

# Finding Interactions or Relationships Between Customer Orders for Building Better Batches by Means of an Estimation of Distribution Algorithm-based approach for the Online Order Batching Problem

Ricardo Pérez-Rodríguez  
ricardo.perez@cimat.mx

Arturo Hernández-Aguirre  
artha@cimat.mx

CIMAT, AC  
Callejón de Jalisco s/n, Mineral de Valenciana  
Guanajuato, Guanajuato, México  
+52 (473) 7350800

## ABSTRACT

Order-picking systems are very common in industry. In this warehouse environment an order batching consists of combining customer orders into picking orders. Order batching is a combinatorial issue because many customer orders arrive throughout the scheduling in real-world situations, therefore to find interactions or relationships between orders for building better batches is a difficult task. This paper introduces an estimation of distribution algorithm-based approach for the online order batching problem to guide the overall search process. The results show how the warehouse performance is improved by estimating relationships between orders. This approach is compared with others published in the literature.

## Keywords

Routing and layout; Time-tabling and scheduling; Metaheuristics; Heuristics; Combinatorial optimization.

## 1. INTRODUCTION

Warehouse activities are essential in supply chain management because these activities are present in all levels of the chain [1]. Among these warehouse activities, order picking is one of the most labor-intensive activities [2]. However, order picking is necessary because, although incoming articles are received and stored in large unit loads (pallets, racks, or steel baskets), customers only require a few different articles at a time, and warehouse performance can be improved through the order-picking process [3]. An efficient order-picking process can raise the service level for the customers and improve the supply chain performance [4]. Additionally, service time is the key factor for efficient performance in the order-picking process of picker-to-parts picking systems, and order-batching is fundamental for shortening the service time in these systems. Therefore, order batching is essential to improving the efficiency and overall effectiveness of the order picking process in warehouses.

Current research has also focused on this topic. [5] writes about the order picking process, and [6] formulate a study on the overall

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

GECCO '16, July 20-24, 2016, Denver, CO, USA  
© 2016 ACM. ISBN 978-1-4503-4206-3/16/07...\$15.00  
DOI: <http://dx.doi.org/10.1145/2908812.2908944>

design of order picking. Meanwhile, [7] consider human factors in the order-picking process. The main studies focus on searching for better solutions to operation policies for the order-picking process such as layout design [8], storage assignment [9], zoning [10], routing [11] and order-batching [12]. These topics are investigated because without effective operation policies, additional material handling costs occur.

Dynamic markets, varieties of products, and customization are so complex that conventional approaches sometimes are not sophisticated enough to deal with these situations. There is a need to improve order-picking efficiency to be competitive in such conditions. Most order-picking processes are more dynamic recently because decisions are made in real time due to orders arriving stochastically throughout the schedule, and the articles that will be required are not known until the order arrives [13]. These constraints require new approaches that quickly identify relations or interactions among arriving orders. Consequently, online order-batching issues have gained importance.

The conclusions of previous studies mentioned above allow for making decisions in order to get promising solutions; however, in real-world environments it is difficult to implement these solutions due to a lack of knowledge about such interactions among orders. Furthermore, in the majority of these works, the customer orders are considered to be independent and isolated from each other, and the batches are only formed according to the available orders and pickers. Also, there is no previous research that studies how the relations or interactions among orders explicitly affect the batching performance. To narrow the research gap, we consider that there exists a relationship or interaction among orders and that this relationship can help identify better procedures for batching. Therefore, such interaction can help make better or more efficient batches. In this study, an Estimation of Distribution Algorithm (EDA) is incorporated in order to group unprocessed orders in fixed time window batching. The EDA proposal, OBEDA, Order Batching Estimation of Distribution Algorithm, works in an online order-batching environment. The objective is to assign certain orders to their respective batches according to a model of probability. If there exists a relation between or interaction among different orders, it will be specified by the probability model, which assigns orders to batches. Therefore, in this system, the distance travelled to collect the articles is shorter, reducing the service time. In order to validate the feasibility of the OBEDA, we established a workload with different customer orders and different types of articles replicating real order-picking conditions.

## 2. LITERATURE REVIEW

Warehouse layout is an important factor which greatly affects order picking process efficiency [14], and some literature exists outlining guidelines for conventional layout design. [15] develop an analytical approach for estimating the average travel distance of a picking tour in picker-to-parts systems where items are assumed to be allocated into a single-block warehouse layout structure using the ABC-storage assignment policy. In addition, [16] build a probabilistic model for estimating the average travel distance of a picking tour in a 2-block class-based storage strategy. They also create an optimization model to enhance the design of the 2-block layout structure by determining the optimal distance at which the middle cross-aisle should be placed within the warehouse. [17] construct a model to minimize the average travel distance required to complete a given pick list. This model creates a more efficient design for warehouse layout structures consisting of multiple cross aisles (i.e., multiple blocks) by optimizing the number of blocks of which a warehouse layout structure consists. [18] implement an analytical model for optimizing the number and positioning of middle cross aisles in a warehouse layout structure in order to minimize the average travel distance required to complete a given pick list. They generate the possible combinations of how a given pick list might be distributed in the warehouses based on the fully enumerated set of patterns, along with their associated probabilities. [19] proposes a simulation approach to design a single-block warehouse layout structure for two different storage assignment policies: the volume-based storage assignment policy with two distinctive cases, within-aisle storage and across-aisle storage; and the random storage assignment policy, which is based on the total travel distance of an order picker. Recently, [20] describe non-traditional aisle designs in which the rules requiring parallel picking aisles and orthogonal cross aisles are relaxed. The authors mention developing two designs: Flying-V and Fishbone. The Flying-V design contains a cross aisle that projects diagonally in a piecewise linear fashion from the depot and offers a reduction in the expected picking travel distance of approximately 10%. The Fishbone design reduces the travel distance by approximately 20%. Furthermore, [21] propose two new aisle designs, the modified Flying-V cross aisle and the inverted-V cross aisle, to facilitate flow between multiple depots and locations in a unit-load warehouse. When compared to a traditional warehouse with a single depot, these new designs offer a reduction of less than 3% in the expected travel distance. Multiple depots have also been examined in the context of an order picking system by [22].

