Int J Adv Manuf Technol (2015) 79:569–588
DOI 10.1007/s00170-015-6835-6

ORIGINAL ARTICLE

A continuous estimation of distribution algorithm for the online order-batching problem

Ricardo Pérez-Rodríguez · Arturo Hernández-Aguirre · S. Jöns

Received: 2 September 2014 / Accepted: 18 January 2015 / Published online: 12 February 2015

Abstract In manual order-picking systems such as picker-to-parts, order pickers walk through a warehouse in order to pick up articles required by customers. Order batching consists of combining these customer orders into picking orders. In online batching, customer orders arrive throughout the scheduling. This paper considers an online order-batching problem in which the turnover time of all customer orders has to be minimized, i.e., the time period between the arrival time of the customer order and its completion time. A continuous estimation of distribution algorithm-based approach is proposed and developed to solve the problem and implement the solution. Using this approach, the warehouse performance can be noticeably improved with a substantial reduction in the average turnover time of a set of customer orders.

Keywords Estimation of distribution algorithm · Warehouse management · Order picking · Order batching · Online optimization

1 Introduction

Estimation of distribution algorithms (EDA), introduced by Mühlenbein and Paaß [20], have been successfully used to solve complex combinatorial optimization problems. Some papers such as Chen et al. [2], Liu et al. [19], Pan and Ruiz [22] are examples of combinatorial optimization problems. Although the loss of diversity and insufficient use of location information of solutions are disadvantages of EDAs, these have been tackled successfully by incorporating other methods such as genetic algorithms (GA) or tabu search (TS) during the evolutionary process. Pérez et al. [23] use this approach. Several works have been done in order to capture the problem structure with more precision, Ganesan et al. [7] and Le Dinh et al. [17] work on this issue. Advanced probabilistic models, which solve combinatorial optimization problems through EDAs, have been proposed to an attempt to integrate higher order interactions to enhance the solution quality. Wang et al. [26] have contributed to research on it.

The order-picking problem is a combinatorial issue. It has been studied and solved through different approaches such as GAs. Öncan's research [21] is in this category. Although important results have been published on order-picking problems using GAs, the individual's inadequate representation related to combinatorial problems contributes to random changes over the offspring that can disturb the solutions and does not have positive effects on the fitness. Facing that situation, an approach is to manipulate the individual's representation to prevent disorders among variables of the problem. Goldberg et al. [8], Kargupta [14], and Harik [10] offer good examples of this approach. The disadvantage of this approach is that it cannot determine the relationship or interaction between variables of the problem. Another approach consists of using EDAs. The motivation for their use is the identification and exploitation of the interaction among the variables involved in the problem in order to assist in the development of the algorithm. The idea is to learn and benefit from the interaction among variables by estimating the distribution of the population and sample from this distribution of the offspring. Although the interaction may or may not be present, generally, this is explicitly unknown even in simple order-picking warehouses.

R. Pérez-Rodríguez (✉) · A. Hernández-Aguirre
CIMAT, A. C., Callejón de Jalisco s/n, Mineral de Valenciana, C.P. 36240 Guanajuato, Guanajuato, México
e-mail: ricardo.perez@cimat.mx

S. Jöns
CONACYT, Av. Insurgentes Sur 1582, Crédito Constructor, C.P. 03940 Benito Juárez, Distrito Federal, México

Henn [11] stated that, when different incoming articles are received and stored in unit loads such as pallets or racks and the customers require a few quantities of different articles, the order picking arises [5].

Order-picking systems involving operators are very common in industry. Henn [11] explained that "manual order-picking systems can be divided into two categories [27]: parts-to-picker, where automated storage and retrieval systems deliver the articles to stationary pickers; picker-to-parts, where pickers normally walk through the warehouse and collect articles." Manual picking systems such as picker-to-parts where the activity of transformation of customer orders into batches (picking orders) is analyzed in this paper. 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. It means that different customer orders can be simultaneously released for picking. If the customer orders are available at the beginning of the scheduling, the order batching is called off line batching, but if the customer orders arrive throughout the scheduling, the order batching is known online batching [29]. It means that customer orders become available dynamically over time. Therefore, the orders are not known in advance. At a specific point in time, it is only known if at least one order will arrive. However, no information regarding the number of orders or their characteristics is given. Thus, deciding which orders should be processed directly has to be done without access to information about those orders. In this paper, the online batching where the batches have to be formed based only on known orders is considered.

Although different algorithms have been proposed in order to solve the online order-batching problem successfully, these algorithms have been proposed without consideration for the relationship or interaction that can exist between different customer orders. For this reason and contrary to current research, the aim of this paper is to find a relationship or interaction between different customer orders known and available in order to build better batches for picking. The global idea is to estimate a relationship or interaction among different customer orders through an EDA. We propose the order batching–continuous estimation of distribution algorithm (OBCEDA) to guide the estimation mentioned above on the overall search process. To the best of our knowledge, this kind of algorithm has not been used to tackle the order-batching problem in order-picking warehouses.

2 Problem statement

Based on Henn [12], the online order-batching problem and an optimization model are explained below.

The picking process being analyzed here is described in the following way: The operators (pickers) start at a depot, and the picker 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 route considered by the picker in order to find and collect articles in this research is the "S-Shape" route because it is easy to understand and has been widely used in industrial environments. Figure 1 shows an example of the "S-Shape" route for a set of articles to be picked. The black rectangles represent pick locations, i.e., the corresponding locations where articles have to be picked.

The order-picking process is usually done with the help of a picking device. 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, since it would result in additional unacceptable sorting efforts. If the picker has already started a tour, it is completed without interruption.

In this paper, a single order picker is considered, i.e., all batches must be processed one after another. Specifically, in the online batching, there is no information given about how many orders or their characteristics will arrive. The decision, 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 in time when an order picker starts to process this batch. The start time of an order is identical to the start time of the batch the order is assigned to. The point in time when the order picker returns to the depot after collecting all articles is called 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 for which an order stays in the system, i.e., the time period between the completion and the start time of an order. This paper focuses on minimizing the average turnover time of all customer orders. All algorithms for the online order-batching problem, which are analyzed in the literature review section below, form and release batches without having complete information on the types and the arrival times of future orders. Therefore, 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 be released directly, or its start should be postponed to a later point in time.

Let n be the number of customer orders known,

m the number of batches to be processed,

a_i the arrival time of order i for all $i \in \{1, \ldots, n\}$,

k_i the number of articles of order i,

K the maximal number of articles that can be included in any batch (device capacity),

S_j the start time of the batch j for all $j \in \{1, \ldots, m\}$

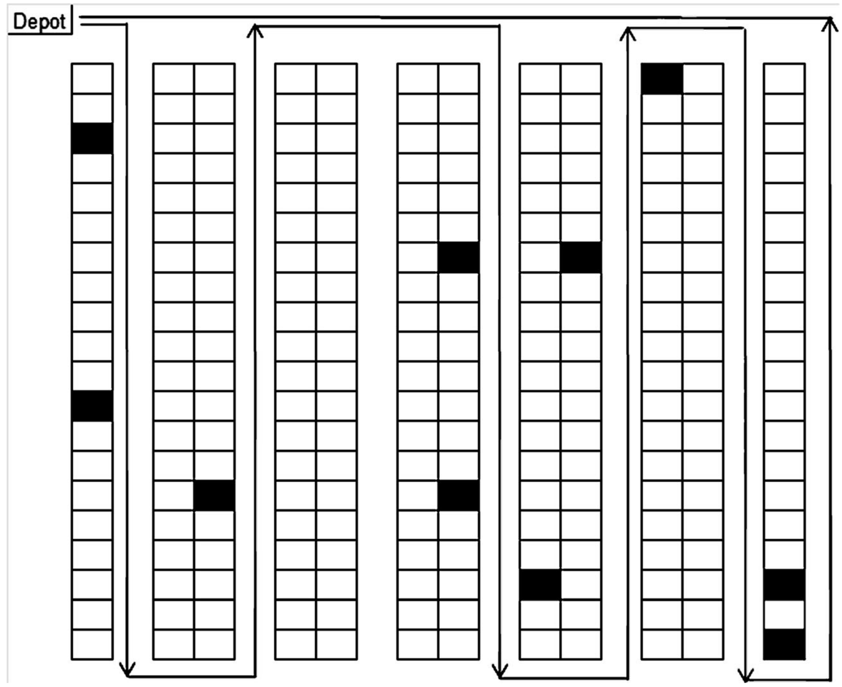

Fig. 1 Example of "S-shape" route

E_j the end time of the batch j for all $j \in \{1, \ldots, m\}$

$x_{ji} = \{1$ if order i is assigned to batch j, 0 otherwise$\}$

An optimization model can be formulated as follows.

