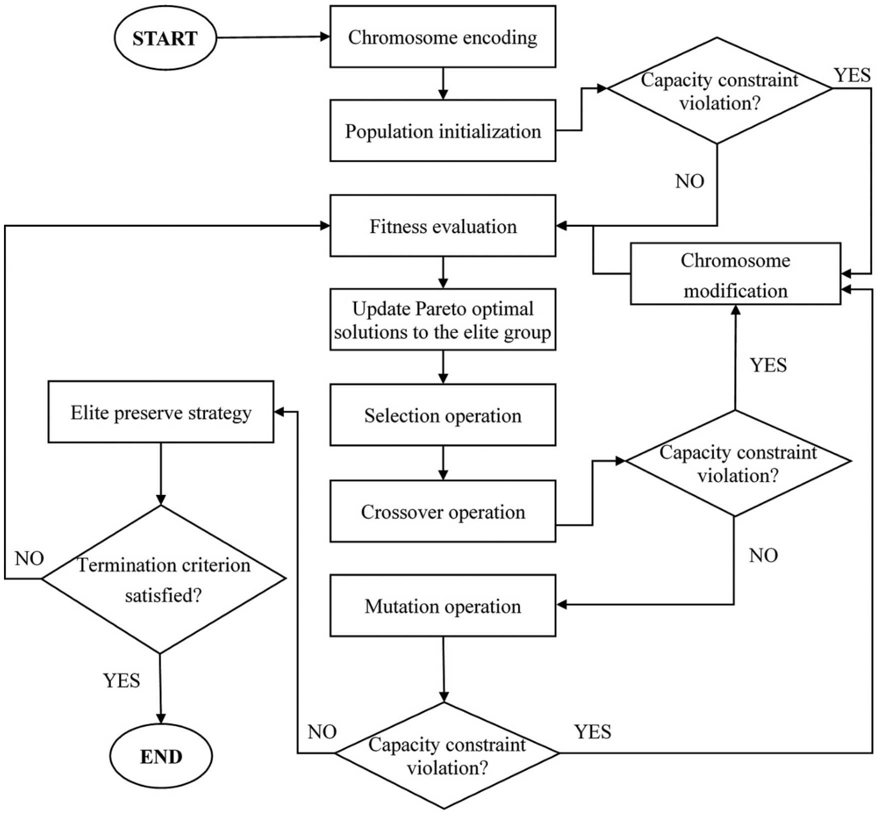


Figure 3. The flow chart for the proposed HMGA.

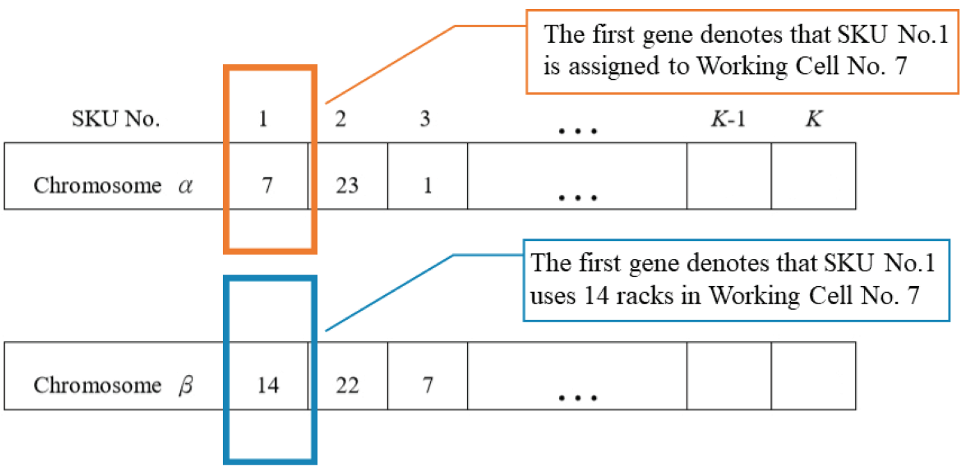


Figure 4. The twin-chromosome encoding design for the GA-based storage assignment algorithm.

k kinds of SKUs need to be stored, the chromosome contains k genes. The gene in chromosome α denotes the assignment of SKU in the working cell, where the value of each gene in chromosome α indicates the storage zone for a stock number of an SKU, as shown in Figure 4. As for chromosome β, the gene in chromosome β denotes the storage space that an SKU required in the forward area of a warehouse.

3.1.2. Fitness function

We devised a Fitness function to evaluate the fitness of each chromosome to the solution and to calculate the fitness value (f_v) by bringing the chromosome into the function. Chromosomes with the higher f_v have better performance in HMGA. The fitness function of HMGA is represented as follow:

$$f_v = w_1\left(\sum_{c=1}^{C}\left|\sum_{k=1}^{K}\sum_{r=1}^{R} n_r p_r X_{crk} - \sum_{k=1}^{K}\frac{p_k n_k}{C}\right|\right)^{-1} + w_2\left(\sum_{k=1}^{K}\left|A\sum_{c=1}^{C}\sum_{r=1}^{R} X_{crk} - n_k p_k\right|\right)^{-1} \quad (7)$$

Where, w_1 and w_2 are the random weight coefficients generated from a uniform distribution between 0 and 1. Since constant weights may cause the GA search in a particular direction, the random weights enable the GA to utilise various search directions to find Pareto optimal solutions (Murata and Ishibuchi 1995).

