Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Computers in Industry 56 (2005) 169–178

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

Batching orders in warehouses by minimizing travel distance with genetic algorithms

Chih-Ming Hsu[a], Kai-Ying Chen[b], Mu-Chen Chen[c,*]

[a]Department of Business Administration, Minghsin University of Science and Technology, Hsinchu, Taiwan, ROC
[b]Department of Industrial Engineering and Management, National Taipei University of Technology, No. 1, Section 3, Chung-Hsiao E. Road, Taipei 106, Taiwan, ROC
[c]Department of Business Management, Institute of Commerce Automation and Management, National Taipei University of Technology, No. 1, Section 3, Chung-Hsiao E. Road, Taipei 106, Taiwan, ROC

Received 3 March 2003; received in revised form 3 November 2003; accepted 28 June 2004
Available online 16 December 2004

Abstract

The power of warehousing system to rapidly respond to customer demands participates an important function in the success of supply chain. Before picking the customer orders, effectively consolidating orders into batches can significantly speed the product movement within a warehouse. There is considerable product movement within a warehouse; the warehousing costs can be reduced by even a small percentage of reduction in the picking distance. The order batching problem is recognized to be NP-hard, and it is extremely difficult to obtain optimal solutions for large-scale problems within a tolerable computation time. Previous studies have mainly focused on the order batching problems in warehouses with a single-aisle and two-dimension layout. This study develops an order batching approach based on genetic algorithms (GAs) to deal with order batching problems with any kind of batch structure and any kind of warehouse layout. Unlike to previous batching methods, the proposed approach, additionally, does not require the computation of order/batch proximity and the estimation of travel distance. The proposed GA-based order batching method, namely GABM, directly minimizes the total travel distance. The potential of applying GABM for solving medium- and large-scale order batching problems is also investigated by using several examples. From the batching results, the proposed GABM approach appears to obtain quality solutions in terms of travel distance and facility utilization.

Keywords: Order batching; Warehouses; Genetic algorithms

1. Introduction

To attain the customer service objectives in the overall supply chain, warehouses serve several value-adding roles, which include transportation consolidation, product mixing, customer service, contingency

* Corresponding author. Tel.: +886 2 2771 2171x3417; fax: +886 2 2776 3964.
E-mail address: bmcchen@ntut.edu.tw (M.-C. Chen).

0166-3615/$ – see front matter
doi:10.1016/j.compind.2004.06.001

protection and smoothing [1]. In the past, warehouses mainly focused on putting raw materials, in-process products and finished goods in storage. With the advent of supply chain management, warehouses have changed their role to strategically achieving the logistics goals of shorter order cycle times, lower inventory levels, lower costs and better customer service [1]. The order processing activities in warehouses of these days are more fast-paced than in the recent past. In order to satisfy customer demands for shorter order cycle times, products may stay in warehouses for just a few days or even a few hours. The warehousing expenditures which companies sustain involve considerable dollars amounts, hurrying the movement of products in warehouses, therefore, has continued to become an essential issue for warehouse managers.

Order picking is a process by which products are retrieved from specified storage locations with respect to customer orders. Order picking is a labor-intensive task in warehousing, improving the performance of order picking generally can lead to a large amount of savings in warehousing costs [2]. The efficiency of order picking is dependable on factors such as the storage racks, warehouse layout and control mechanisms. Order batching can be taken as an important mechanism for reducing travel distances and warehousing costs [3]. The overall logistics service level also can be improved through efficient warehousing operations. In an order picking operation, order pickers may pick one order at the time (single order picking). Batch picking (i.e., picking a number of orders simultaneously) is a better picking scheme due to that it can attain a higher productivity in a warehousing system [4].

Warehouse managers are interested in finding the most economical way of picking orders, which minimizes the costs involved in terms of travel distance or travel time. A batch is a group of orders that is simultaneously picked in a single tour. In the case of batch picking, orders are generally grouped into batches in an optimum manner under the criteria of minimum travel distance or minimum travel time. The order batching problem of minimizing the total travel distance can be generally formulated as follows [5]:

$$\text{minimize} \quad \sum_{k=1}^{\text{NO\_batch}} D_k. \tag{1}$$

Subject to:

$$\sum_{O_i \in \text{Batch}_k} \sum_{j=1}^{\text{NO\_location}} v_{ij} \leq \text{CAP}_{\text{PF}} \quad \forall k; \tag{2}$$

$$\sum_{j=1}^{\text{NO\_location}} v_{ij} \leq \text{CAP}_{\text{PF}}, \quad i = 1, 2, \cdots, \text{NO\_order}; \tag{3}$$

$$\cup_{k=1}^{\text{NO\_batch}} \text{Batch}_k = S; \tag{4}$$

$$\cap_{k=1}^{\text{NO\_batch}} \text{Batch}_k = \varnothing; \tag{5}$$

where Batch_k is the batch (tour) k; CAP_{PF} the capacity of the order picking facility; D_k the distance traveled in batch (tour) k, $D_k \geq 0$; NO_batch the number of batches formed; NO_location the number of locations (items) in the warehouse; NO_order the number of orders to be picked; $S = \{O_1, O_2, \ldots, O_M\}$, the set of orders to be picked; v_{ij} the volume of item j to be picked to fulfill order i, $v_{ij} \geq 0$.