The main idea is to minimize the average turnover time that can be defined by

$$\frac{\sum_{j=1}^{m} E_j - S_j}{m} \tag{1}$$

Equation (2) ensures the assignment of each order to exactly one batch. It is possible by means

$$\sum_{j=1}^{m} x_{ji} = 1,\ \forall i \in \{1,\ \ldots,\ n\} \tag{2}$$

Inequalities (3) guarantee that the capacity of the picking device is not violated with

$$\sum_{i=1}^{n} k_i x_{ji} \leq K,\ \forall j \in \{1, \ldots, m\} \tag{3}$$

The conditions (4) indicate that a batch is started after all customer orders assigned to that batch are known using

$$S_j \geq \max\{a_i \cdot x_{ji}\},\ \forall i \in \{1, \ldots,\ n\},\ \forall j \in \{1, \ldots, m\} \tag{4}$$

From Eq. (5) follows that a batch is started after the previous one is completed by means

$$S_j \geq E_{j-1}\ \forall j \in \{2, \ldots, m\} \tag{5}$$

Finally,

$$S_j \geq 0 \tag{6}$$

$$x_{ji} \in \{0, 1\} \tag{7}$$

3 Literature review

Henn [11] detailed a discussion about the most current research on the order-batching process. A part of that discussion is outlined below.

Kamin [13] analyzed a practical batching problem where greeting cards are retrieved from a warehouse. Pickers use automated-guided vehicles on a fixed course collecting the items according to given customer orders. Those orders arrive throughout the study horizon, and this research focuses on the minimization of average turnover times.

Chew and Tang [3] focus on the optimal number of customer orders that should be assigned to a batch such that the average turnover time is minimized. They employ a queuing network with two queues. In the first queue, customer orders arrive according to a Poisson process and batches are generated by means of the first come first serve (FCFS) rule. If a particular number of customer orders are in the first queue, those orders are assigned to a batch and move onto the second queue. Those orders are released according to the availability of pickers.

A study of the average turnover time of a random customer order for a two-block layout is carried out by Le-Duc and de Koster [18]. A corresponding model for all customer orders arriving during a particular time interval are assigned to batches in a two-block layout is presented by van Nieuwenhuyse and de Koster [25].

Yu and de Koster [29] explain an order-picking area that is divided into several zones of identical size. The articles of each batch are picked up sequentially by zones. For this picking process, the researchers give an estimation of the average turnover times and observe that an optimal batch size exists.

Henn [12] describes an online order-batching problem in a walk-and-pick warehouse in which the completion times of all (dynamically arriving) customer orders (or the makespan) are to be minimized. The author also shows modifications of solution approaches for offline order batching in order to deal with the online situation.

Öncan [21] introduces a GA for the order-batching problem considering traversal and return routing policies. The proposed GA is tested on randomly generated instances and compared with the well-known savings algorithm. According to the author's extensive computational experiments, we can say that the proposed GA yields promising solutions in acceptable computation times.

The main characteristic in all this current research is the common representation of the solution. The authors employ discrete vectors where the number of elements equals the total number of orders to pick up and where each element contains an integer value that represents the batch to be assigned. In addition, the traditional evolutionary operators used in current research do not try to learn about the relationship between variables. These were not built for that purpose.

Although different algorithms have been proposed in order to solve the online order-batching problem successfully, EDAs have not been considered in this perspective. To the best of our knowledge, this kind of algorithm has not been used to tackle the order-batching problem in order-picking warehouses.

4 OBCEDA—for the online batching problem

EDA is a relatively new paradigm in the field of evolutionary computation. Compared with other evolutionary algorithms, the EDA reproduces new population implicitly instead of using traditional evolutionary operators. In the 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. The updating process reflects the evolutionary trend of the population [16].

The OBCEDA is proposed and explained to solve the online order-batching problem in this section. We introduce the solution representation, population initialization, a weakness approach of the probability model, and the probability model proposed.

4.1 Solution representation

In this paper, a solution to the online batching problem previously mentioned is expressed by the assignment of customer orders to batches, i.e., 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 a random value $U[0,1]$, an important difference between our approach and others. This representation is shown in Fig. 2 with 10 orders. This continuous vector does not have a real meaning on the solution that it represents.

4.2 Generation of the population

Initial population members are generated randomly in order to enable a wide range of solutions [9].

4.3 A weakness approach of the probability model

A primary approach for the probability model is to design a probability matrix, i.e., a batch probability matrix. The element p_{ji} of the probability matrix would represent the probability that batch j were used for the i customer order. For all j ($j = 1, \ldots, m$) and for all i ($i = 1, \ldots, n$). The value of p_{ji} would indicate the opportunity of a customer order on a certain batch. Via sampling according to the probability matrix, new individuals could be generated. For every position i, batch j would be selected with the probability p_{ji}. If batch j has already been filled according to the device capacity, it would mean the assignment of batch j has been finished. Then, the whole column $p_{j1}, p_{j2}, \ldots, p_{ji}$ of the probability matrix would be set as zero. This updating mechanism would consider the previous assignments. Although in this approach, the probabilistic model would be updated each time an order is assigned in a batch and this updating would eliminate the possibility of choosing a previous batch, a modification in the sampling process has to be carried out if we would set as zero the whole column of probability matrix mentioned above. In addition, this approach does not consider if exists a relationship between the previous position result and the current position result.

To demonstrate the weakness of the approach above, different trials were carried out assuming that there is no relationship between customer orders. The performance obtained was

Fig. 2 Solution vector

	Customer Orders									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Batch	0.06676	0.37018	0.72182	0.5547	0.80072	0.30412	0.86297	0.43967	0.65066	0.31588

compared with the continuous approach proposed in this research, and these are shown in Section 5. The continuous approach proposed is explained below.

4.4 Probability model proposed

In order to avoid modifying the sampling process mentioned above and to demonstrate that there is a relationship between customer orders, we adopted a continuous optimization procedure instead of a discrete one to solve the online batching problem. This is an important difference between this approach and the current research. The advantage of this representation for each individual, through continuous values, is that they do not have direct meaning to the solution they represent. There is no problem if each individual does not explicitly shows its information on the sequence of batches to be processed. It is not necessary that the probabilistic model be updated each time an order is assigned in the batch sequence, and it is not necessary to make any modification in the sampling process. Rudolph [24] and Bean and Norman [1] can be consulted about continuous optimization procedures. We used the $\mathrm{MIMIC}_{\mathrm{C}}^{\mathrm{G}}$ algorithm to build the probabilistic model introduced by Larrañaga et al. [15]. It is an adaptation of the MIMIC algorithm presented by De Bonet et al. [4] to continuous domains. The $\mathrm{MIMIC}_{\mathrm{C}}^{\mathrm{G}}$ algorithm uses a chain structured probabilistic model where the probability distribution of all the variables except the head node is conditioned on the value of the variable preceding them in the chain. It means a marginal univariate function and $n-1$ pairs of conditional density functions to build the probabilistic model. Thus, the current position result for any customer order is conditioned to the previous position result.

Once the individuals have been generated from the algorithm $\mathrm{MIMIC}_{\mathrm{C}}^{\mathrm{G}}$, they must be decoded to be represented as a valid batch sequence. Hence, we need a method to decode these real vectors into discrete vectors. Figure 3 details an example of a real vector and its decoding.