The first part of Equation (7) $\left(\sum_{c=1}^{C}\left|\sum_{k=1}^{K}\sum_{r=1}^{R} n_r p_r X_{crk} - \sum_{k=1}^{K}\frac{p_k n_k}{C}\right|\right)^{-1}$ is to articulate the balance of the workload between working cells. The second part of Equation (7) $\left(\sum_{k=1}^{K}\left|A\sum_{c=1}^{C}\sum_{r=1}^{R} X_{crk} - n_k p_k\right|\right)^{-1}$ is to measure the gap between the stock in the forward area of a certain SKU and the estimated picking demand of that SKU. To determine the performance of each solution, we will bring chromosome α into the first part of the Equation (7) and chromosome β into the second part of Equation (7) to separately calculate the value for each part and then combine the results from the two parts to obtain the final f_v.

3.1.3. Selection

In implementing the *selection* process (step 3), the roulette wheel selection operation is used to choose a chromosome from the current generation for the next round of genetic operation. Every chromosome in the population is assigned a probability, which is proportional to its fitness value. The fact a chromosome has higher fitness value implies that the higher the probability of that chromosome being selected to the next generation. Thus, the fitness function f_v will affect the probability of selection. This paper adopts the method proposed by Murata and Ishibuchi (1995) to determine the chromosome selection probability $p_{(x)}$ for random weight multi-objective GA of our GA model. The selection probability $p(x)$ of a chromosome x in population M is defined in Equation (7). This step is repetitive for selecting $M/2$ pair of chromosomes from current populations.

$$p_{(x)} = \frac{f_v - f_{min}(M)}{\sum_{x\in M}\{f_v - f_{min}(M)\}}$$

where

$$f_{min}(M) = min\{f(x)|x \in M\} \quad (8)$$

3.1.4. Crossover and mutation

Following the *selection* process is the *crossover* operation (step 4). Chromosomes between generations have a specific ratio of crossover. HMGA select genes from parent chromosomes and engenders a new offspring. This paper uses the two-point crossover operation, which is to randomly select two crossover points within two selected parent chromosomes and then exchange the genes of the two chromosomes between those points to generate two new offspring chromosomes, as illustrated in Figure 5. The chromosome α and chromosome β will do the crossover operation separately.

The *mutation* operation (step 5) is another essential operation in the HMGA after the *crossover* operation has performed. The *mutation* operation randomly selects a gene from the chromosome and alters the code of that gene having the surplus capacity in a warehouse working cell, as depicted in Figure 6. The *mutation* takes place with a certain probability in each generation and is used to explore new solutions in the solution space, avoiding the traps of falling into local optima solutions in HMGA. In the proposed HMGA, the *mutation* operation is required to modify the gene of chromosome having a surplus capacity in a warehouse working cell with top priority over other genes.

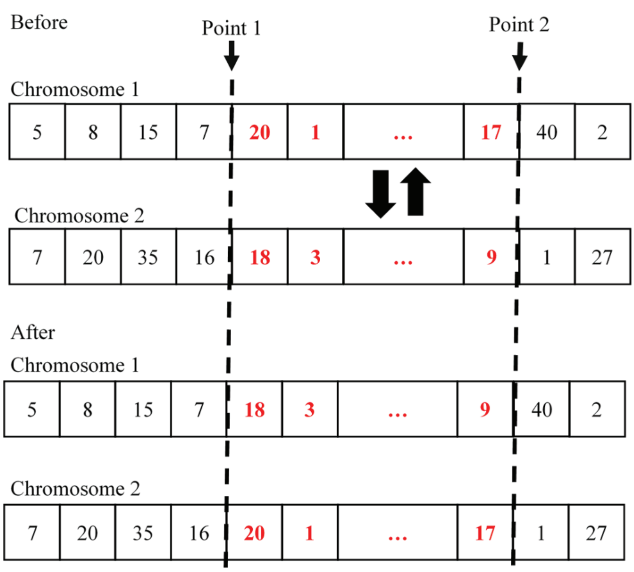


Figure 5. Crossover operation.

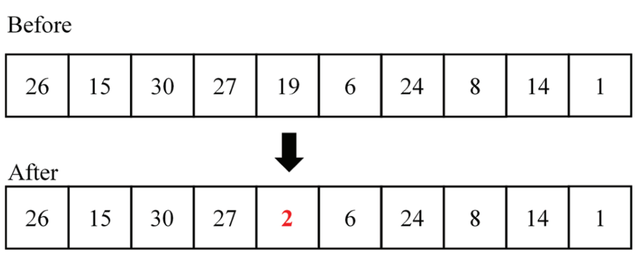


Figure 6. Mutation operation.

Specifically, for chromosome α, the mutation operation of HMGA alters the gene having a surplus capacity in a working cell. In the case of chromosomes β, the mutation operation of HMGA replace gene with a random value between the predefined upper_storage_limit and and R_m* upper_storage_limit, where R_m is a sensitive parameter used to decide the mutation range.