In the above formulation, Constraint (2) limits the volume of items of all orders in one batch. Constraint (3) states that the total volume of any order cannot exceed the capacity of order picking machine. Constraint (4) necessitates that all orders must be picked, and Constraint (5) prohibits dividing any order into two or more batches.

It is extremely difficult to obtain an exact solution for the above mathematical model since the total travel distance $\sum D_k$ in the objective function depends on the configuration of formed batches and the layout of warehouse. Only a very limited amount of research applied the optimization technique to the batching procedure. Additionally, the batching optimization methods based on integer programming are restricted to small-scaled problems, which only have a small set of orders [6,7].

Researchers have developed several order batching heuristics since it is very difficult and maybe impractical to obtain exact solutions with reasonable computation efforts. A number of batching heuristics [5,8–12] have been introduced in the literature. The paper by van den Berg [4] has made a survey of these batching heuristics. Instead of directly minimizing the distance traveled by operators and/or S/R (storage/retrieval) machines, previous studies considered various order proximity and distance approximation measures to cluster orders. In a comparative study of batching heuristics, Pan and Liu [13] recommended a heuristic of Hwang et al. [11], which partitions the rack into clusters of storage locations and measures proximity between orders by the overlap in clusters. Similar to the optimization based methods, most batching heuristics developed previously also concentrated on resolving problems with a small amount of orders.

Furthermore, a relatively simple environment of warehouse with a single-aisle layout was tackled in previous studies such as Refs. [5,8,9,11–14]. Gibson and Sharp [10] has developed a batching method for a more realistic warehouse environment with a parallel-aisle layout and a large set of orders. To simplify the large-scale order batching problem, Gibson and Sharp, however, considered a distance approximation measure being the sum of distances between each item of seed order and the closest item in the candidate order. In addition, Gibson and Sharp showed that their approximation approach outperformed the method of space filling curves given in [14].

Most previous heuristics first pick a seed order for a batch and afterward expand the batch with orders that have a close relationship with the seed order. Such a batch is expanded until the limit of capacity of storage/retrieval (S/R) machine is reached. The order batching problem is complicated because the computation of total travel distance relies on the structure of batches and the layout of warehouse. The essential issues of these batching methods are defining measures for the proximity of orders/batches and approximating travel distance or travel time. Therefore, previous studies have mainly focused on the order batching problems in the relatively simple warehouse with a single-aisle and 2D layout. In such warehouse architecture, product items are retrieved from the known storage locations only with horizontal travel; therefore vertical movement of picking facility may be disregarded. In the advanced warehousing systems (e.g., automated storage/retrieval system, namely AS/RS), a 3D layout is frequently adopted to increase the cubic utilization of storage space. The calculation of travel distance is made more complicated to simultaneously take horizontal and vertical movements into consideration. Additionally, the proximity measures and approximated distance metrics are not suited for the case of 3D warehouse environment.

Instead of using proximity measures and approximating travel distances, this paper attempts to develop an optimization approach for order batching by directly minimizing the total travel distance as expressed in Eq. (1). Genetic algorithms (GAs) have been successfully applied to a wide array of difficult real-world problems [15–17]. GAs do not have many mathematical requirements for optimization problems and can deal with any kind of objective functions and constraints defined in discrete, continuous, or mixed search spaces [15]. Due to the great flexibility of GAs, additionally, they enable the efficient implementation of a specific solution by hybridizing domain-dependent heuristics. Therefore, it is beneficial to develop an order batching approach based on GAs, which directly minimizes the total travel distance. Furthermore, the GA-based batching method, namely GABM, can resolve problems with any kind of batch structure and any kind of warehouse layout. The proposed GABM approach does not require the computation of order/batch proximity and the estimation of travel distance.

This paper provides a near-optimal solution based on GAs for order batching in the warehousing systems. The proposed GABM approach automatically groups a sizeable set of customer orders into batches in an optimal way. The rest of this paper is organized as follows. Section 2 presents the proposed GABM approach for order batching, which directly minimizes the total travel distance. Section 3 reports the computational results. Finally, Section 4 concludes this study.