The major procedure of the OBCEDA is listed as follows:

Step 1. Set the generation index $g=0$. Initialize an initial population $S(0)$ of size M.

Step 2. Select a subset D from $S(g)$ of size N, where $N \leq M$.

Step 3. Establish a probabilistic model P according to $\mathrm{MIMIC}_{\mathrm{C}}^{\mathrm{G}}$, which describes the distribution characteristics of D.

Step 4. Generate a set K of new individuals by sampling P.

Step 5. Select the best individuals from $K \cup S(g)$ and assign them to the next generation $S(g+1)$.

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

Figure 4 details the overall OBCEDA process, and Fig. 5 depicts a flowchart about the algorithm process.

Fig. 3 Representation of an individual to a valid batch sequence

	Customer Orders									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real value	0.09512	0.78667	0.4657	0.40552	0.15579	0.67031	0.05324	0.42156	0.22162	0.88939

Step 1. Find the minimum value. Assign to batch

Capacity device 45	Customer Orders									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real value	0.09512	0.78667	0.4657	0.40552	0.15579	0.67031	0.05324	0.42156	0.22162	0.88939
Articles	6	35	36	6	8	29	45	18	14	33
Batch							1			

Step 2. Find the next minimum value. Assign to batch if the capacity device is available. If not, assign to another batch

Capacity device 45	Customer Orders									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real value	0.09512	0.78667	0.4657	0.40552	0.15579	0.67031	0.05324	0.42156	0.22162	0.88939
Articles	6	35	36	6	8	29	45	18	14	33
Batch	2						1			

Step 3. Return step 2 until all customer orders have been assigned to

Capacity device 45	Customer Orders									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real value	0.09512	0.78667	0.4657	0.40552	0.15579	0.67031	0.05324	0.42156	0.22162	0.88939
Articles	6	35	36	6	8	29	45	18	14	33
Batch	2	6	4	2	2	5	1	3	2	7

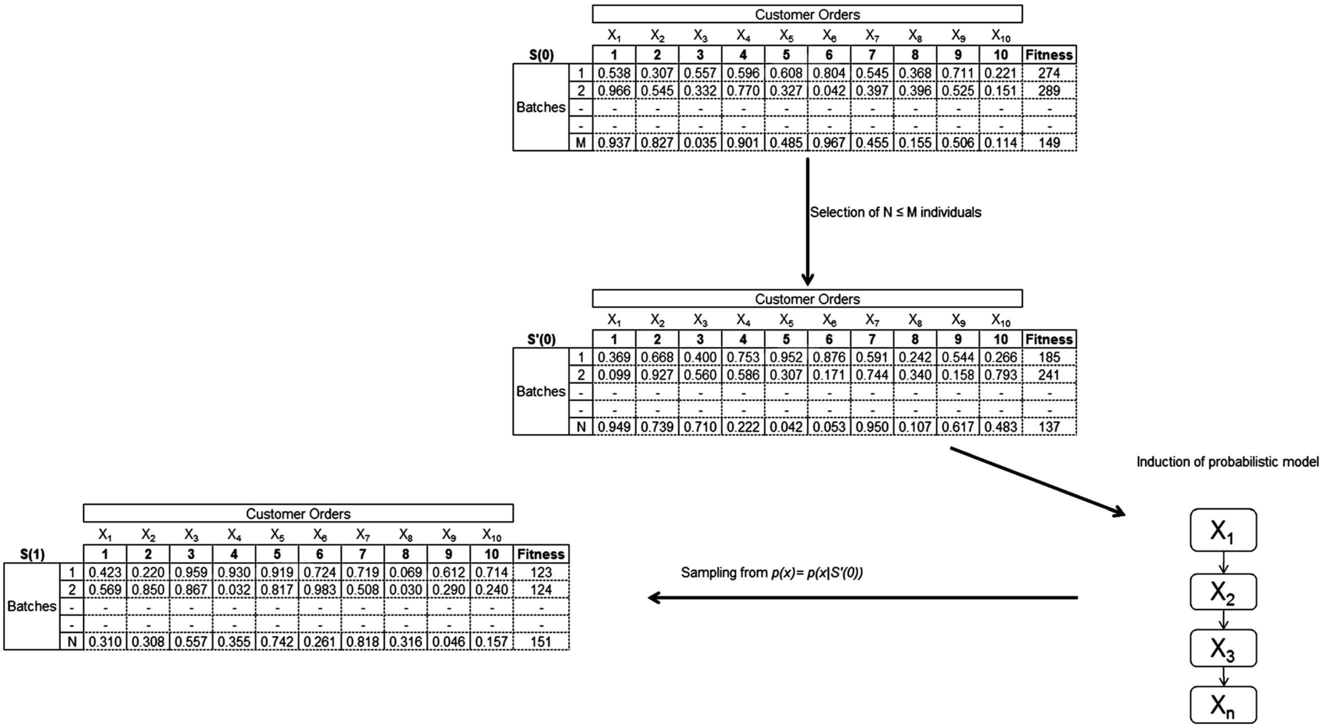

Fig. 4 The OBCEDA algorithm

5 Results and comparison

A GA is proposed as a benchmark for comparison with the OBCEDA scheme. GA works with tournament selection. The "edge recombination operator" is used as a cross operator based on Whitley et al. [28], and a mutation operator changes batches among different positions.

A TS based on Henn [11] is utilized as a benchmark for comparison with the OBCEDA scheme, too. For the local search phase, we implemented the operator "swap move," meaning the interchanging of two customer orders from different batches. A tabu list is built on 10 neighbors at most.

We used a Dell® Vostro® 3500 computer, Intel® Core™ i3 processor, 2.6 GHZ, 4 GB of RAM, Windows® 7 for 64 bits to run each algorithm. The algorithms have been encoded in DevC++®.

To account for the stochastic nature of the order-picking warehouse, we ran 50 trials for the algorithms where 75 individuals belong to each generation. Each trial contains a stop criterion. It means that any trial returns the optimal vector when the difference between the average fitness of the trial and the best is <5 %.

We established a workload to evaluate and find the best average turnover time. The workload mentioned 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. 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 Henn [11]. 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 one. The depot is in the lefthand corner of the warehouse. We further assume that an order picker walks through 10 storage locations in 30 s, and they need 10 s 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 was used. For an order, we choose the quantity of articles uniformly distributed in $\{5, \ldots, 25\}$, resulting in three or five orders per batch on average, in accordance with the aforementioned capacities of the picking device. For the total number of orders n, we consider 30 and 60. The orders should arrive within a planning period of 8 h. The inter-arrival times, i.e., the time between the arrival of order i and order $i+1$, are exponentially distributed with a parameter λ called 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 $E[X(t)] = \lambda\ t$ holds. In our numerical experiments, we choose λ in a way that the expectation $E[X(t)]$ is equal to n for $t = 8[h]$. In summary, we use the following values for λ: for $n = 30$, $\lambda = 0.08625$; for $n = 60$, $\lambda = 0.125$.