3.1.5. Elite preserve policy

When applying GA to solve multi-objective optimisation problems, it is difficult to find a single optimal solution. One general approach to solve problems regarding multi-objective optimisation is to discover all possible tradeoffs among multiple objective functions and present the set of Pareto optimal solutions to the decision-maker (Murata and Ishibuchi 1995). Our proposed HMGA also tries to solve multi-objective optimisation problems. Under this circumstance, a tentative set of Pareto optimal solutions must be preserved and stored at every generation in the execution of the multi-objective GA (Murata and Ishibuchi 1995). Figure 7 illustrates the distribution of solutions for HMGA, where multi-objective GA solution points with higher value to both objective 1 or objective 2 represent good performance in a certain direction. Some feasible solutions for which none of the objective functions can be improved in value without deterioration of at least one other objective function (Mavrotas and Florios 2013). These solutions are named Pareto-optimal solutions or non-dominated solutions. For example, point A in Figure 7 can be treated as a non-dominated solution, which performs well in objective 1 but shows average performance in objective 2. The elite preserve policy can preserve these non-dominated solutions into an elite group to avoid these solutions been removed in subsequent GA selection operation. In HMGA, the algorithm will randomly remove the genes in the chromosomes with a lower fitness value f_v and replace them with genes randomly picked from the Pareto-optimal solutions preserved in the elite groups.

3.1.6. Termination test

The algorithm runs until it reaches a predefined termination condition. In this paper, the termination criterion is when the number of generations in HMGA approaches the preset value, and the chromosome with the best performance in the last generation is chosen to be the optimal solution. For example, at the beginning of the algorithm, we set

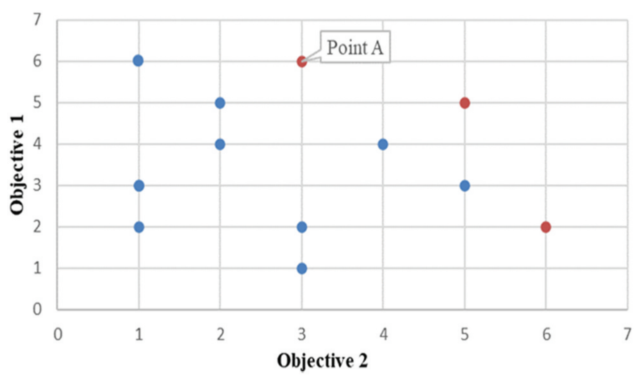


Figure 7. Example of Pareto optimal solution.

a maximum no. of generations (G), and initialise the generation $g = 0$. If all procedures have been executed, then the HMGA evaluates whether the number of generation equal to the maximum generations ($g = G$); if $g = G$, stop the algorithm; otherwise, repeat the algorithm ($g = g + 1$).

3.1.7. Chromosome modification mechanism

The population initialisation, crossover, and mutation operations in the proposed HMGA may cause the number of SKUs to exceed the capacity of racks in warehouse working cells. Therefore, this paper proposed a chromosome correction mechanism originally developed by Pan et al. (2015) to modify chromosomes after initialisation, crossover, and mutation operations, if there exists storage capacity overflow in a working cell (a violation of storage capacity constraints). All chromosomes need to be corrected to conform to the assumption, especially for the initial solutions that are generated randomly and the solutions generated by crossover and mutation operations. Adapted from the chromosome correction mechanism proposed by Pan et al. (2015), we developed the chromosome modification mechanism for this paper, and its processing steps are described in Table 3.

3.2. Implementation of the HMGA for the CPS-based pick-and-pass system

Based on the algorithm flow chart of Figure 3 and major genetic operations discussed in previous sections, the procedure of the proposed heuristic storage assignment algorithm based on random weight multi-objective GA (HMGA) for the CPS-based PKPS can be described in the following steps in Table 4.

4. Experimental design and simulation

4.1. Description of the experiment

To demonstrate the effectiveness of the proposed heuristic storage assignment policy based on HMGA, this paper compares the HMGA with random storage assignment policy (RND) and First-Come-First-Servestorage assignment policy (FCFS). RND storage assignment policy randomly assigns SKUs to different working cells in each picking line. RND serves as a baseline benchmark for evaluating algorithm performance. FCFS allocates SKUs according

Table 3. Chromosome modification mechanism.

The steps for chromosome modification mechanism
Step 0: Set working cell $wc = 1$.
Step 1: Check any occurrence of overflow in working cell wc. If any overflow been observed, go to Step 2; otherwise, go to Step 6.
Step 2: Find one SKU b using the least space in working cell wc.
Step 3: Find one working cell a using the least racks in the same picking line i.
Step 4: Move SKU b to working cell a. If overflow found in the working cell a, go to Step 5; otherwise, go to Step 1.
Step 5: Move SKU b back to working cell wc again, exchange SKU occupying the most space in the working cell a and working cell wc, then go to Step 2.
Step 6: If all working cells been checked without overflow, exit the chromosome modifying mechanism; otherwise, set $wc = wc + 1$ and go to Step 2.

Table 4. HMGA algorithm.