2. The proposed order batching approach

2.1. Basics of genetic algorithms

Genetic algorithms (GAs) were first introduced by Holland [18], who was inspired by the notion of natural and biological evolution. In GAs, the concept that mimics from population genetics and evolution theory is used to construct the optimization algorithms. They attempt to optimize the fitness of a population of elements through recombining and mutating their genes. To apply the genetic evolutionary concept to a real-world optimization problem, two issues must be addressed: encoding the potential solutions, and defining the fitness function (objective function) to be optimized.

A solution, namely a chromosome, is encoded as a string composed of several components (genes). The initial population of chromosomes is generated according to some principles or randomly selected. The algorithm performs an evaluation to measure the quality (fitness) of the potential solutions. Optimization using GAs is achieved by (a) selecting pairs of chromosomes with probabilities proportionate to their fitness, and (b) matching them to create new offspring. In addition to matching (crossover), small mutations are induced in new offspring. The replacement of bad solutions with new solutions is based on some fixed strategies. The chromosomes evolve through successive iterations, called generations. The evaluation, optimization and replacement of solutions are repeated until the stopping criteria are satisfied [16].

The general structure of GAs can be described as follows [19].

Procedure: genetic algorithms

Step 1. Define a genetic representation of a feasible solution of the problem.
Step 2. Create an initial population $P(0) = x_1^0, \ldots, x_N^0$. Set $t = 0$.
Step 3. Compute the average fitness $\bar{f}(t) = \sum_i^N f(x_i)/N$. Assign each individual the normalized fitness value $f(x_i)/\bar{f}(t)$.
Step 4. Assign each x_i a probability $p(x_i, t)$ proportional to its normalized fitness. Using this distribution, select N individuals from $P(t)$. This gives the set of the selected parents.
Step 5. Pair all parents at random forming $N/2$ pairs. Apply crossover with a certain probability to each pair.
Step 6. Apply mutation with a certain probability to each offspring.
Step 7. Form a new population $P(t + 1)$ by using the surviving mechanism.
Step 8. Set $t = t + 1$ and return to Step 3.

There are three major advantages when applying GAs to optimization problems [15]. First, GAs do not have many mathematical requirements for the optimization problems and can handle any kind of objective functions and constraints defined in discrete, continuous, or mixed search spaces. Second, the ergodicity of evolution operators makes GAs very effective at performing global searches (in probability) and finding global optima. Third, GAs provide a great hybridizing flexibility with domain-dependent heuristics to enable the efficient implementation of a specific solution. GAs have been successfully applied to a wide array of difficult real-world problems [15–17]. Goldberg [16] compared GAs with conventional search techniques including a calculus-based method, enumeration method and random method. He found that GAs could be highly efficient and reliable in solving the combinatorial, unimodal and multimodal problems. These results indicate that GAs are robust, even in a complex solution space and concurrently show efficiency and efficacy. Detailed discussions on the foundation of GAs can be found in [15–17].

2.2. The GA-based batching method

As abovementioned, the exact solutions of order batching problems are extremely difficult to obtain since the function of total travel distance is dependable on the configuration of formed batches and the layout of warehouse. By taking the advantages of GAs, we develop a GA-based optimization approach to deal with the order batching problem with any kind of batch structure and any kind of warehouse layout. In addition, a domain-dependent heuristic is incorporated into the GA procedure to efficiently generate solutions meeting the capacity limitation.

To apply GAs to resolving the order batching problems, the encoding of a solution, the fitness function and evolutionary mechanisms are defined as follows.

2.2.1. Encoding of solution

In an order batching problem, the feasible solution is encoded through a string composed of integers, which group each order into a specific batch. For example, the string of integers (1,2,3,2,1,3) groups the orders 1 and 5 into batch 1, the orders 2 and 4 into batch 2, and the orders 3 and 6 into batch 3 in a order batching problem with six orders.

2.2.2. Fitness function

The total distance traveled of an order picking mechanism can be determined by using the information about facility layout in a warehouse. To minimize the operating cost of a warehousing system, it is desired to decrease the total distance traveled. Hence, the fitness function that we want to maximize can be defined as

$$\text{Fitness}_i = \text{Distance}_L - \text{Distance}_i$$

where Distance_L is the longest travel distance among the current feasible solutions; Distance_i and Fitness_i are the travel distance and fitness value of a feasible solution i.