Operational policies including storage location assignment, picker routing, order batching and zoning also affect order picking process efficiency.

With a random storage policy, the storage location for incoming loading units is selected randomly from all available empty locations [23]. Thus, all storage locations have an equal probability of being visited in a picking tour. This also holds true if an incoming loading unit is assigned to the closest open location and retrievals are performed on a first-in-first-out (FIFO) basis [24]. In comparison to other policies, random storage utilizes space more efficiently [25]. Additionally, random storage is used in many order-picking systems because it is fairly easy and inexpensive to implement [26].

There are many studies comparing the performance of different routing strategies. However, we will not present further details because the respective results differ depending on the assumptions and the policies considered as well as their interaction with other

operational parameters. Nevertheless, the problem of finding the shortest route for a given set of locations can be transformed into a special case of the classical Traveling Salesman Problem (TSP). Diverse algorithms include single-block warehouses with decentralized depositing, two-block warehouses with a central depot, and multiple-block warehouses with a central depot. Unlike optimal routings, heuristics lead to non-optimal routes, but they offer feasible solutions by making use of a specific set of rules resulting in routes that are easy to understand [27]. In the well-known traversal, also known as transversal or s-shape, routing policy, an aisle containing at least one pick has to be traversed entirely, i.e. the order picker enters the aisle at one end and exists at the other end. Under a return routing policy the picker enters and exits the aisle from the same end. [27] introduces the composite routing policy, which combines the traversal and return policies. This strategy considers the distance between two pick locations in adjacent aisles that have the greatest distance to the lower cross aisle, and uses either a traversal or a return policy to minimize this distance.

The literature on zoning is rather limited. While some studies use queueing models to estimate throughput in pick-to-belt order picking systems, others propose heuristic methods to assign items to storage locations to achieve balanced workload across zones. In a simulation study, [19] examine different zone shapes.

[28] examine the joint order batching and order picker routing problem in conventional multi-parallel-aisle picker-to-part order picking systems. A mathematical model is formulated, and a simulated annealing algorithm is developed to batch orders and to determine pick tours. The authors offer a realistic model and improve classical batching and routing heuristics, and they pay special attention to the practical applicability of the model. The proposed methods are compared and evaluated in an extensive numerical study, and it is shown that the developed approach leads to an improved solution for the joint order batching and order picker routing problem in comparison to classical heuristics for this problem.

[29] present a simulated annealing method to reduce order picker's travel times in a picker-to-parts system, where they combine picker-routing and order-batching approaches in their research. Precedence constraints such as weight, fragility, shape, size, stacking of the articles on the tour, and preferences for unloading sequence are considered in this study. Also, in the order-batching phase, they batch orders comparing all pairs of orders and estimating the savings between them. Therefore, the simulated annealing method maximizes the total savings in travel distance, and it also estimates the best batches using inherent information in batches of one or two customer orders. This method is fast, and it compares different heuristics from literature to find optimal solutions. For the routing phase, they use an A\*-type shortest path algorithm, which is not constrained by the particular layout. The algorithm considers the precedence constraints mentioned above in the routing.

[30] analyze how local search-based metaheuristics and attribute-based hill climber algorithms used in the order-batching problem can be applied to minimize the total tardiness in an order-batching and sequencing problem with due dates. The attribute-based hill climber algorithms use a set of attributes to guide the search out of local minima in addition to employing a set of moves to explore the neighborhood of a current solution. Two moves are used to search for new solutions: shifting an order from one batch to another batch and swapping two orders between two batches. In

the results, the proposed methods provide solutions that may allow order-picking systems to operate more efficiently.

[31] develop an order-batching framework in narrow-aisle order picking systems to control picker blocking. This research offers strategies that challenge the traditional assumptions regarding wide-aisles and narrow-aisles in the order picking processes. The aforementioned framework includes a sequencing problem formulated as a mixed integer program, which is solved by a simulated annealing heuristic for large-scale problems. In addition, they present a procedure to solve both the batching and sequencing problems simultaneously. The experimental results exhibit that the batching method reduces picker blocking over a variety of storage policies showing that batching and sequencing permit rules to be designed for order picking systems.

[32] make discrete time queue models in order to calculate the throughput time distribution of a one-block warehouse system using general distribution assumptions. This research builds queue models specifically to determine the optimal batch size for both multi-line and single-line picking processes, and it is not only restricted to Poisson type arrivals. With this approach, the order-batching problem is analyzed so that it only needs a stationary arrival stream of orders to function. Also, the authors suggest that the batch size is slightly higher than one would choose if only minimizing average throughput time.