The processing steps of the proposed HMGA algorithm
Step 1: Determine the storage space for all types of SKU by Equation (5).
Step 2: Initialisation: Set population (M), number of generations (G), crossover rate (R_c), mutation rate (R_m), elite group (E), m = 1, g =0, E = 5.
Step 3: Randomly produce m chromosome in generation g.
Step 4: Modify the chromosomes m in generation g using chromosome modifying mechanism if there exists a violation of storage capacity constraints.
Step 5: If m = M, go to Step 6; otherwise, set $m = m + 1$ and return to Step 2.
Step 6: Compute fitness value f_v using Equation (6), v = 1, 2 . . .M.
Step 7: Find Pareto optimal solutions from generation g.
Step 8: Compare newly found Pareto optimal solutions with existing ones in elite group E and store solutions with better performance in E. (discarding inferior solutions in E)
Step 9: Create next-generation (g + 1) by selecting M chromosomes from generation g via Equation (7).
Step 10: Randomly select $M \times R_c$ chromosomes in the generation (g + 1) to crossover, correct the offspring, replace the parent chromosomes, and apply chromosome modifying mechanism if there exists a violation of storage capacity constraints
Step 11: Randomly select $M \times R_m$ in the generation (g + 1) chromosomes to mutation operation, and apply chromosome modifying mechanism if there exists a violation of storage capacity constraints.
Step 12: Randomly select $M_e$chromosomes in the generation (g + 1), and replace them with M_e chromosome chosen from E.
Step 13: If $g = G$, stop the algorithm and choose the chromosome with the best performance as approximate solutions; else, set $g = g + 1$ and return to Step 6.

to the arriving time of SKUs in the warehouse; earlier the SKU arrives earlier the SKU assigned. FCFS is a simple and widely adopted practice by many logistics operators.

In the experiment, we acquired the test dataset from three sources. The first test dataset generated from the provided HMGA programmed using Visual Basic Applications (VBA) for Microsoft Excel. We also used two VBA modules to generate another two datasets, the RND and FCFS, respectively. We first evaluated the sensitivity of HMGA and then fed the three datasets into the mathematical model of SAP (comprise of Equations (1)-(4)) to simulate the system performance of the CPS-based PKPS using HMGA under different scenarios. Using the simulation results, we compared the picking task performance among the three different storage assignment policies. All experiments are run on a notebook with 2.5 GHz Intel i5 CPU and 8 G RAM. The next section discusses the results of the numerical experiment.

4.2. Results and discussion

We first evaluate the sensitivity of HMGA using the fitness function (Equation(7)), the higher the fitness value, the better. Table 5 records the fitness value of HMGA under different GA parameter after running the algorithm for ten times. The results indicated that the calibration of the crossover rate had a remarkable impact on fitness value. The higher crossover rate leads to the high-frequency gene transfer in the algorithm; however, it may cause chromosome with better performance been crossover excessively, leading to lower fitness value. The experimental design for different scenarios is shown in Table 6.

Figure 8 displays the result of bringing each storage assignment policy into the mathematical model of SAP for ten runs. In the mathematical model of SAP, the smaller objective function (Equation (1)) value denotes the solution is closer to the optimal solution. Figure 8 shows that the simulation value of HMGA is smaller than RND and FCFS in all scenarios,

Table 5. Comparisons of f_v under different HMGA parameters (i = 2, j = 5, r = 60, c = 20, k = 50).

GA parameters		f_v Value		
Population (m)		20	30	40
	Mean	0.3754	0.3253	0.3070
	S.D.	0.1500	0.2841	0.1333
Crossover rate (R_C)		0.4	0.6	0.8
	Mean	0.3346	0.3253	0.2028
	S.D.	0.1515	0.2841	0.0595
Mutation rate (R_m)		0.01	0.03	0.04
	Mean	0.3211	0.3167	0.3253
	S.D.	0.1247	0.0598	0.2841

Table 6. Summary of the experimental scenarios (M = 30, R_c = 0.6, R_m= 0.04).

	Scenario No.				
System Specification	1	2	3	4	5
Picking lines (i)	2	2	3	3	5
Working cell (c) [per line]	5	5	5	10	10
No. of racks (r)	60	60	60	60	60
Capacity (c)	20	20	20	20	20
No. of SKU (k)	50	80	50	250	300

indicating HMGA performed better in term of operation efficiency, specifically represented by the mathematical model of SAP with the overarching goal of minimising the waste of wait time among pick-and-pass operations resulting from emergency replenishment and imbalance of workload among working cells. Finally, Table 7 shows the improvement profile for the HMGA against RND and FCFS; the unit of the result data in Table 7 is in seconds. Table 7 also indicates that HMGA has a significant advantage improvement over the other two methods, implying that the proposed HMGA can significantly enhance the performance of the CPS-based PKPS supporting e-commerce logistics operations.