2.2.3. Crossover mechanism

A two-point crossover method is utilized in the proposed approach. After exchanging genes in crossover, the obtained offspring strings may be infeasible due to the violation of capacity restriction. Therefore, a correction mechanism is suggested to adjust the infeasible solutions to meet the capacity restriction. By randomly selecting two crossover points, the paired chromosomes (feasible solutions) mutually exchange information and structures of genes. For example, randomly generating two crossover points ($CP_1 = 4$ and $CP_2 = 11$), the paired solutions exchange components as follows:

Original paired solutions:	Solution #1: (**1,1,4,1**; 5,4,3,4,1,1,3,2,3,5; **5,4,3,5,3,2**)
	Solution #2: (*4,1,4,3*; 4,1,3,3,2,3,1,3,3,3; *5,5,3,3,1,5*)
	CP_1 CP_2
Matched paired solutions:	Solution #1: (*4,1,4,3*; 5,4,3,4,1,1,3,2,3,5; *5,5,3,3,1,5*)
	Solution #2: (**1,1,4,1**; 4,1,3,3,2,3,1,3,3,3; **5,4,3,5,3,2**)
	CP_1 CP_2

To ensure that the volume of each batch in each feasible solution can meet the limited capacity of picking facility, some corrections for the matched paired solutions are required. Before presenting the correction mechanism, some notations are firstly defined as follows:

CAP_{PF}: the capacity of picking facility;
O_i: order i, for $i = 1, 2, \ldots, \text{NO\_order}$;
Batch(O_i): the batch that order O_i is grouped into;
Vol(j): the volume of the batch j, $j = 1, 2, \ldots, \text{NO\_batch}$;
$\text{Batch\_Dist}^{\text{Old}}(\text{Batch}(O_i))$: the distance that the original batch (Batch(O_i)) travels, for $i = 1, 2, \ldots, \text{NO\_order}$;
$\text{Batch\_Dist}^{\text{New}}(\text{Batch}(O_i))$: the distance that the new batch (Batch(O_i)) (after removing the order O_i) travels, for $i = 1, 2, \ldots, \text{NO\_order}$;
$\text{Batch\_Dist}^{\text{Old}}(j)$: the distance that the original batch j travels, for $j = 1, 2, \ldots, \text{NO\_batch}$;
$\text{Batch\_Dist}^{\text{New}}(\text{Batch}(j, O_i))$: the distance that the new batch j (after adding the order O_i) travels, for $i = 1, 2, \ldots, \text{NO\_order}$ and $i = 1, 2, \ldots, \text{NO\_batch}$.

The correction mechanism is designed to move orders from the over capacitated batches to other batches with surplus capacity. Provided that several such batches exist, the criterion of maximum improvement in distance traveled is brought into play. The correction mechanism can be algorithmically stated as follows.

```
Correction mechanism
For each order O_i
{
   If (Vol(Batch(O_i)) > Capa_PF)
   {
      Dist_Imp(k) = Batch_Dist^Old(Batch(O_i)) +
         Batch_Dist^Old(k)
            −Batch_Dist^New(Batch(O_i)) −
               Batch_Dist^New(k, O_i),
            for k = 1, 2, ..., NO_batch;
      Dist_Imp(k*) = Max(Dist_Imp(k));
      Batch(O_i) = k*;
   }
}
```

2.2.4. Mutation mechanism

The mutation mechanism allows each component in a solution exchanging its information with another randomly selected component with a very small probability, i.e. the mutation rate. The mutation mechanism can be illustrated as follows:

Original solution: (1,1,4,**1**,5,4,3,4,1,1,3,2,3,*5*,5,4,3,5,3,2)
Mutated solution: (1,1,4,*5*,5,4,3,4,1,1,3,2,3,**1**,5,4,3,5,3,2)

Notably, the paired components can only exchange their information while each batch can still meet the capacity limitation of picking facility.

2.2.5. Selection mechanism

The selection mechanism forms a matching pool by selecting a certain number of solutions from the current feasible solutions. The probability with which a feasible solution i is selected into this matching pool is proportional to its fitness value Fitness_i, that is, the *roulette wheel selection* is adopted herein.

2.2.6. Surviving mechanism

In the present approach, the probability with which a feasible solution can survive in the next generation (cycle) is determined by

$$\text{Surv\_Pro}_i = (1 - \text{Pro\_Base})^{\text{Rank}_i} \quad (6)$$

where Surv_Pro_i is the surviving probability of solution i; Pro_Base the small probability, e.g. 0.05; Rank_i the rank of solution i by ranking the fitness values in a descending order.

3. Performance study

To demonstrate the effectiveness of the proposed GA-based approach to medium- and large-sized order batching problems, we present computational results with 10 test examples, referred to as Problems 1–10, generated in a random manner. Problems 1–6 and Problems 7–10 are test examples for 2D and 3D warehouse layouts, respectively. The basic description of these problems is summarized in Table 1. This table includes the following information of test problems: number of customer orders (NO_order), number of products items or locations in the warehouse (NO_location), total volume of items of all orders ($\text{VOL}_{\text{total}}$), capacity of picking facility (CAP_{PF}) and minimum possible number of batches ($\text{NO\_batch}_{\min}$). The minimum possible number of batches is calculated by $\text{NO\_batch}_{\min} = \lfloor \text{VOL}_{\text{total}}/\text{CAP}_{\text{PF}} \rfloor$. The proposed GABM approach is coded in Visual C++ 6.0, and it is run on an IBM compatible PC with a Pentium IV processor.