[11] propose tabu search algorithms integrated with a novel clustering algorithm to solve the order-batching and picker routing problems jointly for multiple-cross-aisle warehouse systems. They first develop a clustering algorithm that accounts for route similarity, and then this algorithm is integrated with a tabu search to generate a fast and effective initial solution to the order-batching problem.

[33] introduce a route-selecting order batching formulation. Instead of constructing routes for batches, they use the S-shape routing method to create a set of all possible routes to which batches can be assigned in a warehouse layout. With this formulation, a best fit route is able to be selected for each batch. In order to formulate these calculations, the authors assume that each order picker can pick up to ten orders in ten bins, and that an average order size is two lines. Finally, they compare this formulation with C&W(ii) and seed algorithms. In this comparison their formulation dominates the other heuristics, and C&W(ii) performs better than the seed algorithms.

Currently, some related studies begin to aim at settling problems in warehouse management dynamically. [34] develop a polling-based dynamic order picking system for online retailers. Similarly, [35] present a solution for a dynamic rescheduling problem in warehouse environments involving new orders arriving randomly while static orders had been given in advance. [36] design a Markov decision process based approach to employ an optimal decision making policy for order batching. [12] describes an online order-batching problem in a walk-and-pick warehouse that minimizes the completion times of all dynamically arriving customer orders, otherwise known as the makespan. The author also shows modifications of solution approaches for off-line order batching in order to deal with the online situation. [13] use a G/G/m queuing network approximation model to analyze the impact of batching and zoning on order picking system performance with online order arrivals. They claim that batch sizes have a large impact on mean order throughput time and that when the number of zones becomes larger, errors increase.

[37] introduces a GA for the order-batching problem that considers traversal and return routing policies. As in many studies, the solution is represented by integer numbers, and Öncan uses a uniform crossover operator to produce new offspring. In an attempt to improve the new solution, four operators are used as part of a heuristic method, where they remove orders from their batches and re-assign them to other batches. The proposed GA has been tested on randomly generated instances and compared with well-known savings algorithms. According to the author's extensive computational experiments, the proposed GA yields promising solutions in acceptable computation times.

[38] use a genetic algorithm for the order-batching problem that is not limited by the restrictions of any batch structure, single-aisle or multiple-aisle, or any warehouse layout. This proposed method does not require the estimation of travel distance as used in previous studies, which use various order proximity and distance approximation measures to evaluate cluster orders. Alternatively, the aforementioned genetic algorithm directly minimizes the total travel distance. Additionally, the authors encode the feasible solution through a string of integers, and several examples are given in order to apply the proposed algorithm for solving medium- and large-scale order batching problems in terms of travel distance and facility utilization.

EDA is a relatively new paradigm in the field of evolutionary computation. Introduced by [39] and compared with other evolutionary algorithms, the EDA reproduces new population implicitly instead of using traditional evolutionary operators. To create an EDA, a probability model of the most promising area is built by statistical information based on the search experience, and then the probability model is used for sampling to generate new individuals. The EDA makes use of the probability model to describe the distribution of the solution space while the updating process reflects the evolutionary trend of the population. The EDA has successfully been used to solve complex combinatorial optimization problems, and its application has been proven in studies such as [40-42]. However, there are currently no publications that use an EDA for order-batching.

Finally, although different algorithms have been proposed in order to solve the online order-batching problem successfully, they occasionally use queue models to tackle the problem. In this case, customer orders are considered independent and isolated from each other, which could be a disadvantage if there is a relation or interaction between customer orders as they arrive in the warehouse. The current objective of preserving any relationship or interaction using a probability model has not been considered in this perspective. To the best of our knowledge, probability models by means of a discrete EDA have not been used to tackle the order-batching problem in order-picking warehouses in deep. Although there are studies related such as [43] where the solution representation is based on real-valued domains, this new research does not need to decode the solution to be represented as a valid batch sequence as the previous work. In addition, using a solution representation based on real-valued domains, the probability distribution on a set of batches is approximated. This paper works over the discrete vector directly in order to preserve the original probability distribution.

### 3. PROBLEM STATEMENT

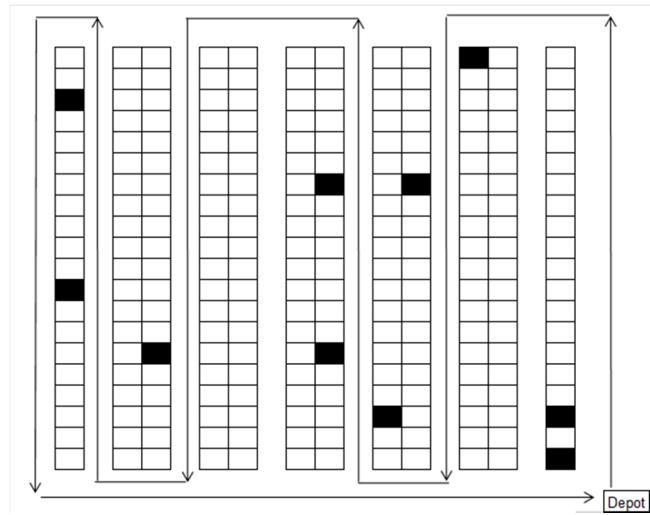
Below is an explanation of the online order-batching problem based on [12]. During the picking process, which is being analyzed in this paper, the operators (pickers) start at a depot, and the picker normally walks through the warehouse and picks up