5. Conclusion

In a traditional PKPS where multiple pickers perform picking and packing tasks, the emergency replenishments and workload imbalance could escalate the chance of job idleness in the picking line. The traditional storage assignment policy like RND and FCFS is unsuitable for the complex situation in e-commerce based on a warehouse system with large numbers of SKUs and in a variety of lot sizes. Workload balance among working cells and emergency replenishment could noticeably affect the operation efficiency in pick-and-pass warehouse systems. To improve the efficiency of picking operation, we develop a heuristic storage assignment policy based on random weight HMGA for a CPS-based PKPS. The most notable contributions of this paper are listed below:

(1) The proposed HMGA considers not only the SAP but also the replenishment problem and used the random-weight parameter in the fitness function to utilise various search directions to find Pareto optimal solutions.

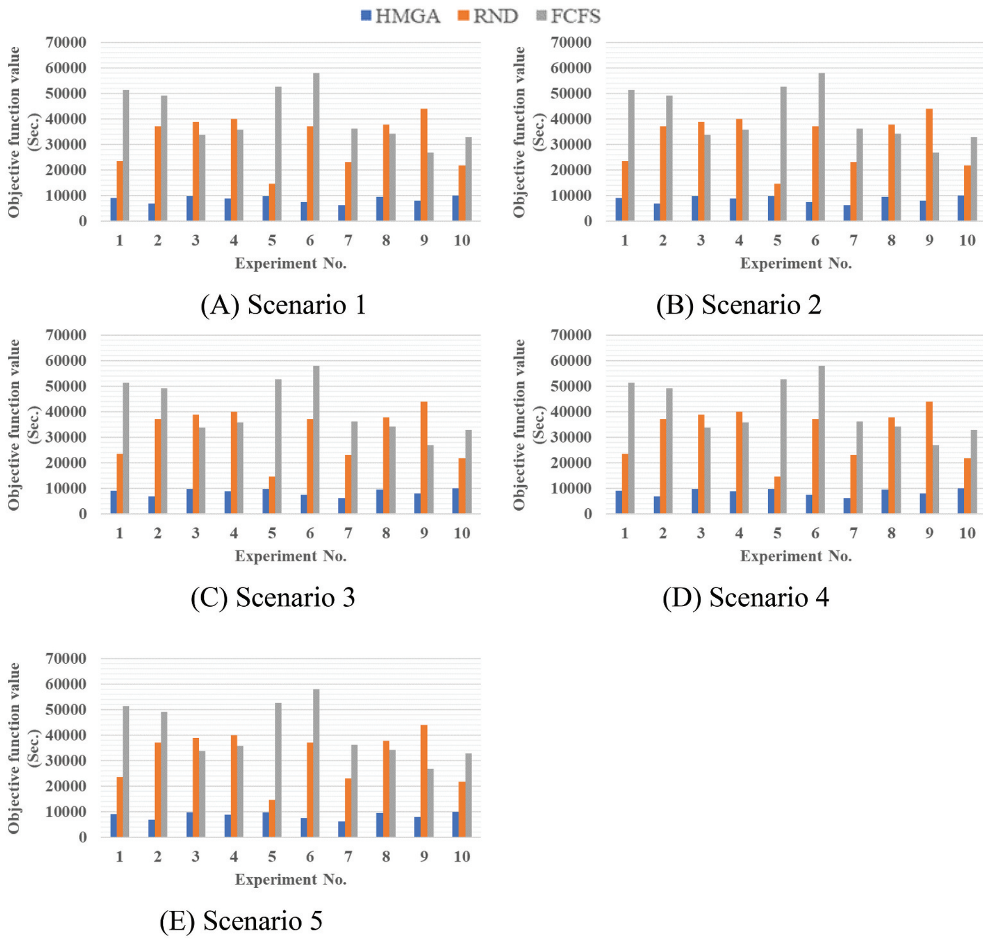


Figure 8. Graphical comparison of the simulation results under different storage assignment policies for all scenarios in the experiment.

(2) To mitigate the occurrence of emergency replenishment during the picking operation and to simultaneous balance the workload among different picking lines and working cells within each picking line, HMGA decoding the stock quantity and the allocation of SKUs into twin-chromosome to simultaneously fulfill the two objectives in the calculation.

(3) The numerical results of the experiments show that the over picking operation throughput of our proposed HMGA is better than that of RND and FCFS. Thus HMGA proved to be the best model for the CPS-based PKPS with the overarching goal of minimising the waste of wait time resulting from emergency replenishment and imbalance of workload among working cells.

With the rise of e-business, the PKPS has become popular and common in recent years. Incorporating the IoT technology in the logistics industry, we can transform the traditional PKPS into a CPS-based PKPS. A CPS-based PKPS with a well-designed storage assignment algorithm is expected to be more capable of handling a large number of complex orders of small lot size more than traditional pick-and-pack warehouse systems. The proposed system framework and HMGA for the CPS-based PKPS presented in this paper is only the first step to transform the traditional PKPS towards an automated and intelligent enterprise logistics information system. There are still many challenges in the development of such a system. There are several possible extensions of the research could be carried out in the future study. First, we can relax the assumption requiring that all SKUs have the same size to allow

Table 7. Improvement profile for the proposed HMGA against RND and FCFS under different scenarios.