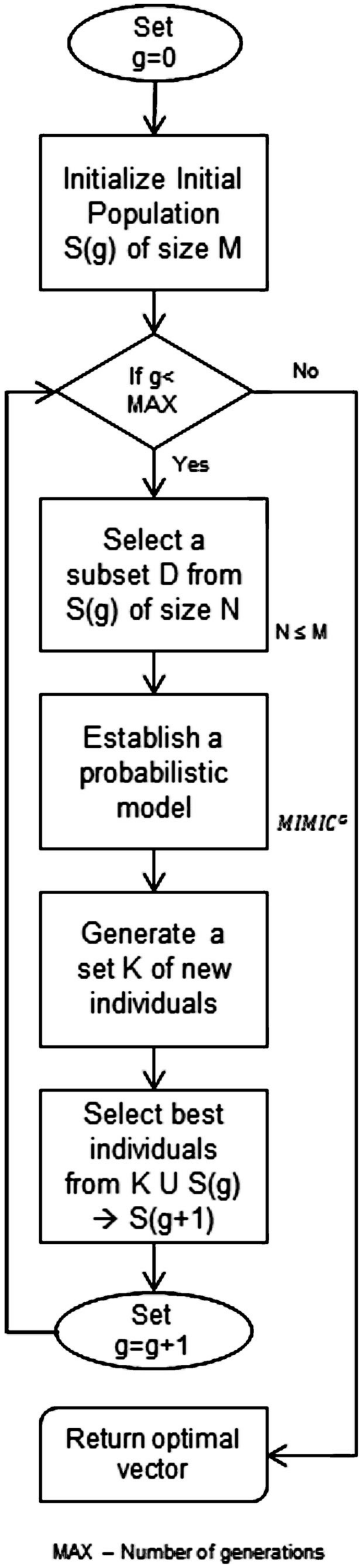

Fig. 5 The algorithm process

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

$$\mathrm{RPI}(c_i) = \left(c_i - c^*\right)/c^* \times 100 \tag{8}$$

where c_i is the average turnover time obtained in the ith 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.

Table 1 details the average obtained for each trial when the total number of orders is 30 and the K capacity is 45 articles. We analyze the performance between averages of GA and OBCEDA algorithms.

As we can see in Table 1, there is a significant difference between the averages of both algorithms. The performance of OBCEDA was superior in 28 of the 50 trials.

Table 2 shows the variance obtained for each trial. We analyze whether there is a statistically significant difference between variances of both algorithms.

According to Table 2, there is no statistically significant difference between variances of both algorithms. The performance was the same in 50 of the 50 trials with α=0.01 of significance level. We consider that the stability of both algorithms is practically the same (100 % of the time).

Table 3 details the average obtained for each trial when the total number of orders is 30 and the K capacity is 75 articles. We analyze the performance between averages of GA and OBCEDA algorithms.

On Table 3, there is a significant difference between the averages of both algorithms. The performance of OBCEDA was superior in 46 of the 50 trials.

Table 4 shows the variance obtained for each trial. We analyze whether there is a statistically significant difference between variances of both algorithms.

As we can see in Table 4, there is no statistically significant difference between variances of both algorithms. The performance was the same in 50 of the 50 trials with α=0.01 of significance level. We consider that the stability of both algorithms is practically the same (100% of the time).

Table 5 details the average obtained for each trial when the total number of orders is 60 and the K capacity is 45 articles. We analyze the performance between averages of GA and OBCEDA algorithms.

Based on Table 5, there is a significant difference between the averages of both algorithms. The performance of OBCEDA was superior in 29 of the 50 trials.

Table 6 shows the variance obtained for each trial. We analyze whether there is a statistically significant difference between variances of both algorithms.

Table 1 Comparison of results for each average with n=30, K=45

Trial	GA μ_1	OBCEDA μ_2
1	0.0683	0.0533
2	0.0278	0.0039
3	0.0827	0.1028
4	0.1017	0.1246
5	0.1523	0.0428
6	0.0961	0.0529
7	0.0919	0.0443
8	0.0743	0.0671
9	0.1493	0.0554
10	0.0256	0.1369
11	0.1377	0.1523
12	0.1129	0.1033
13	0.1224	0.0314
14	0.0929	0.0618
15	0.0306	0.0510
16	0.0609	0.0389
17	0.1057	0.0706
18	0.1087	0.0551
19	0.1082	0.0454
20	0.1053	0.0566
21	0.1145	0.0464
22	0.0881	0.0011
23	0.1217	0.0039
24	0.0721	0.0093
25	0.0084	0.0518
26	0.0939	0.0775
27	0.0566	0.1231
28	0.0089	0.1160
29	0.0541	0.0098
30	0.0057	0.1033
31	0.0396	0.0940
32	0.0549	0.0953
33	0.1173	0.0683
34	0.1672	0.0759
35	0.1290	0.0526
36	0.0306	0.0207
37	0.0443	0.0329
38	0.0477	0.0611
39	0.0120	0.0455
40	0.0492	0.0967
41	0.0241	0.0487
42	0.0060	0.0185
43	0.0000	0.0185
44	0.0731	0.0905
45	0.0082	0.0000
46	0.0951	0.1400
47	0.1469	0.0454
48	0.0374	0.1412
49	0.0196	0.1099
50	0.0896	0.0910
μ the average	22/50	28/50

Table 2 Comparison of results for each variance with n=30, K=45

Trial	GA σ^2	OBCEDA σ^2	σ^2/σ^2	H_0: $\sigma^2=\sigma^2$ α=0.01 F_c<0.545 or F_c>1.832
1	1.8576	2.0551	0.9039	0.9038*
2	1.7491	1.9120	0.9148	0.9148*
3	1.8960	2.1986	0.8623	0.8623*
4	1.9468	2.2620	0.8607	0.8606*
5	2.0823	2.0248	1.0284	1.0283*
6	1.9317	2.0541	0.9404	0.9404*
7	1.9206	2.0291	0.9465	0.9465*
8	1.8735	2.0954	0.8941	0.8940*
9	2.0743	2.0615	1.0062	1.0061*
10	1.7431	2.2979	0.7586	0.7585*
11	2.0432	2.3424	0.8722	0.8722*
12	1.9767	2.2004	0.8983	0.8983*
13	2.0020	1.9918	1.0051	1.0051*
14	1.9232	2.0800	0.9246	0.9246*
15	1.7567	2.0485	0.8576	0.8575*
16	1.8375	2.0135	0.9126	0.9126*
17	1.9574	2.1055	0.9297	0.9296*
18	1.9655	2.0606	0.9538	0.9538*
19	1.9642	2.0323	0.9665	0.9664*
20	1.9565	2.0649	0.9475	0.9475*
21	1.9812	2.0354	0.9734	0.9733*
22	1.9103	1.9040	1.0033	1.0032*
23	2.0005	1.9120	1.0463	1.0462*
24	1.8677	1.9277	0.9688	0.9688*
25	1.6974	2.0510	0.8276	0.8276*
26	1.9261	2.1255	0.9062	0.9061*
27	1.8261	2.2577	0.8088	0.8088*
28	1.6986	2.2371	0.7593	0.7592*
29	1.8194	1.9292	0.9431	0.9430*
30	1.6902	2.2004	0.7681	0.7681*
31	1.7808	2.1733	0.8194	0.8193*
32	1.8216	2.1772	0.8367	0.8366*
33	1.9885	2.0989	0.9474	0.9474*
34	2.1219	2.1208	1.0005	1.0005*
35	2.0199	2.0534	0.9837	0.9836*
36	1.7567	1.9608	0.8959	0.8959*
37	1.7933	1.9962	0.8984	0.8983*
38	1.8023	2.0778	0.8674	0.8674*
39	1.7067	2.0328	0.8396	0.8395*
40	1.8065	2.1812	0.8282	0.8282*
41	1.7393	2.0419	0.8518	0.8518*
42	1.6909	1.9544	0.8652	0.8651*
43	1.6748	1.9543	0.8569	0.8569*
44	1.8704	2.1632	0.8646	0.8646*
45	1.6967	1.9006	0.8927	0.8927*
46	1.9291	2.3068	0.8363	0.8362*
47	2.0677	2.0323	1.0174	1.0173*
48	1.7748	2.3102	0.7683	0.7682*
49	1.7273	2.2194	0.7782	0.7782*
50	1.9144	2.1646	0.8844	0.8843*
σ^2 the variance				50/50