The schematic layout of warehouse considered in this experimental study is illustrated in Fig. 1. The warehouse is rectangular and consists of a number of parallel pick aisles. A depot coordinates the flow of order picking in the warehouse. A picking tour consists of a picker and/or picking machine leaving the depot, making a tour with the S-shape strategy [20] through a storage zone, and returning to the depot. For the aisle structure shown in Fig. 1, an order picker enters every aisle where an item has to be picked and travels the entire aisle. An exception is made for the last aisle provided that the number of aisles traveled in a tour is odd.

To conduct the experimentations, the following assumptions are made [5,20]:

(1) All order data are acknowledged beforehand.
(2) Splitting any order among two or more batches is not permitted; therefore the maximum order volume is lesser than the capacity of S/R facility.
(3) The location of depot is located at the left corner in the warehouse zone.
(4) Items are retrieved from the known storage locations within a warehouse with horizontal travel; therefore vertical movement of picking facility may be disregarded. This assumption will be relaxed in the problems of 3D layout.
(5) At the same time, pickers retrieve product items from the right and left sides within an isle.
(6) Picking facilities are able to traverse an aisle in both directions.

Notice that the proposed GA-based batching method can be applied to any kind of warehouse layouts. The above assumptions are made for the basis of experimentations, and most of them can be relaxed due to the mathematical flexibility of GAs.

Most previous studies have studied the order batching problems in the cases of single-aisle

Table 1
Summary of test examples

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
No. of orders	40	80	100	200	250	300	100	200	250	300
No. of items	80	160	200	300	400	200	200	300	400	200
Total volume	970.3	1550.5	1928.2	7231.6	14638.3	3409	1928.2	7231.6	14638.3	3409
Capacity	100	100	100	200	500	100	100	200	500	100
Minimum no. of batches	10	16	20	37	30	35	20	37	30	35
Layout	2D	2D	2D	2D	2D	2D	3D	3D	3D	3D

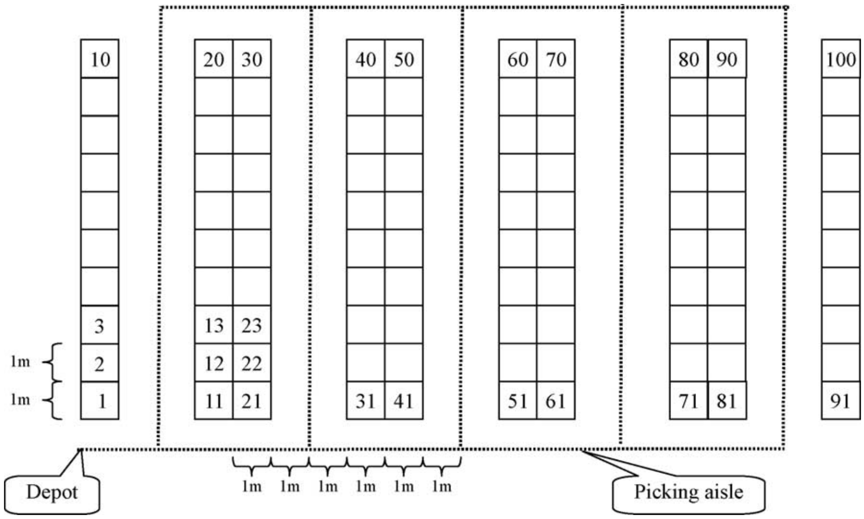

Fig. 1. The schematic layout of warehouse.

situations. Gibson and Sharp [10], however, have developed an order batching method for warehouses with a similar layout as sketched in Fig. 1. Their approach is based on the approximated distance metric. They assumed that each item's aisle location in a warehouse is known but did not consider an item's specific location within an aisle. Due to the use of approximated distance metric, Gibson and Sharp [10] simplified the calculation of travel distance. Their proposed batching approach could speedily generate heuristic solutions for the relatively large batching problems. However, the approximated distance may deviate far from the actual distance traveled. Due to that the order picking is a routine task in warehousing, even a small amount of improvement can result in significant savings for companies. The first-come first-served (FCFS) batching heuristic is straightforward [10]. However, this batching method is frequently implemented in warehouses due to its simplicity. For the FCFS batching heuristic, the first n orders from the input order list are clustered so that the batch size is as close to the capacity of picking facility as possible. Then, the next m orders are clustered. The batching process is performed until all the orders are batched.