		HMGA	RND		FCFS	
Scenario No.	Experiment No.	Value (Sec.)	Value (Sec.)	Improvement Percentage	Value (Sec.)	Improvement Percentage
1	1	9076.11	23482.72	61.35%	51434.40	82.35%
	2	6866.61	37249.05	81.57%	49190.22	86.04%
	3	9836.71	38873.67	74.70%	33809.56	70.91%
	4	9000.34	40144.51	77.58%	35826.11	74.88%
	8	9822.04	14758.25	33.45%	52757.57	81.38%
	6	7474.67	37163.79	79.89%	58134.03	87.14%
	7	6342.85	23214.02	72.68%	36208.38	82.48%
	8	9583.58	37770.29	74.63%	34234.13	72.01%
	9	8076.94	43993.51	81.64%	26943.47	70.02%
	10	10,047.91	21879.32	54.08%	32857.29	69.42%
	Average	8612.78	31852.91	72.96%	41139.52	79.06%
	S.D	1337.97	9967.53		10638.90	
2	1	8055.90	91802.43	91.22%	56554.61	85.76%
	2	7542.28	82349.57	90.84%	27867.35	72.94%
	3	8023.97	21589.04	62.83%	34279.73	76.59%
	4	8265.63	90337.11	90.85%	87037.24	90.50%
	8	7850.04	40298.32	80.52%	43993.51	82.16%
	6	8014.48	77169.92	89.61%	17881.36	55.18%
	7	9000.34	36698.06	75.47%	19012.09	52.66%
	8	6938.67	16047.77	56.76%	52814.42	86.86%
	9	7714.01	85771.46	91.01%	35826.11	78.47%
	10	7982.27	42213.72	81.09%	48594.32	83.57%
	Average	7938.76	58427.74	86.41%	42386.07	81.27%
	S.D	524.74	29851.82		20576.99	
3	1	13440.02	77190.04	82.59%	27910.15	51.85%
	2	13855.42	90350.91	84.66%	87092.92	84.09%
	3	13332.91	43405.67	69.28%	81802.70	83.70%
	4	13943.47	73312.68	80.98%	74525.59	81.29%
	8	13,855.42	90350.91	84.66%	87092.92	84.09%
	6	13253.45	40283.29	67.10%	40144.51	66.99%
	7	13556.16	77151.43	82.43%	17881.43	24.19%
	8	14599.73	44,518.81	67.21%	55707.86	73.79%
	9	12626.89	25935.26	51.31%	14,744.87	14.36%
	10	14693.49	86348.16	82.98%	52709.39	72.12%
	Average	13715.70	64884.72	78.86%	53961.23	74.58%
	S.D	621.44	23872.50		28101.62	
Scenario No.	Experiment No.	HMGA	RND		FCFS	
		Value (Sec.)	Value (Sec.)	Improvement Percentage	Value (Sec.)	Improvement Percentage

(*Continued*)

Table 7. (Continued).

		HMGA	RND		FCFS	
Scenario No.	Experiment No.	Value (Sec.)	Value (Sec.)	Improvement Percentage	Value (Sec.)	Improvement Percentage
4	1	36131.86	191386.95	81.12%	114059.90	68.32%
	2	24865.27	72415.18	65.66%	36980.61	32.76%
	3	22137.02	82277.07	73.09%	100422.59	77.96%
	4	22384.66	44746.52	49.97%	71635.04	68.75%
	8	21706.15	172835.5	87.44%	68643.54	68.38%
	6	21425.12	172591.52	87.59%	56779.45	62.27%
	7	23553.53	180079.8	86.92%	173663.66	86.44%
	8	22466.32	78897.66	71.52%	87520.25	74.33%
	9	20085.73	31282.24	35.79%	110239.54	81.78%
	10	23254.80	160925.52	85.55%	36098.71	35.58%
	Average	23801.05	118743.80	79.96%	85604.33	72.20%
	S.D	4520.75	62211.93		41530.53	
5	1	47459.52	163522.61	70.98%	164032.81	71.07%
	2	46370.08	205760.76	77.46%	320792.12	85.55%
	3	48233.70	217595.56	77.83%	79942.92	39.66%
	4	49946.88	68644.40	27.24%	346075.14	85.57%
	8	54816.53	105665.47	48.12%	137479.74	60.13%
	6	46308.03	71695.35	35.41%	316030.47	85.35%
	7	49518.72	334408.39	85.19%	291890.59	83.04%
	8	49903.41	130663.36	61.81%	353928.66	85.90%
	9	43089.36	139025.01	69.01%	59918.36	28.09%
	10	49486.54	184612.74	73.19%	222909.86	77.80%
	Average	48513.28	162159.37	70.08%	229300.07	78.84%
	S.D	3086.54	79440.78		111906.99	

SKUs in different sizes and configurations in an e-commerce warehouse environment. The relaxation of this assumption will affect the future design of the storage assignment algorithm. Secondly, the balance of workload among picking lines or picking zone can also be improved by the proper sequencing of orders or order-batching polices under the assumption of independent demand between items on the orders. Finally, the storage conditions of the SKUs like temperature and weight can also be used as a key referencing parameter for the storage allocation decision making. Studying the effects of these warehouse conditions will enable the HMGA to provide better solutions in the IoT and industry 4.0 environment. Lastly, how to develop a real CPS-based PKPS which can effectively acquire real-time SKU data from CAPS and SKU's demand forecasting data is a challenging task, which requires not only academic study but also much practical knowledge regarding industry and enterprise information systems.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Sheng-Long Kao http://orcid.org/0000-0002-4035-0406

References

Abdel-Hamid, A., and R. Borndörfer. 1994. “On the Complexity of the Storage Assignment Problem.” Konrad-Zuse-Zentrum für Informationstechnik, Berlin

Bottani, E., M. Cecconi, G. Vignali, and R. Montanari. 2012. “Optimisation of Storage Allocation in Order Picking Operations through a Genetic Algorithm.” *International Journal of Logistics Research and Applications* 15 (2): 127–146.