*There is no statistically significant difference between samples

Table 3 Comparison of results for each average with n=30, K=75

Trial	GA μ_1	OBCEDA μ_2
1	0.0487	0.0423
2	0.1066	0.0574
3	0.1713	0.0830
4	0.1153	0.0405
5	0.2359	0.0000
6	0.1150	0.0371
7	0.0303	0.0714
8	0.1745	0.0432
9	0.1018	0.0139
10	0.1016	0.0470
11	0.1848	0.0743
12	0.1168	0.0745
13	0.0645	0.0264
14	0.1377	0.0269
15	0.1048	0.0290
16	0.0997	0.0045
17	0.0583	0.0343
18	0.1333	0.0591
19	0.0873	0.0352
20	0.1371	0.0088
21	0.0493	0.0123
22	0.1233	0.0495
23	0.2256	0.0338
24	0.1056	0.0817
25	0.0196	0.0266
26	0.0790	0.0212
27	0.1365	0.0396
28	0.1884	0.0468
29	0.1066	0.0227
30	0.1595	0.0188
31	0.1664	0.0455
32	0.0761	0.0115
33	0.1440	0.0002
34	0.1181	0.0350
35	0.1201	0.0057
36	0.0334	0.0304
37	0.0779	0.0503
38	0.0675	0.0307
39	0.1966	0.0217
40	0.0812	0.0538
41	0.2116	0.0433
42	0.1277	0.0342
43	0.0932	0.0151
44	0.1432	0.0139
45	0.0238	0.0090
46	0.1326	0.0439
47	0.0711	0.0100
48	0.0000	0.0426
49	0.0302	0.0136
50	0.0341	0.0456
μ the average	4/50	46/50

Table 4 Comparison of results for each variance with n=30, K=75

Trial	GA σ^2	OBEDA σ^2	σ^2/σ^2	H_0: $\sigma^2=\sigma^2$ α=0.01 F_c<0.545 or F_c>1.832
1	3.7302	4.3949	0.8487	0.8487*
2	3.9911	4.4733	0.8922	0.8922*
3	4.2832	4.6057	0.9300	0.9299*
4	4.0304	4.3857	0.9190	0.9189*
5	4.5744	4.1760	1.0954	1.0953*
6	4.0292	4.3682	0.9224	0.9223*
7	3.6473	4.5455	0.8024	0.8024*
8	4.2973	4.3996	0.9768	0.9767*
9	3.9696	4.2478	0.9345	0.9345*
10	3.9686	4.4191	0.8980	0.8980*
11	4.3439	4.5604	0.9525	0.9525*
12	4.0372	4.5615	0.8851	0.8850*
13	3.8012	4.3126	0.8814	0.8814*
14	4.1314	4.3152	0.9574	0.9574*
15	3.9833	4.3261	0.9208	0.9207*
16	3.9601	4.1991	0.9431	0.9430*
17	3.7735	4.3535	0.8668	0.8667*
18	4.1117	4.4818	0.9174	0.9174*
19	3.9043	4.3582	0.8959	0.8958*
20	4.1289	4.2217	0.9780	0.9780*
21	3.7326	4.2399	0.8803	0.8803*
22	4.0664	4.4322	0.9175	0.9174*
23	4.5282	4.3509	1.0407	1.0407*
24	3.9866	4.5988	0.8669	0.8668*
25	3.5989	4.3138	0.8343	0.8342*
26	3.8668	4.2859	0.9022	0.9022*
27	4.1262	4.3809	0.9419	0.9418*
28	4.3599	4.4183	0.9868	0.9867*
29	3.9910	4.2937	0.9295	0.9294*
30	4.2298	4.2735	0.9898	0.9897*
31	4.2612	4.4115	0.9659	0.9659*
32	3.8538	4.2355	0.9099	0.9098*
33	4.1598	4.1772	0.9958	0.9958*
34	4.0433	4.3570	0.9280	0.9280*
35	4.0521	4.2055	0.9635	0.9635*
36	3.6611	4.3333	0.8449	0.8448*
37	3.8622	4.4363	0.8706	0.8705*
38	3.8149	4.3349	0.8801	0.8800*
39	4.3970	4.2883	1.0253	1.0253*
40	3.8766	4.4542	0.8703	0.8703*
41	4.4650	4.4004	1.0147	1.0146*
42	4.0862	4.3528	0.9387	0.9387*
43	3.9308	4.2540	0.9240	0.9240*
44	4.1560	4.2480	0.9783	0.9783*
45	3.6179	4.2225	0.8568	0.8568*
46	4.1083	4.4030	0.9331	0.9330*
47	3.8308	4.2280	0.9060	0.9060*
48	3.5103	4.3963	0.7985	0.7984*
49	3.6467	4.2467	0.8587	0.8587*
50	3.6640	4.4123	0.8304	0.8304*
	σ^2 the variance			50/50

*There is no statistically significant difference between samples

Table 5 Comparison of results for each average with n=60, K=45

Trial	GA μ_1	OBEDA μ_2
1	0.1017	0.0579
2	0.0338	0.0616
3	0.0863	0.0120
4	0.0302	0.0088
5	0.0585	0.0129
6	0.0121	0.0589
7	0.0743	0.0135
8	0.0619	0.0036
9	0.0434	0.0653
10	0.0980	0.0434
11	0.0436	0.0208
12	0.1029	0.0977
13	0.0696	0.0345
14	0.0612	0.0175
15	0.0686	0.0875
16	0.0939	0.0426
17	0.0541	0.0334
18	0.0331	0.0158
19	0.0134	0.0254
20	0.0348	0.0139
21	0.0388	0.0142
22	0.0798	0.1071
23	0.0630	0.0725
24	0.0063	0.0424
25	0.0000	0.0670
26	0.0096	0.1031
27	0.0841	0.0000
28	0.0751	0.0413
29	0.0526	0.1123
30	0.0523	0.0695
31	0.1047	0.0303
32	0.1246	0.0501
33	0.0750	0.0495
34	0.0074	0.0209
35	0.0516	0.0120
36	0.0304	0.0693
37	0.0420	0.0097
38	0.0717	0.0516
39	0.0747	0.0581
40	0.0432	0.0666
41	0.0295	0.0139
42	0.0980	0.0375
43	0.0400	0.0149
44	0.0577	0.0671
45	0.0540	0.0564
46	0.0601	0.0757
47	0.0211	0.0633
48	0.0138	0.0199
49	0.0089	0.0228
50	0.0935	0.0291
μ the average	21/50	29/50

Table 6 Comparison of results for each variance with n=60, K=45

Trial	GA σ^2	OBEDA σ^2	σ^2/σ^2	H_0: σ^2=σ^2 α=0.01 F_c<0.545 or F_c>1.832
1	2.2171	2.9458	0.7526	0.7526*
2	2.0187	2.9595	0.6821	0.6821*
3	2.1720	2.7745	0.7829	0.7828*
4	2.0082	2.7626	0.7270	0.7269*
5	2.0908	2.7778	0.7527	0.7526*
6	1.9552	2.9493	0.6630	0.6629*
7	2.1371	2.7800	0.7687	0.7687*
8	2.1006	2.7432	0.7658	0.7657*
9	2.0468	2.9735	0.6884	0.6883*
10	2.2061	2.8916	0.7630	0.7629*
11	2.0472	2.8073	0.7292	0.7292*
12	2.2205	3.0942	0.7176	0.7176*
13	2.1234	2.8584	0.7429	0.7428*
14	2.0986	2.7950	0.7508	0.7508*
15	2.1203	3.0563	0.6938	0.6937*
16	2.1943	2.8886	0.7596	0.7596*
17	2.0780	2.8546	0.7279	0.7279*
18	2.0166	2.7888	0.7231	0.7230*
19	1.9591	2.8246	0.6936	0.6935*
20	2.0217	2.7818	0.7268	0.7267*
21	2.0333	2.7828	0.7307	0.7306*
22	2.1529	3.1293	0.6880	0.6879*
23	2.1040	3.0004	0.7012	0.7012*
24	1.9385	2.8881	0.6712	0.6711*
25	1.9199	2.9798	0.6443	0.6443*
26	1.9480	3.1144	0.6255	0.6254*
27	2.1657	2.7297	0.7934	0.7933*
28	2.1393	2.8841	0.7418	0.7417*
29	2.0736	3.1485	0.6586	0.6586*
30	2.0726	2.9889	0.6934	0.6934*
31	2.2256	2.8429	0.7829	0.7828*
32	2.2839	2.9165	0.7831	0.7830*
33	2.1389	2.9145	0.7339	0.7338*
34	1.9417	2.8079	0.6915	0.6915*
35	2.0706	2.7743	0.7463	0.7463*
36	2.0086	2.9882	0.6722	0.6721*
37	2.0427	2.7659	0.7385	0.7385*
38	2.1293	2.9224	0.7286	0.7286*
39	2.1382	2.9467	0.7256	0.7256*
40	2.0459	2.9783	0.6869	0.6869*
41	2.0062	2.7816	0.7212	0.7212*
42	2.2061	2.8696	0.7688	0.7687*
43	2.0367	2.7852	0.7313	0.7312*
44	2.0885	2.9800	0.7008	0.7008*
45	2.0777	2.9400	0.7067	0.7067*
46	2.0954	3.0121	0.6957	0.6956*
47	1.9815	2.9659	0.6681	0.6680*
48	1.9603	2.8039	0.6992	0.6991*
49	1.9460	2.8150	0.6913	0.6913*
50	2.1929	2.8384	0.7726	0.7725*
	σ^2 the variance			50/50