In this paper, our GA-based batching method (GABM) is, therefore, compared to the Gibson and Sharp's batching method (GSBM) and the FCFS method. The FCFS method can be described as a baseline for comparison with the proposed GA-based order batching approach. Table 2 lists the following results of FCFS: number of batches formed (NO_batch), total travel distance (Distance), average distance per batch (D_{average}), standard deviation of distance (D_{stdev}), average utilization (%) of S/R machine per batch (Utilization) and standard deviation of utilization of S/R machine (U_{stdev}). The order batching scheme primarily aims to minimize the travel distance. As a result, the travel time and picking cost

Table 2
Summary of batching results for 2D problems by using FCFS

	P1	P2	P3	P4	P5	P6
NO_batch	12	18	22	40	32	37
Distance	912.0	2737.0	4230.0	11032	12480.0	7120.0
D_{average}	76.0	152.1	192.3	275.8	390.0	192.4
D_{stdev}	0	10.4	5.9	19.3	10.7	9.3
Utilization (%)	80.9	86.1	87.6	87.9	91.5	92.1
U_{stdev}	19.6	17.0	10.4	12.1	10.8	11.6

Table 3
The GA-specific parameters

Population size	20
Maximum generation	500
Crossover rate	0.6
Mutation rate	0.05
Consecutive generations[a]	40

[a] The maximum number of consecutive generations in which the best fitness function cannot be further improved.

Table 4
Summary of batching results for 2D problems by using GABM

	P1	P2	P3	P4	P5	P6
NO_batch	[10, 10]	[16, 16]	[20, 20]	[37, 37]	[30, 30]	[35, 35]
Distance	[711, 719]	[2200, 2200]	[3244, 3284]	[8336, 8421]	[9805, 9905]	[4911, 4935]
$D_{average}$	[71.1, 71.9]	[137.5, 137.5]	[162.2, 164.2]	[225.3, 227.6]	[326.8, 330.2]	[136.4, 140.7]
D_{stdev}	[7.5, 8.9]	[9.3, 9.3]	[14.2, 21.4]	[18.2, 25.9]	[26.3, 34.0]	[17.3, 22.7]
Utilization	[97.0, 97.0]	[96.9, 96.9]	[96.4, 96.4]	[95.0, 95.0]	[97.6, 97.6]	[94.8, 97.5]
U_{stdev}	[2.9, 4.5]	[2.6, 2.6]	[2.5, 4.5]	[3.9, 8.0]	[2.0, 3.2]	[2.0, 5.9]
CPU time (s)	19.7	170.9	2458.1	1762.4	1502.3	1031.9

*The values in [a, b] respectively indicate the minimum and maximum of each performance measure in 10 runs.

Table 5
Summary of batching results for 2D problems by using GSBM

	P1	P2	P3	P4	P5	P6
NO_batch	[11, 12]	[16, 18]	[20, 21]	[38, 39]	[31, 31]	[35, 36]
Distance	[793, 907]	[2507, 2657]	[3582, 3814]	[9169, 9571]	[10564, 10933]	[5400, 5683]
$D_{average}$	[72.1, 76.0]	[147.4, 151.1]	[175.2, 181.6]	[241.3, 251.8]	[344.2, 352.7]	[150.0, 157.9]
D_{stdev}	[1.4, 7.3]	[7.4, 15.8]	[14.5, 21.4]	[21.8, 28.6]	[23.5, 37.7]	[20.2, 23.8]
Utilization	[80.9, 88.2]	[86.1, 91.2]	[91.8, 96.4]	[90.2, 92.5]	[94.4, 97.6]	[94.8, 94.8]
U_{stdev}	[16.2, 23.8]	[15.7, 23.3]	[6.4, 20.6]	[10.7, 17.6]	[1.6, 16.2]	[13.1, 14.6]
CPU time (s)	0.5	1.1	1.8	2.3	2.6	2.8

can be reduced in warehousing. The lesser number of batches may incur less shifting cost of batch. The number of required order pickers and/or picking facilities may equal to the number of batches provided that the set of customer orders being batched is processed simultaneously. Additionally, the workload balance of each batch in picking operations is an essential concern, and it can be evaluated in terms of standard deviation of distance and standard deviation of utilization of S/R machine.

The initial population for the GA-based batching method (GABM) is generated in a random manner. Additionally, the GA-specific parameters are listed in Table 3. The seed order of GSBM is also picked in a random manner. Due to the stochastic nature of GABM and GSBM, 10 runs are performed for these five test examples. Tables 4 and 5 summarize the batching results of GABM and GSBM, respectively. The results listed in these two tables present the minimum and maximum values for each performance measure in the 10 experimental runs.