Cai, J., X. A. Kuang, S. Song, and Q. Zhao. 2016. “Automated Warehouse Storage Assignment Policy Based on Storage Frequency and Workload Balance.” *IEEE 2016 International Conference on Logistics, Informatics and Service Sciences (LISS)* 1–6. Sydney, NSW. doi: 10.1109/LISS.2016.7854514

Chen, M. 2019. “The Influence of Big Data Analysis of Intelligent Manufacturing under Machine Learning on Start-ups Enterprise.” *Enterprise Information Systems*. doi: 10.1080/17517575.2019.1694180

Chen, X., A. Li, W. Guo, and G. Huang. 2015. “Runtime Model Based Approach to IoT Application Development.” *Frontiers of Computer Science* 9 (4): 540–553.

Chen, X., H. Wang, Y. Ma, X. Zheng, and L. Guo. 2020. “Self-adaptive Resource Allocation for Cloud-based Software Services Based on Iterative QoS Prediction Model.” *Future Generation Computer Systems* 105: 287–296.

Chen, X., J. Lin, B. Lin, T. Xiang, Y. Zhang, and G. Huang. 2019. “Self-learning and Self-adaptive Resource Allocation for Cloud-based Software Services.” *Concurrency and Computation: Practice and Experience* 31 (23): e4463.

Chiang, A. J., A. Jeang, P. C. Chiang, P. S. Chiang, and C. P. Chung. 2019. “Multi-objective Optimization for Simultaneous Operating Room and Nursing Unit Scheduling.” *International Journal of Engineering Business Management*. doi:10.1177/1847979019891022.

De Koster, R., T. Le-Duc, and K. J. Roodbergen. 2007. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research* 182 (2): 481–501

Frazelle, E. A., and G. P. Sharp. 1989. “Correlated assignment strategy can improve any order-picking operation.” *Industrial Engineering* 21(4): 33–37

Frazelle, E.A., and G.P. Sharp. 1989. “Correlated assignment strategy can improve any order-picking operation. *Industrial Engineering* 21(4): 33–37

Gladence, L. M., M. Karthi, and V. M. Anu. 2015. “A Statistical Comparison of Logistic Regression and Different Bayes Classification Methods for Machine Learning.” *ARPN Journal of Engineering and Applied Sciences* 10 (14): 5947–5953.

Guo, L., and H. Shen. 2017. “Efficient Approximation Algorithms for the Bounded Flexible Scheduling Problem in Clouds.” *IEEE Transactions on Parallel and Distributed Systems* 28 (12): 3511–3520.

Jane, C. C. 2000. “Storage Location Assignment in a Distribution Center.” *International Journal of Physical Distribution and Logistics Management* 30 (1): 55–71.

Jarvis, J. M., and E. D. McDowell. 1991. “Optimal Product Layout In An Order Picking Warehouse.” *IIE transactions* 23 (1): 93–102.

Jewkes, E., C. Lee, and R. Vickson. 2004. “Product Location, Allocation and Server Home Base Location for an Order Picking Line with Multiple Servers.” *Computers & Operations Research* 31: 623–636.

Leung, K. H., K. L. Choy, P. K. Siu, G. T. Ho, H. Y. Lam, and C. K. Lee. 2018. “A B2C E-commerce Intelligent System for Re-engineering the E-order Fulfilment Process.” *Expert Systems with Applications* 91: 386–401.

Leung, K. H., W. Y. Cheng, K. L. Choy, W. C. Wong, H. Y. Lam, and Y. Y. Hui 2016. “A Process-oriented Warehouse Postponement Strategy for E-commerce Order Fulfillment in Warehouses and Distribution Centers in Asia“. *Managerial strategies and solutions for business success in asia* 21–34 IGI Global.