*There is no statistically significant difference between samples

According to Table 6, there is no statistically significant difference between variances of both algorithms. The performance was the same in 50 of the 50 trials with α=0.01 of significance level. We consider that the stability of both algorithms is practically the same (100% of the time).

Table 7 details the average obtained for each trial when the total number of orders is 60 and the K capacity is 75 articles. We analyze the performance between averages of GA and OBCEDA algorithms.

On Table 7, there is a significant difference between the averages of both algorithms. The performance of OBCEDA was superior in 37 of the 50 trials.

Table 8 shows the variance obtained for each trial. We analyze whether there is a statistically significant difference between variances of both algorithms.

Based on Table 8, there is no statistically significant difference between variances of both algorithms. The performance was the same in 50 of the 50 trials with α=0.01 of significance level. We consider that the stability of both algorithms is practically the same (100% of the time).

The same GA is used for comparison with the primary approach, i.e., batch probability matrix in the same stochastic nature of the order-picking warehouse.

Table 9 details the average obtained for each trial when the total number of orders is 30 and the K capacity is 45 articles. We analyze the performance between averages of both algorithms.

As we can see in Table 9, there is no a significant difference between the averages of both algorithms.

Table 10 shows the average obtained for each trial when the total number of orders is 30 and the K capacity is 75 articles. We analyze the performance between averages of both algorithms.

On Table 10, there is no a significant difference between the averages of both algorithms.

Table 11 details the average obtained for each trial when the total number of orders is 60 and the K capacity is 45 articles. We analyze the performance between averages of both algorithms.

According to Table 11, there is no a significant difference between the averages of both algorithms.

Table 12 shows the average obtained for each trial when the total number of orders is 60 and the K capacity is 75 articles. We analyze the performance between averages of both algorithms.

Based on Table 12, there is no a significant difference between the averages of both algorithms.

Table 13 details the average obtained for each trial when the total number of orders is 30 and the K capacity is 45 articles. We analyze the performance between averages of TS and OBCEDA algorithms.

As we can see in Table 13, there is no a significant difference between the averages of both algorithms. The performance of OBCEDA was almost equal to the TS performance.

Table 7 Comparison of results for each average with n=60, K=75

Trial	GA μ_1	OBEDA μ_2
1	0.0983	0.0800
2	0.0777	0.0745
3	0.1062	0.0753
4	0.0767	0.0394
5	0.0716	0.0775
6	0.0875	0.0297
7	0.0862	0.1110
8	0.0276	0.0254
9	0.0443	0.0290
10	0.0484	0.0166
11	0.0342	0.0628
12	0.0896	0.0400
13	0.1073	0.0758
14	0.0374	0.0028
15	0.0805	0.0263
16	0.0035	0.0055
17	0.0814	0.1256
18	0.0858	0.0471
19	0.0784	0.0435
20	0.1546	0.0375
21	0.0164	0.0000
22	0.0292	0.0168
23	0.0000	0.0445
24	0.0490	0.0295
25	0.0387	0.0364
26	0.0442	0.0646
27	0.0812	0.1125
28	0.0024	0.0681
29	0.0737	0.0146
30	0.1107	0.1534
31	0.0953	0.0441
32	0.0907	0.0573
33	0.1233	0.0480
34	0.0333	0.0296
35	0.1010	0.0798
36	0.0778	0.0722
37	0.1000	0.0645
38	0.0276	0.0086
39	0.0028	0.0027
40	0.0622	0.0621
41	0.0549	0.0500
42	0.0735	0.0283
43	0.0445	0.0253
44	0.1045	0.0012
45	0.1071	0.0156
46	0.0556	0.0189
47	0.0742	0.0525
48	0.0898	0.1151
49	0.0029	0.0196
50	0.0272	0.0353
μ the average	13/50	37/50

Table 8 Comparison of results for each variance with n=60, K=75

Trial	GA	OBCEDA		H_0: $\sigma^2=\sigma^2$ α=0.01
	σ^2	σ^2	σ^2/σ^2	F_c<0.545 or F_c>1.832
1	3.5983	4.3354	0.8300	0.8299*
2	3.5123	4.1280	0.8508	0.8508*
3	3.6314	4.1316	0.8789	0.8789*
4	3.5080	3.9604	0.8858	0.8857*
5	3.4865	4.1419	0.8418	0.8417*
6	3.5533	3.9142	0.9078	0.9077*
7	3.5477	4.3019	0.8247	0.8246*
8	3.3022	3.9603	0.8338	0.8338*
9	3.3721	4.1111	0.8202	0.8202*
10	3.3895	3.8516	0.8800	0.8800*
11	3.3299	4.0718	0.8178	0.8178*
12	3.5619	3.9630	0.8988	0.8988*
13	3.6359	4.1341	0.8795	0.8795*
14	3.3435	3.7857	0.8832	0.8832*
15	3.5240	3.8978	0.9041	0.9041*
16	3.2014	3.7984	0.8428	0.8428*
17	3.5278	4.3717	0.8070	0.8069*
18	3.5462	3.9971	0.8872	0.8871*
19	3.5149	3.9800	0.8831	0.8831*
20	3.8342	3.9514	0.9703	0.9703*
21	3.2556	3.7723	0.8630	0.8630*
22	3.3091	4.5210	0.7319	0.7319*
23	3.1869	3.9845	0.7998	0.7998*
24	3.3921	4.3585	0.7783	0.7782*
25	3.3487	3.9457	0.8487	0.8486*
26	3.3717	4.0804	0.8263	0.8263*
27	3.5268	4.3091	0.8184	0.8184*
28	3.1969	4.0974	0.7802	0.7802*
29	3.4953	3.8418	0.9098	0.9098*
30	3.6502	4.5043	0.8104	0.8103*
31	3.5861	3.9825	0.9005	0.9004*
32	3.5667	4.0459	0.8816	0.8815*
33	3.7031	4.0012	0.9255	0.9254*
34	3.3262	4.0025	0.8310	0.8310*
35	3.6096	4.3758	0.8249	0.8249*
36	3.5124	4.1171	0.8531	0.8531*
37	3.6057	4.0800	0.8838	0.8837*
38	3.3023	4.0363	0.8181	0.8181*
39	3.1986	4.0352	0.7927	0.7926*
40	3.4472	4.0686	0.8473	0.8472*
41	3.4169	4.2337	0.8071	0.8070*
42	3.4947	3.9074	0.8944	0.8943*
43	3.3732	4.1156	0.8196	0.8196*
44	3.6242	3.7780	0.9593	0.9592*
45	3.6352	3.8467	0.9450	0.9450*
46	3.4199	3.8624	0.8854	0.8854*
47	3.4977	4.0230	0.8694	0.8694*
48	3.5627	4.3215	0.8244	0.8244*
49	3.1991	3.8659	0.8275	0.8275*
50	3.3010	3.9410	0.8376	0.8376*
	σ^2 the variance			50/50