articles from different storage locations. Afterwards, the picker returns to the depot and hands over the picked articles. The picker uses an S-Shape route to find and collect articles. There is no limitation if we consider the S-Shape route for this research because this route has been widely used in industrial environments and it is easy to implement and understand. In addition, the S-Shape route has been recognized as a benchmark in the related literature. Figure 1 demonstrates the straightforward character of the routing scheme for a set of articles to be picked. The black rectangles symbolize the corresponding locations where articles have to be picked (pick locations). We note that from a theoretical-planning point of view it would be desirable to solve the order batching and the picker routing problems simultaneously. However, practically, this appears to be unrealistic due to the complexity and the size of the problem analyzed. A decision on the S-Shape route has already been made, which will serve as input for a subsequent order batching decision.

Picking devices are generally used to complete the order picking process. Consequently, different orders can be combined until the capacity of the device is exhausted. The splitting of an order into two or more batches is prohibited, as it would result in an additional unacceptable sorting effort. It would increase the activities in the warehouse for the pickers and, consequently, the service time of the supply chain would increase. Finally, if the picker has already started a tour, it must be completed without interruption.

In online batching there is no information given about the quantity or the characteristics of the orders that will arrive. The decision about which orders should be processed directly has to be made without considering information about future incoming orders. The point in time when an order becomes available is called arrival time. The start time (release time) of a batch is the point when an order picker starts to process this batch. The start time of an order is the same as the start time of the batch to which the order is assigned. The point in time when the order picker returns to the depot after collecting all articles is called the completion time of a batch or of an order. The customer order waiting time can be determined as the length of the time between the arrival time and the start time of an order. The turnover time (response time) is the amount of time that an order stays in the system, i.e., the time period between the completion and the start time of an order. This measure can be seen as an indicator of the service level, which can be defined as the number of orders that can be processed in a given period of time. If the number of arriving orders is too large for processing each customer order separately in an appropriate amount of time, customer orders must be combined into batches. This paper focuses on situations of grouping customer orders into batches such that the average turnover time of all customer orders is minimized as well as discussing the situation where there is a single order picker and all batches must be processed one after another. Although here it is analyzed for academic purposes, a single order picker is relevant in order to consider idle time situations while customer orders are arriving in the warehouse. Furthermore, a single order picker is necessary in order to be consistent within the results detailed below.

Any algorithm for the online order-batching problem has to form and release batches without having complete information on the characteristics and arrival times of future orders.



**Figure 1. Example of an S-Shape route in a single-block warehouse.**

When a set of unprocessed orders arrives and an order picker becomes available, those unprocessed orders can be grouped into one or more batches that should either be released directly or whose start should be postponed until a later point in time, as defined previously.

## 4. OBEDA – FOR THE ONLINE BATCHING PROBLEM

In this section, we propose the OBEDA to solve the online order-batching problem. We also introduce the solution representation, population initialization, and a probability model.

### 4.1 Solution representation (chromosomes codification)

Any solution to the online batching problem mentioned should be expressed by the assignment of customer orders to batches. An order assignment vector represents a solution where the number of elements equals the total number of orders to pick up, where each element contains an integer value that represents the batch to be assigned. This representation is depicted in Figure 2 with 10 orders and 3 batches. Please note that the first batch consists of orders {5; 7; 9}, the second batch includes orders {1; 3; 4; 6}, and the third batch incorporates orders {2; 8; 10}.

Customer Orders										
Batch	1	2	3	4	5	6	7	8	9	10
Batch	2	3	2	2	1	6	2	1	3	1

**Figure 2. Solution representation.**

### 4.2 Generation of the population

Initial population members are generated randomly in order to enable a wide range of solutions [44].

### 4.3 Probability model

The probability model is designed as a probability matrix, i.e., batch probability matrix. The element  $p_{ji}$  of the probability matrix represents the probability that the batch  $j$  be used for the  $i$  customer order. For all  $j$  ( $j = 1, \dots, m$ ) and for all  $i$  ( $i = 1, \dots, n$ ). The value of  $p_{ji}$  indicates the opportunity of a customer order for a certain batch. New promising individuals may be generated through sampling according to the probability matrix.

For every position  $i$ , batch  $j$  is selected with the probability  $p_{ji}$ . If the batch  $j$  has already filled according to the device capacity, it means the assignment of the batch  $j$  has been finished. Then, the whole column  $p_{j1}, p_{j2}, \dots, p_{ji}$  of the probability matrix will be set as zero. This updating mechanism considers the previous assignments.

The major procedure of the OBEDA is listed as follows:

Step 1. Set the generation index  $g = 0$ . Initialize an initial population  $S(0)$  of size  $M$ . It means to generate discrete numbers for each customer order in each chromosome representation. Each discrete number corresponds to a batch  $j$  for each customer order.

Step 2. Select a subset  $D$  from  $S(g)$  of size  $N$ , where  $N \leq M$ . A selection method should be implemented in order to determine what vectors are candidates for detecting relations or interactions among customer orders. The fitness values of the discrete vectors are useful for the selection method.