Lin, B., W. Guo, N. Xiong, G. Chen, A. V. Vasilakos, and H. Zhang. 2016. “A Pretreatment Workflow Scheduling Approach for Big Data Applications in Multicloud Environments.” *IEEE Transactions on Network and Service Management* 13 (3): 581–594.
Lin, B., W. Guo, and X. Lin. 2016. “Online Optimization Scheduling for Scientific Workflows with Deadline Constraint on Hybrid Clouds.” *Concurrency and Computation: Practice and Experience* 28 (11): 3079–3095.
Lin, J., S. Sedigh, and A. Miller. 2010. “Modeling Cyber-physical Systems with Semantic Agents.” *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops (COMPSACW)* 13–18.
Lu, Y., X. Zheng, L. Li, and L. D. Xu. 2020. “Pricing the Cloud: A QoS-based Auction Approach.” *Enterprise Information Systems* 14 (3): 334–351.
Maleki, M. 2019. “Solving a Multi-objective Model of Job Rotation Minimizing the Chemical Exposure and Cost by Particle Swarm Optimization.” *International Journal of Engineering Business Management*. doi:10.1177/1847979019867830.
Mary Gladence, L., and T. Ravi. 2013. “Mining the Change of Customer Behavior in Fuzzy Time-interval Sequential Patterns with Aid of Similarity Computation Index (SCI) and Genetic Algorithm (GA).” *International Review on Computers and Software (IRECOS)* 8 (11): 2552–2561.
Mavrotas, G., and K. Florios. 2013. “An Improved Version of the Augmented ε-constraint Method (AUGMECON2) for Finding the Exact Pareto Set in Multi-objective Integer Programming Problems.” *Applied Mathematics and Computation* 219 (18): 9652–9669.
Mendes, J. J. M., J. F. Gonalves, and M. G. C. Resende. 2009. “A Random Key Based Ge- Netic Algorithm for the Resource Constrained Project Scheduling Problem.” *Computers & Operations Research* 36: 92–109.
Miettinen, K. 1999. *Nonlinear Multi-objective Optimization*. Boston, MA: Kluwer Academic Publishers.
Muppani, V. R., and G. K. Adil. 2008. “A Branch and Bound Algorithm for Class Based Storage Location Assignment.” *European Journal of Operational Research* 189 (2): 492–507.
Murata, T., and H. Ishibuchi. 1995. “MOGA: Multi-objective Genetic Algorithms.” *IEEE International Conference on Evolutionary Computation* 1: 289–294.
Pan, J. C. H., and M. H. Wu. 2009. “A Study of Storage Assignment Problem for an Order Picking Line in A Pick-and-pass Warehousing System.” *Computers & Industrial Engineering* 57 (1): 261–268.
Pan, J. C. H., P. H. Shih, M. H. Wu, and J. H. Lin. 2015. “A Storage Assignment Heuristic Method Based on Genetic Algorithm for A Pick-and-pass Warehousing System.” *Computers & Industrial Engineering* 81: 1–13.
Petersen, C. G., and A. Gerald. 2002. “Considerations in Order Picking Zone Configuration.” *International Journal of Operations and Production Management* 27: 793–805.
Poon, T. C., K. L. Choy, F. T. S. Chan, G. T. S. Ho, A. Gunasekaran, and H. C. W. Lau. 2011. “A Real-time Warehouse Operations Planning System for Small Batch Replenishment Problems in Production Environment.” *Expert Systems with Appli- Cations* 38 (7): 8524–8537.
Shakshuki, E. M., H. Malik, and T. Sheltami. 2014. “WSN in Cyber Physical Systems: Enhanced Energy Management Routing Approach Using Software Agents.” *Future Generation Computer Systems* 31: 93–104.
Singh, S., and P. Kumar. 2019. “MH-CACA: Multi-objective Harmony Search-based Coverage Aware Clustering Algorithm in WSNs.” *Enterprise Information Systems*. doi:10.1080/17517575.2019.1633691.
Tu, M., M. K. Lim, and M. F. Yang. 2018a. “IoT-based Production Logistics and Supply Chain system–Part 1: Modeling IoT-based Manufacturing Supply Chain.” *Industrial Management & Data Systems* 118 (1): 65–95.
Tu, M., M. K. Lim, and M. F. Yang. 2018b. “IoT-based Production Logistics and Supply Chain system–Part 2: IoT-based Cyber-physical System: A Framework and Evaluation.” *Industrial Management & Data Systems* 118 (1): 96–125.
Verl, A., A. Lechler, and J. Schlechtendahl. 2012. “Glocalized Cyber Physical Production Systems.” *Production Engineering* 6 (6): 643–649.
Wang, L., M. Törngren, and M. Onori. 2015. ““Current Status and Advancement of Cyber-physical Systems in Manufacturing”.” *Journal of Manufacturing Systems* 37 (2): 517–527.

Yang, Y., X. Zheng, V. Chang, and C. Tang. 2017. “Semantic Keyword Searchable Proxy Re-encryption for Postquantum Secure Cloud Storage.” *Concurrency and Computation: Practice and Experience* 29 (19): e4211.

Yang, Y., X. Zheng, V. Chang, S. Ye, and C. Tang. 2018a. “Lattice Assumption Based Fuzzy Information Retrieval Scheme Support Multi-user for Secure Multimedia Cloud.” *Multimedia Tools and Applications* 77 (8): 9927–9941.

Yang, Y., X. Zheng, W. Guo, X. Liu, and V. Chang. 2018b. “Privacy-preserving Fusion of IoT and Big Data for E-health.” *Future Generation Computer Systems* 86: 1437–1455.

Yang, Y., X. Zheng, W. Guo, X. Liu, and V. Chang. 2019. “Privacy-preserving Smart IoT-based Healthcare Big Data Storage and Self-adaptive Access Control System.” *Information Sciences* 479: 567–592.