*There is no statistically significant difference between samples

Table 9 Comparison of results for each average

Trial	GA μ_1	OBEDA μ_2
1	0.0640	0.0456
2	0.0888	0.0661
3	0.0756	0.2347
4	0.0760	0.0663
5	0.0447	0.0158
6	0.0419	0.0708
7	0.1628	0.1021
8	0.1223	0.1054
9	0.0873	0.0662
10	0.0642	0.0493
11	0.1205	0.1056
12	0.1472	0.0874
13	0.1782	0.0458
14	0.0988	0.0504
15	0.0717	0.0466
16	0.0559	0.1500
17	0.0862	0.0787
18	0.1546	0.0372
19	0.0943	0.0000
20	0.0462	0.0874
21	0.0494	0.0268
22	0.0777	0.0068
23	0.0296	0.0285
24	0.1471	0.0803
25	0.0909	0.1041
26	0.0999	0.0285
27	0.0775	0.0726
28	0.1041	0.0573
29	0.0719	0.2358
30	0.1011	0.2190
31	0.0196	0.1393
32	0.0977	0.0406
33	0.1311	0.0746
34	0.1111	0.0455
35	0.1676	0.0533
36	0.0333	0.0717
37	0.0000	0.0410
38	0.0869	0.0413
39	0.0609	0.0236
40	0.0191	0.0709
41	0.1938	0.0778
42	0.0573	0.0460
43	0.0483	0.0706
44	0.1193	0.0493
45	0.1306	0.0833
46	0.1439	0.0285
47	0.0522	0.2561
48	0.0691	0.0935
49	0.1190	0.1171
50	0.1973	0.0456
μ the average	14/50	36/50

Table 10 Comparison of results for each average

Trial	GA μ_1	OBEDA μ_2
1	0.0179	0.0196
2	0.0161	0.0526
3	0.0461	0.0911
4	0.0361	0.0824
5	0.0460	0.0076
6	0.0678	0.0824
7	0.0150	0.0134
8	0.0377	0.0786
9	0.0505	0.0180
10	0.0222	0.0144
11	0.0952	0.0104
12	0.0797	0.0786
13	0.0082	0.0219
14	0.0204	0.0577
15	0.0155	0.0824
16	0.0122	0.0911
17	0.0417	0.0219
18	0.0399	0.0911
19	0.0795	0.0526
20	0.0192	0.0040
21	0.0247	0.0331
22	0.0154	0.0219
23	0.1039	0.0824
24	0.0358	0.0786
25	0.0704	0.0274
26	0.0154	0.0526
27	0.0240	0.0824
28	0.0813	0.0274
29	0.0057	0.0824
30	0.0166	0.0000
31	0.0763	0.0000
32	0.0155	0.0786
33	0.0141	0.0489
34	0.0522	0.0134
35	0.0296	0.0237
36	0.0000	0.0219
37	0.0000	0.0076
38	0.0314	0.0219
39	0.1191	0.0237
40	0.0258	0.0786
41	0.0284	0.0313
42	0.0053	0.0526
43	0.0069	0.0040
44	0.0192	0.0196
45	0.0606	0.0144
46	0.0481	0.0237
47	0.0837	0.0274
48	0.0184	0.0274
49	0.0177	0.0076
50	0.0204	0.0526
μ the average	27/50	23/50

Table 11 Comparison of results for each average

Trial	GA μ_1	OBEDA μ_2
1	0.1066	0.0967
2	0.0723	0.1565
3	0.1636	0.1176
4	0.1721	0.1444
5	0.2653	0.2061
6	0.0617	0.1573
7	0.1521	0.2029
8	0.1722	0.2986
9	0.0944	0.2346
10	0.1828	0.1884
11	0.2004	0.1046
12	0.0869	0.1902
13	0.0988	0.2069
14	0.1308	0.1893
15	0.1197	0.1335
16	0.0434	0.2642
17	0.2122	0.2146
18	0.1707	0.2210
19	0.2001	0.1624
20	0.1524	0.1779
21	0.0945	0.1479
22	0.2321	0.1077
23	0.1882	0.1764
24	0.0758	0.1346
25	0.0889	0.3957
26	0.1069	0.1208
27	0.2236	0.0000
28	0.1709	0.0977
29	0.2342	0.2440
30	0.1895	0.1594
31	0.1143	0.1078
32	0.1415	0.2170
33	0.1815	0.1427
34	0.1404	0.2798
35	0.0757	0.1105
36	0.0000	0.2093
37	0.2543	0.1876
38	0.0765	0.2402
39	0.1039	0.1616
40	0.1686	0.1232
41	0.0907	0.1257
42	0.1236	0.2434
43	0.1504	0.1193
44	0.2074	0.1881
45	0.2497	0.1846
46	0.0795	0.1325
47	0.1323	0.1745
48	0.1542	0.2273
49	0.0359	0.2126
50	0.1287	0.1952
μ the average	32/50	18/50

Table 12 Comparison of results for each average

Trial	GA μ_1	OBEDA μ_2
1	0.1134	0.1076
2	0.0888	0.0784
3	0.1112	0.0900
4	0.0649	0.1230
5	0.1107	0.1116
6	0.0952	0.0402
7	0.1097	0.1073
8	0.0011	0.0675
9	0.1626	0.1455
10	0.1694	0.1063
11	0.0339	0.0938
12	0.0379	0.0540
13	0.0902	0.0000
14	0.1702	0.1344
15	0.0875	0.1261
16	0.0929	0.0765
17	0.1396	0.0714
18	0.0919	0.1033
19	0.1581	0.1359
20	0.1478	0.0376
21	0.0466	0.1181
22	0.1775	0.0747
23	0.0752	0.0700
24	0.0652	0.0891
25	0.0690	0.1364
26	0.1238	0.0160
27	0.0961	0.1296
28	0.0699	0.1156
29	0.1453	0.1008
30	0.0906	0.1234
31	0.1071	0.0167
32	0.0742	0.0692
33	0.1620	0.1562
34	0.1441	0.1644
35	0.0289	0.0476
36	0.1262	0.1459
37	0.0375	0.0708
38	0.0910	0.0481
39	0.1228	0.1386
40	0.1634	0.0721
41	0.0617	0.1038
42	0.0956	0.1411
43	0.0325	0.1079
44	0.1217	0.0847
45	0.1320	0.0224
46	0.0000	0.1206
47	0.1453	0.1119
48	0.0005	0.0704
49	0.0977	0.0717
50	0.1092	0.0838
μ the average	23/50	27/50

Table 13 Comparison of results for each average with n=30, K=45

Trial	TS μ_1	OBCEDA μ_2
1	0.3775	0.2527
2	0.1690	0.2190
3	0.1817	0.1439
4	0.1090	0.2904
5	0.1587	0.1302
6	0.0617	0.1071
7	0.3481	0.2961
8	0.1000	0.3600
9	0.3422	0.2889
10	0.2105	0.2478
11	0.0224	0.0892
12	0.2304	0.1196
13	0.0988	0.2635
14	0.1258	0.2929
15	0.4547	0.1071
16	0.0959	0.2574
17	0.0573	0.3644
18	0.1259	0.2188
19	0.1372	0.1125
20	0.3131	0.2538
21	0.0889	0.2893
22	0.1413	0.2520
23	0.1887	0.3518
24	0.1109	0.2188
25	0.0000	0.2394
26	0.0880	0.2868
27	0.5087	0.0000
28	0.0991	0.2394
29	0.3521	0.2527
30	0.4920	0.1860
31	0.3997	0.2411
32	0.2893	0.2394
33	0.0654	0.2477
34	0.3532	0.1109
35	0.0931	0.2538
36	0.3486	0.2445
37	0.0205	0.1057
38	0.1424	0.3020
39	0.1073	0.2574
40	0.0833	0.2970
41	0.3349	0.2440
42	0.4223	0.2317
43	0.1449	0.2799
44	0.2786	0.1231
45	0.3247	0.0932
46	0.2450	0.2358
47	0.3779	0.0932
48	0.1369	0.2756
49	0.2867	0.2904
50	0.1045	0.2803
μ the average	28/50	22/50

Table 14 shows the variance obtained for each trial. We analyze whether there is a statistically significant difference between variances of both algorithms.