Step 3. Establish a probabilistic model  $P$ , which describes the distribution characteristics of  $D$ . A probability matrix, i.e., batch probability matrix, should be built.

Step 4. Generate a set  $K$  of new individuals by sampling  $P$ . For each individual, generate a random number and, according to the probability matrix, select what batch should be elected for the customer order. Repeat until all customer orders have been batched.

Step 5. Select the best individuals from  $K \cup S(g)$  and assign them to the next generation  $S(g + 1)$ . The fitness values of the discrete vectors are very useful to select the best individuals.

Step 6. Let  $g = g + 1$ . If  $g < GN$ , where  $GN$  is the maximum number of generations, return to step 2. Otherwise, output the best solution in  $S(g)$ .

Figure 3 details the overall OBEDA process. Detailed procedures about discrete EDAs can be consulted in [45].

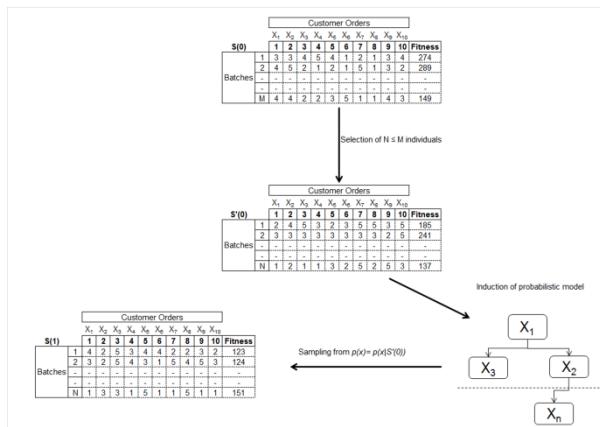


Figure 3. The OBEDA algorithm.

## 5. RESULTS AND COMPARISON

A GA is proposed as a benchmark for comparison with the OBEDA scheme. GA works with tournament selection. The ‘edge recombination operator’ is used as a cross operator based on [46], and a mutation operator changes batches among different positions. The GA for comparison uses the closeness between best individual and the average fitness of the population as stopping criterion. The GA contains 50 individuals per generation.

We established a workload to evaluate and find the best average turnover time. The workload contains different customer orders, due dates, and differing types of articles required in a workday, replicating the order-picking process conditions. Each corresponding customer order includes different numbers of articles, and the arrival times of the customer orders are indeterminable. Our experiments were based on a single-block warehouse with two cross aisles. It is assumed that there is one in the front and one in the back of the picking area. This layout was used by [2]. As the author explains, the picking area consists of 900 storage locations, where a different article has been assigned to each storage location. The storage locations are arranged into 10 aisles with 90 storage locations each. The aisles are numbered from 1 to 10; aisle 1 is the leftmost aisle, and aisle 10 the rightmost. The depot is in the left-hand corner of the warehouse. We further assume that an order picker walks 10 storage locations in 30 seconds and that they need 10 seconds to search and collect an article from a storage location. For the capacity of the picking device  $K$ , we assume two different values, namely 45 and 75 articles. For the routing strategy, the S-Shape route is used. For an order, we choose the quantity of articles uniformly distributed in  $\{5, \dots, 25\}$ , resulting in 3 or 5 orders per batch on average, in accordance with the aforementioned capacities of the picking device. For the total number of orders  $n$  we consider quantities of 30 and 60. The orders should arrive within a planning period of eight hours. The inter-arrival times, i.e., the time between the arrival of order  $i$  and order  $i+1$ , are exponentially distributed with a parameter  $\lambda$  called the arrival rate. Let  $X(t)$  be the number of incoming orders in the time interval  $[0, t]$ . In the case of exponentially distributed inter-arrival times  $E[X(t)] = \lambda \cdot t$  holds true. In our numerical experiments we choose  $\lambda$  so that the expectation  $E[X(t)]$  is equal to  $n$  for  $t = 8[h]$ . To summarize, we use the following values for  $\lambda$ : for  $n = 30$ :  $\lambda = 0.0625$ , for  $n = 60$ :  $\lambda = 0.125$ .

As a response variable for the experiment, we measure the relative percentage increase (RPI)

$$RPI(c_i) = (c_i - c^*) / c^* \times 100 \quad (1)$$

where  $c_i$  is the average turnover time obtained in the  $i$ th replication by a given algorithm configuration, and  $c^*$  is the best objective value found by any of the algorithm configurations. Note that for this problem, there are no known effective exact techniques, and comparing against an optimum solution is not possible.

For space reasons, we only show a case, i.e., Table 1 details the average obtained for each trial when the number of orders is 30 and the  $K$  capacity is 45 articles, however different cases have been considered in this research, e.g., when the total number of orders is 30 and the  $K$  capacity is 75 articles, when the total number of orders is 60 and the  $K$  capacity is 45 articles, when the total number of orders is 60 and the  $K$  capacity is 75 articles. All cases results, i.e., all trials are available for consulting. We analyze whether there is a statistically significant difference between averages of both algorithms. The method used to analyze the statistical significance is a test mean difference where  $\mu_1$  is the mean of GA and  $\mu_2$  is the mean of the proposed algorithm. All tests satisfy statistical assumptions. Therefore, non-parametric statistical analysis of the results is not required. In addition, Table 2 details the results obtained for each trial when the total number of orders is 30 and 60 and the  $K$  capacity is 45 and 75 articles. We analyze whether there is a statistically significant difference between averages of both algorithms.