From Tables 4 and 5, GSBM and GABM lead to a considerable improvement compared to the straightforward FCFS strategy. Table 6 summarizes the distance improvement of GABM to GSBM and FCFS. From Tables 4–6, GABM achieves much improvement against GSBM. The proposed GABM not only forms less number of batches, but also reduces the travel distance. Although GABM acquires the less number of batches increases the batch size, it does not increase the travel distance. It is due to that the number of locations visited is extensively reduced with a better

Table 6
Comparisons of distances of GABM, GSBM and FCFS for 2D problems

	P1	P2	P3	P4	P5	P6
d_{GABM}/d_{FCFS}	0.79	0.81	0.77	0.76	0.79	0.69
d_{GSBM}/d_{FCFS}	0.93	0.94	0.88	0.85	0.87	0.79
d_{GABM}/d_{GSBM}	0.85	0.86	0.87	0.89	0.91	0.88

Table 7
Summary of batching results for 3D problems by using FCFS

	P7	P8	P9	P10
NO_batch	22	40	32	37
Distance	3488.0	7894.0	10668.0	5784.0
$D_{average}$	158.5	197.4	333.4	156.3
D_{stdev}	14.4	27.3	31.7	20.0
Utilization	87.6	87.9	91.5	92.1
U_{stdev}	10.4	12.1	10.8	11.6

Table 8
Summary of batching results for 3D problems by using GABM

	P7	P8	P9	P10
NO_batch	[21, 21]	[38, 39]	[31, 31]	[36, 36]
Distance	[2563, 2602]	[4889, 5023]	[6050, 6238]	[2705, 2807]
D_{average}	[122.0, 123.9]	[126.9, 131.8]	[195.2, 201.2]	[75.1, 78.0]
D_{stdev}	[17.7, 21.5]	[22.6, 28.3]	[34.9, 44.6]	[13.8, 20.5]
Utilization	[91.8, 91.8]	[90.2, 92.5]	[94.4, 94.4]	[94.7, 94.7]
U_{stdev}	[6.8, 14.5]	[6.1, 13.7]	[5.5, 8.7]	[4.2, 6.4]
CPU time (s)	4601.4	19957.0	14700.2	16235.9

batch picking program. The higher utilization of picking facility may decrease the number of batches formed without increasing the travel distance. For the workload balance, GABM provides better results in terms of standard deviation of travel distance and standard deviation of utilization. The lower standard deviations of travel distance and S/R utilization indicate the better workload balance.

Problems 7–10 are test examples for 3D warehouse layouts. In these four test problems, the information of product item and customer order is duplicated from Problems 3–6. Apart from the height of five racks, the layout of 3D warehouse considered in this experimental study is similar to the one as shown in Fig. 1. As mentioned, GSBM was developed for order batching problems with the parallel-aisle and 2D warehouse environment. It is not suited for dealing with batching problems with 3D layouts. Therefore, Problems 7–10 are only resolved by the proposed GABM approach. Their computational results are compared with those obtained by FCFS. Tables 7 and 8 summarize the results of FCFS and GABM, respectively. Table 9 compares the total travel distances between FCFS and GABM. From these tables, GABM significantly outperforms FCFS in order batching problems of 3D layout.

The major objective of order batching is to minimize total travel distance. Observing the comparisons in Tables 6 and 9, it is worthy for GABM to require more computational efforts. In this experimental study, the longest CPU times of GABM in 2D and 3D problems are 0.7 and 5.5 h, respectively. In practice, the warehouse operates on a batch principle. The planning of the day's work (e.g., the orders that are to be processed are grouped into batches) usually occurs the day before. Therefore, the proposed GABM approach can be potentially implemented in the real-world warehousing operations. Definitely, the computation time can be reduced if a more powerful computer is used.

Table 9
Comparisons of distances of GABM and FCFS for 3D problems

	P7	P8	P9	P10
$d_{\text{GABM}}/d_{\text{FCFS}}$	0.74	0.63	0.58	0.48

4. Conclusions

In order to improve the customer service level, products may wait in warehouses for just a short time. Therefore, hurrying the movement of products in warehouses has continued to become an essential issue for warehouse administrators. They are interested in discovering the most economical way of picking customer orders which minimizes the costs contained in terms of distance traveled and/or time spent. Previous studies have mainly focused on the order batching problems in the relatively simple warehouse with a single-aisle and two-dimension layout. In the advanced warehousing systems, a three-dimension layout is frequently adopted to increase the cubic utilization of storage space. In this paper, we propose an GA-based order batching method (GABM) for atomatically grouping customer orders into batches. It is extremely difficult to obtain an optimal solution effectively for the mathematical model of order batching since the total travel distance depends on the configuration of formed batches and the layout of warehouse. Instead of using proximity measures and approximating travel distances, this paper develops an optimization approach for order batching by directly minimizing the total travel distance. Furthermore, the proposed GABM approach can resolve problems with any kind of batch structure and any kind of warehouse layout. The potentials of our GABM approach for solving the medium- and large-sized order batching problems is demonstrated by the present computational results with several examples. The results encourage the development of an effective optimization method based on genetic algorithms to resolving the real-world order batching problems.