**Table 1. Comparison of results for each average**

Trial	GA	OBEDA	$\mu_1 - \mu_2$	$H_0: \mu_1 - \mu_2 \leq 0$
				$\alpha = 0.01$
				<b><math>Z_c = 1.644</math></b>
1	0.0901	0.0835	0.0066	0.6724
2	0.0599	0.0158	0.0441	9.962 *
3	0.0508	0.0072	0.0436	4.543 *
4	0.0310	0.0144	0.0165	2.269 *
5	0.0908	0.0155	0.0753	8.725 *
6	0.0755	0.0365	0.0390	5.511 *
7	0.1441	0.0134	0.1308	33.11 *
8	0.0989	0.0144	0.0845	10.55 *
9	0.1355	0.0144	0.1211	19.57 *
10	0.0700	0.0000	0.0700	7.069 *
11	0.1425	0.0185	0.1239	13.38 *
12	0.0645	0.0082	0.0563	6.193 *
13	0.0000	0.0041	-0.0041	0.6113
14	0.0671	0.0155	0.0516	5.384 *
15	0.0636	0.0185	0.0451	5.648 *
16	0.0848	0.0133	0.0715	11.10 *
17	0.0865	0.0314	0.0551	5.607 *
18	0.1002	0.0122	0.0880	9.420 *
19	0.0908	0.0028	0.0879	14.62 *
20	0.0840	0.0134	0.0706	14.77 *
21	0.0808	0.0082	0.0725	7.082 *
22	0.0634	0.0122	0.0512	11.03 *
23	0.0759	0.0082	0.0676	7.408 *
24	0.1272	0.0071	0.1200	12.76 *
25	0.0276	0.0012	0.0264	4.511 *
26	0.0561	0.0122	0.0439	5.523 *
27	0.1145	0.0061	0.1085	14.81 *
28	0.0639	0.0028	0.0611	9.856 *
29	0.0953	0.0144	0.0808	10.08 *
30	0.0590	0.0016	0.0574	6.956 *
31	0.0273	0.0124	0.0149	1.5363
32	0.1140	0.0155	0.0985	10.51 *
33	0.0678	0.0082	0.0596	7.157 *
34	0.1428	0.0144	0.1283	13.62 *
35	0.0881	0.0179	0.0702	9.154 *
36	0.0622	0.0198	0.0424	4.313 *
37	0.1285	0.0134	0.1151	12.60 *
38	0.1064	0.0386	0.0677	31.40 *
39	0.0630	0.0324	0.0306	4.671 *
40	0.0718	0.0144	0.0574	9.288 *
41	0.1134	0.0080	0.1054	10.97 *
42	0.1002	0.0082	0.0919	49.37 *
43	0.1270	0.0144	0.1126	51.82 *
44	0.0787	0.0016	0.0770	14.82 *
45	0.0574	0.0144	0.0430	7.523 *
46	0.0695	0.0208	0.0487	5.549 *
47	0.0575	0.0454	0.0120	1.4842
48	0.0575	0.0386	0.0188	5.930 *
49	0.0575	0.0208	0.0367	4.439 *
50	0.0575	0.0082	0.0492	7.611 *
$\mu$ the average				46/50

**Table 2. Comparison of results for all trials.**

Orders	Capacity	GA	OBEDA	GA	OBEDA
n	K	$\mu_1$	$\mu_2$	$\sigma^2$	$\sigma^2$
30	45	0.0276	<b>0.0012</b>	0.0018	<b>0.0010</b>
30	75	0.0118	<b>0.0056</b>	<b>0.0016</b>	0.0017
60	45	0.0558	<b>0.0061</b>	0.0037	<b>0.0006</b>
60	75	0.0517	<b>0.0086</b>	0.0040	<b>0.0011</b>

$\mu$  the average

$\sigma^2$  the variance

As we can see in Table 1, there is a statistically significant difference between the averages of both algorithms. The performance of OBEDA was superior in 46 of the 50 trials with a significance level of  $\alpha=0.10$ . According to the Table 2, there is a

statistically significant difference between the averages of both algorithms. The performance of OBEDA was superior in all trials with a significance level of  $\alpha = 0.10$ .

## 6. CONCLUSIONS

This paper deals with the online order-batching problem, where batches have to be constructed and released in order to minimize the average turnover time for a given set of known and available customer orders. The problem of online order batching is pivotal for the efficient management and control of picker-to-parts order picking systems. To solve this problem we proposed the application of the OBEDA. By means of numerical experiments it was shown that this approach generates solutions superior to those generated by a GA. Implementing these solutions can improve customer service by delivering orders on time and by avoiding delays in other process stages. The OBEDA can detect relationships or interactions between orders and can improve warehouse performance significantly by means of a probability model. Since the OBEDA presents stability, it appears very suitable for implementation in software systems for practical purposes. The present results include several examples that detail the potential of the OBEDA approach for solving the online order-batching problem. The results encourage the development of an effective optimization method based on a probability model to resolve the real-world order-batching problems, which generally occur in order-picking environments where time window batching is prevalent. This is a managerial implication because it is necessary to set a variable or fixed time window batching before implementing the algorithm. For this paper, a fixed time window batching was set every hour. The proposed probabilistic model uses one-hour periods, and unprocessed customer orders are the input used to build the initial population. The solution representation, i.e., the solution vector size, is another managerial implication because we need to decide if the previously batched orders can be re-batched with new unprocessed orders arriving when the picker is already busy. In this research, re-batching is not permitted, and the OBEDA only batches unprocessed orders. Furthermore, orders with priority during the service time are another managerial implication. An analysis about how they affect order-picking performance considering orders with priority may be required in order to integrate such orders into the algorithm. As an extension of the OBEDA, we conclude that an effective estimation of interactions among orders can address the order-batching problem successfully. Considering the positive results of this experiment, we plan to estimate relationships for other related planning issues of order-picking systems such as layout design, item location, and picker routing. Further research may deal with an extension of the OBEDA that uses higher probabilistic models to represent more complex interactions or relationships among variables of the order-picking performance. Effective modules for specific users in the industry are required, and learning about probabilistic models would be useful to improve the order-batching. The OBEDA could also possibly be applied to other types of picking systems including automated and parts-to-picker systems. Finally, the literature does not sufficiently consider the interaction among orders for batching with other related planning issues by means of a probability model, and it might be worthwhile to provide a different solution approach, which integrates decisions on these planning issues in order to minimize the average turnover time.

## 7. REFERENCES

- [1] Choy, K.L., Ho, G.T.S., and Lee, C.K.H. 2014. A RFID-based storage assignment system for enhancing the efficiency of order picking. *Journal of Intelligent Manufacturing*. 1-19. DOI= <http://link.springer.com/article/10.1007%2Fs10845-014-0965-9#page-1>
- [2] Henn, S. 2009. Metaheuristics for the order batching problem in manual order picking systems. *Ottovon-Guericke-University Magdeburg. Faculty of Economics and Management*. Verband der Hochschullehrer für Betriebswirtschaft e.V., 3, 1, 82-105.
- [3] de Koster, M., Le-Duc, T., and Roodbergen, K. 2007. Design and Control of Warehouse Order Picking: A Literature Review. *European Journal of Operational Research*, 182, 2, 481-501.
- [4] Chen, F., Wang, H., Xie, Y., and Chao, Qi. 2014. An ACO-based online routing method for multiple order pickers with congestion consideration in warehouse. *Journal of Intelligent Manufacturing*. 1-20. DOI= <http://link.springer.com/article/10.1007%2Fs10845-014-0871-1>
- [5] Wäscher, G. 2004. Order Picking: A survey of planning problems and methods. In *Supply Chain Management and Reverse Logistics*, H. Dyckhoff, R. Lacke, and J. Reese Eds. Springer, Berlin, 323-347.
- [6] Battini, D., Calzavara, M., Persona, A., and Sgarbossa, F. 2015. Order picking system design: the storage assignment and travel distance estimation (SA&TDE) joint method. *International Journal of Production Research*, 53, 4, 1077-1093.
- [7] Grosse, E.H., Glock, C.H., Jaber, M.Y., and Neumann, W.P. 2015. Incorporating human factors in order picking planning models: framework and research opportunities. *International Journal of Production Research*, 53, 3, 695-717.
- [8] Roodbergen, K.J., and Vis, I.F.A. 2006. A model of warehouse layout. *IIE Transactions*, 38, 10, 799-811.
- [9] Gu, J., Goetschalckx, M., and McGinnis, L. 2007. Research on warehouse operation: A comprehensive review. *European Journal of Operations Research*, 177, 1, 1-21.
- [10] de Koster, M., To, L., and Zaerpour, N. 2012. Determining of number of zones in a pick-and-sort order picking system. *International Journal of Production Research*, 50, 3, 757-771.
- [11] Kulak, O., Sahin, Y., and Taner, M.E. 2012. Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms. *Flexible Services and Manufacturing Journal*, 24, 1, 52-80.
- [12] Henn, S. 2010. Algorithms for on-line order batching in an order-picking warehouse. In *Proceedings of the 3rd International Conference on Information Systems, Logistics and Supply Chain ILS 2010* (Casablanca, Morocco, April 13-17, 2010). Business Process Consulting.
- [13] Yu, M., and de Koster, R. 2009. The impact of order batching and picking area zoning on order picking system performance. *European Journal of Operational Research*, 198, 2, 480-490.
- [14] Tetemke, M. 2013. *Improving order-picking efficiency via storage assignments strategies*. Master Thesis. Enschede: University of Louisville - Universiteit Twente.
- [15] Le-Duc, T., and de Koster, R. 2004. Travel distance estimation in single-block ABC-storage strategy warehouses. In *Distribution logistics: Advanced solutions to practical problems*, B. Fleischmann, and A. Klose Eds. Springer, 184-200.