References

[1] J.J. Coyle, E.J. Bardi, C.J. Langley Jr., The Management of Business Logistics, 6th ed. West Publishing Company, New York, 1996.

[2] J.A. Tompkins, J.A. White, Y.A. Bozer, E.H. Frazelle, J.M.A. Tanchoco, J. Trevino, Facilities Planning, 2nd ed. Wiley, New York, 1996.

[3] K.J. Roodbergen, R. de Koster, Routing methods for warehouses with multiple cross aisles, International Journal of Production Research 39 (2001) 1865–1883.

[4] J.P. van den Berg, A literature survey on planning and control of warehousing systems, IIE Transactions 31 (1999) 751–762.

[5] E.A. Elsayed, Algorithms for optimal material handling in automatic warehousing systems, International Journal of Production Research 19 (1981) 525–535.

[6] R.D. Armstrong, W.D. Cook, A.L. Saipe, Optimal batching in a semi-automated order picking system, Journal of the Operational Research Society 30 (1979) 711–720.

[7] A. Kusiak, A. Vanelli, K.R. Kumar, Clustering analysis: models and algorithms, Control and Cybernetics 15 (1986) 139–153.

[8] E.A. Elsayed, R.G. Stern, Computerized algorithm for ordering in automated warehousing systems, International Journal of Production Research 21 (1983) 579–586.

[9] E.A. Elsayed, I.O. Unal, Order batching algorithms and travel time estimation for automated storage/retrieval systems, International Journal of Production Research 27 (1989) 1097–1114.

[10] D.R. Gibson, G.P. Sharp, Order batching procedures, European Journal of Operational Research 58 (1992) 57–67.

[11] H. Hwang, W. Baek, M.-K. Lee, Clustering algorithms for order picking in an automated storage and retrieval system, International Journal of Production Research 26 (1988) 189–201.

[12] H. Hwang, M.-K. Lee, Order batching algorithms for a man-on-board automated storage and retrieval system, Engineering Cost and Production Economics 13 (1988) 285–294.

[13] C.-H. Pan, S.-Y. Liu, A comparative study of order batching algorithms, Omega, International Journal of Management Science 23 (1995) 691–700.

[14] J.J. Bartholdi, L.K. Platzman, Heuristics based on space filling curves for combinatorial problems in Euclidean space, Management Science 34 (1988) 291–305.

[15] M. Gen, R. Cheng, Genetic Algorithms and Engineering Optimization, Wiley, New York, 2000.

[16] D.E. Goldberg, Genetic algorithm in search, in: Optimization and Machine Learning, Addison-Wesley, New York, 1989.

[17] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms, Springer, London, 1999.

[18] J.H. Holland, Adaptation in Nature and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[19] H. Mühlenbein, Genetic algorithms, in: E. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley, New York, 1997, pp. 137–171.

[20] R.W. Hall, Distance approximation for routing manual pickers in a warehouse, IIE Transactions 25 (1993) 76–87.

Chih-Ming Hsu is currently an Associate Professor in the Department of Business Administration at Ming Hsin University of Science and Technology, Taiwan. He holds a Ph.D. in Industrial Engineering and Management from National Chiao Tung University, Taiwan. His present research interests include quality engineering, optimization methods in industrial applications and data mining applications in CRM.

Kai-Ying Chen received the B.S., M.Sc. and Ph.D. degrees all in Mechanical Engineering from National Taiwan University in 1988, 1990 and 1996, respectively. After graduating, he served in Industrial Technology Research Institute as an engineer for 4 years. Dr. Chen is currently an Assistant Professor with the Department of Industrial Engineering at National Taipei University of Technology, Taipei, Taiwan. His research interests are in the fields of shop floor control, management information system and Petri-Net modeling.

Mu-Chen Chen received his Ph.D. and M.Sc. degrees both in Industrial Engineering and Management from National Chiao Tung University, and his B.S. degree in Industrial Engineering from Chung Yuan Christian University. He is currently a Professor in the Department of Business Management, and Institute of Commerce Automation and Management at National Taipei University of Technology, Taiwan. His teaching and research interests include meta-heuristics, simulation analysis and logistics management. He is also involved with research and industry in a range on NSC (National Science Council, Taiwan) and enterprise projects.