- [16] Le-Duc, T., and de Koster, R. 2005. Travel distance estimation and storage zone optimization in a 2-block class-based storage strategy warehouse. *International Journal of Production Research*, 43, 17, 3561-3581.
- [17] Roodbergen, K., Gunter, P., and Iris, F. 2008. Designing the layout structure of manual order picking areas in warehouses. *IIE Transactions*, 40, 1032-1045.
- [18] Berglung, P., and Batta, R. 2012. Optimal placement of warehouse cross-aisles in a picker-to-part warehouse with class-based storage. *IIE Transactions*, 44, 2, 107-120.
- [19] Petersen, C. 2002. Considerations in order picking zone configuration. *International Journal of Operations & Production Management*, 27, 7, 793-805.
- [20] Gue, K., and Meller, R. 2009. Aisle configurations for unit-load warehouses. *IIE Transactions*, 41, 3, 171-182.
- [21] Gue, K., Ivanović, G., and Meller, R. 2012. A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. *Transportation Research Part E: Logistics and Transportation Review*, 48, 4, 795-806.
- [22] Eisenstein, D. 2008. Analysis and optimal design of discrete order picking technologies along a line. *Naval Research Logistics (NRL)*, 55, 4, 350-362.
- [23] de Koster, M., Le-Duc, T., and Roodbergen, K. 2007. Design and Control of Warehouse Order Picking: A Literature Review. *European Journal of Operational Research*, 182, 2, 481-501.
- [24] Tompkins, J. 2003. *Facilities Planning (3 ed.)*. Hoboken., NJ: Wiley.
- [25] Arnold, D., and Furmans, K. 2009. *Materialfluss in Logistiksystemen 6 ed.* Berlin: Springer.
- [26] Roodbergen, K.J., and Vis, I.F.A. 2006. A model of warehouse layout. *IIE Transactions*, 38, 10, 799-811.
- [27] Petersen, C. 1997. An evaluation of order picking routeing policies. *International Journal of Operations & Production Management*, 17, 11, 1098-1111.
- [28] Grosse, E., Glock, C., and Ballester-Ripoll, R. 2014. A simulated annealing approach for the joint order batching and order picker routing problem with weight restrictions. *International Journal of Operations and Quantitative Management*, 20, 2, 65-83.
- [29] Matusiak, M., de Koster, R., Kroon, L., and Saarinen, J. 2014. A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. *European Journal of Operational Research*, 236, 3, 968-977.
- [30] Henn, S., and Schmid, V. 2013. Metaheuristics for order batching and sequencing in manual order picking systems. *Computers & Industrial Engineering*, 66, 2, 338-351.
- [31] Hong, S., Johnson, A.L., and Peters, B.A. 2012. Batch picking in narrow-aisle order picking systems with consideration for picker blocking. *European Journal of Operational Research*, 221, 3, 557-570.
- [32] Schleyer, M., and Gue, K. 2012. Throughput time distribution analysis for a one-block warehouse. *Transportation Research Part E: Logistics and Transportation Review*, 48, 3, 652-666.
- [33] Hong, S., Johnson, A.L., and Peters, B.A. 2012. Large-scale order batching in parallel-aisle picking systems. *IIE Transactions*, 44, 2, 88-106.
- [34] Gong, Y.M., and de Koster, R. 2008. A polling-based dynamic order picking system for online retailers. *IIE Transactions*, 40, 11, 1070-1082.
- [35] Rubrico, J.I.U., Higashi, T., Tamura, H., and Ota, J. 2011. Online rescheduling of multiple picking agents for warehouse management. *Robotics and Computer-Integrated Manufacturing*, 27, 1, 62-71.
- [36] Bukchin, Y., Khmelnitsky, E., and Yakuel, P. 2012. Optimizing a dynamic order-picking process. *European Journal of Operational Research*, 219, 2, 335-346.
- [37] Öncan, T. 2013. A Genetic Algorithm for the order batching problem in low-level picker-to-part warehouse systems. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013* (Hong Kong, March 13-15, 2013).
- [38] Hsu, C., Chen, K., and Chen, M. 2005. Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Computers in Industry*, 56, 169-178.
- [39] Mühlenbein, H., and Paß, G. 1996. From recombination of genes to the estimation of distributions: I. binary parameters. In *Parallel Problem Solving from Nature PPSN IV*, H. Voigt, W. Ebeling, I. Rechenberg, and H. Schwefel Eds. Springer, Berlin, 178-187.
- [40] Chen, S.-H., Chen, M.-C., Chang, P.-C., Zhang, Q., and Chen, Y.-M. 2010. Guidelines for developing effective Estimation of Distribution Algorithms in solving single machine scheduling problems. *Expert Systems with Applications*, 37, 6441-6451.
- [41] Liu, H., Gao, L., and Pan, Q. 2011. A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem. *Experts Systems with Applications*, 38, 4348-4360.
- [42] Pan, Q.-K., and Ruiz, R. 2012. An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. *Omega*, 40, 166-180.
- [43] Pérez, R., Hernández, A. and S, Jöns. 2015. A continuous estimation of distribution algorithm for the online order-batching problem. *International Journal of Advanced Manufacturing Technology*, 79, 1-4, 569-588.
- [44] Greenwood, A., Vanguri, S., Eksioglu, B., Jain, P., Hill, T., and Miller, J. 2005. Simulation Optimization Decision Support System for Ship Panel Shop Operations. In *Proceedings of the 2005 Winter Simulation Conference*, M. E. Kuhl, N. M. Steiger, F. Armstrong, and J. Joines Eds. 2078-2086.
- [45] Larrañaga, P., and Lozano, J. 2002. *Estimation of distribution algorithms: a new tool for evolutionary computation*. Kluwer Academic Publishers.