According to Table 14, there is no statistically significant difference between variances of both algorithms. The performance was the same in 50 of the 50 trials with α=0.01 of significance level. We consider that the stability of both algorithms is practically the same (100% of the time).

Table 15 details the average obtained for each trial when the total number of orders is 30 and the K capacity is 75 articles. We analyze the performance between averages of TS and OBCEDA algorithms.

In Table 15, there is no a significant difference between the averages of both algorithms. The performance of TS was almost equal to the OBCEDA performance.

Table 16 shows the variance obtained for each trial. We analyze whether there is a statistically significant difference between variances of both algorithms.

As we can see in Table 16, there is no statistically significant difference between variances of both algorithms. The performance was the same in 50 of the 50 trials with α= 0.01 of significance level. We consider that the stability of both algorithms is practically the same (100% of the time).

Table 17 details the average obtained for each trial when the total number of orders is 60 and the K capacity is 45 articles. We analyze the performance between averages of TS and OBCEDA algorithms.

Based on Table 17, there is no a significant difference between the averages of both algorithms. The performance of OBCEDA was equal to the TS performance.

Table 18 shows the variance obtained for each trial. We analyze whether there is a statistically significant difference between variances of both algorithms.

According to Table 18, there is no statistically significant difference between variances of both algorithms. The performance was the same in 50 of the 50 trials with α=0.01 of significance level. We consider that the stability of both algorithms is practically the same (100% of the time).

Table 19 details the average obtained for each trial when the total number of orders is 60 and the K capacity is 75 articles. We analyze the performance between averages of TS and OBCEDA algorithms.

In Table 19, there is no a significant difference between the averages of both algorithms. The performance of TS was equal to the OBCEDA performance.

Table 20 shows the variance obtained for each trial. We analyze whether there is a statistically significant difference between variances of both algorithms.

Based on Table 20, there is no statistically significant difference between variances of both algorithms. The performance was the same in 50 of the 50 trials with α=0.01 of significance level. We consider that the stability of both algorithms is practically the same (100 % of the time).

Table 14 Comparison of results for each variance with n=30, K=45

Trial	TS	OBCEDA		H_0: σ^2=σ^2 α=0.01
	σ^2	σ^2	σ^2/σ^2	F_c<0.545 or F_c>1.832
1	125.7944	124.5419	1.0101	1.0100*
2	106.7471	121.1925	0.8808	0.8808*
3	107.9158	113.7335	0.9488	0.9488*
4	101.2756	128.2918	0.7894	0.7894*
5	105.8111	112.3647	0.9417	0.9416*
6	96.9550	110.0653	0.8809	0.8808*
7	123.1124	128.8606	0.9554	0.9553*
8	100.4550	135.2071	0.7430	0.7429*
9	122.5681	128.1388	0.9565	0.9565*
10	110.5371	124.0600	0.8910	0.8909*
11	93.3743	108.2932	0.8622	0.8622*
12	112.3611	111.3124	1.0094	1.0094*
13	100.3437	125.6205	0.7988	0.7987*
14	102.8095	128.5400	0.7998	0.7998*
15	132.8382	110.0653	1.2069	1.2069*
16	100.0840	125.0076	0.8006	0.8006*
17	96.5545	135.6494	0.7118	0.7117*
18	102.8163	121.1731	0.8485	0.8485*
19	103.8465	110.5967	0.9390	0.9389*
20	119.9150	124.6490	0.9620	0.9620*
21	99.4361	128.1765	0.7758	0.7757*
22	104.2181	124.4740	0.8373	0.8372*
23	108.5453	134.4024	0.8076	0.8076*
24	101.4463	121.1731	0.8372	0.8372*
25	91.3205	123.2169	0.7411	0.7411*
26	99.3633	127.9321	0.7767	0.7766*
27	137.7682	99.4168	1.3858	1.3857*
28	100.3711	123.2169	0.8146	0.8145*
29	123.4721	124.5419	0.9914	0.9914*
30	136.2524	117.9047	1.1556	1.1556*
31	127.8229	123.3900	1.0359	1.0359*
32	117.7426	123.2169	0.9556	0.9555*
33	97.2861	124.0463	0.7843	0.7842*
34	123.5672	110.4488	1.1188	1.1187*
35	99.8206	124.6490	0.8008	0.8008*
36	123.1482	123.7344	0.9953	0.9952*
37	93.1874	109.9347	0.8477	0.8476*
38	104.3179	129.4371	0.8059	0.8059*
39	101.1226	125.0076	0.8089	0.8089*
40	98.9345	128.9500	0.7672	0.7672*
41	121.9025	123.6800	0.9856	0.9856*
42	129.8825	122.4588	1.0606	1.0606*
43	104.5475	127.2521	0.8216	0.8215*
44	116.7600	111.6600	1.0457	1.0456*
45	120.9731	108.6876	1.1130	1.1130*
46	113.6895	122.8600	0.9254	0.9253*
47	125.8306	108.6876	1.1577	1.1577*
48	103.8160	126.8163	0.8186	0.8186*
49	117.4956	128.2918	0.9158	0.9158*
50	100.8590	127.2871	0.7924	0.7923*
σ^2 the variance				50/50

*There is no statistically significant difference between samples

Table 15 Comparison of results for each average with n=30, K=75

Trial	TS μ_1	OBCEDA μ_2
1	0.2052	0.1447
2	0.1135	0.0000
3	0.1469	0.1292
4	0.1964	0.0092
5	0.2017	0.1817
6	0.0209	0.1485
7	0.1331	0.1486
8	0.1203	0.1335
9	0.1326	0.2004
10	0.1702	0.0056
11	0.0382	0.1446
12	0.1773	0.0000
13	0.1539	0.1664
14	0.2088	0.1664
15	0.1368	0.0077
16	0.0123	0.1770
17	0.0057	0.0117
18	0.0090	0.0348
19	0.0114	0.1339
20	0.0168	0.1447
21	0.1573	0.1664
22	0.0132	0.0056
23	0.0132	0.0348
24	0.1142	0.0040
25	0.1585	0.1403
26	0.0762	0.0307
27	0.1203	0.1379
28	0.1295	0.0092
29	0.2000	0.1844
30	0.1288	0.0686
31	0.0447	0.0076
32	0.2127	0.0722
33	0.1756	0.1403
34	0.0000	0.1485
35	0.1615	0.1296
36	0.1637	0.0348
37	0.0689	0.1403
38	0.0759	0.0686
39	0.0701	0.1292
40	0.0745	0.1446
41	0.0767	0.1770
42	0.1305	0.0040
43	0.0147	0.1485
44	0.0037	0.1568
45	0.1574	0.1485
46	0.1820	0.1814
47	0.1863	0.0041
48	0.1677	0.0000
49	0.0832	0.1770
50	0.1701	0.0056
μ the average	23/50	26/50

Table 16 Comparison of results for each variance with n=30, K=75

Trial	TS σ^2	OBCEDA σ^2	σ^2/σ^2	H_0: σ^2=σ^2 α=0.01 F_c<0.545 or F_c>1.832
1	167.6475	160.6033	1.0439	1.0438*
2	154.8867	140.2985	1.1040	1.1039*
3	159.5242	158.4225	1.0070	1.0069*
4	166.4242	141.5938	1.1754	1.1753*
5	167.1617	165.7933	1.0083	1.0082*
6	142.0100	161.1325	0.8813	0.8813*
7	157.6092	161.1458	0.9781	0.9780*
8	155.8317	159.0333	0.9799	0.9798*
9	157.5525	168.4067	0.9355	0.9355*
10	162.7692	141.0815	1.1537	1.1537*
11	144.4162	160.5908	0.8993	0.8992*
12	163.7633	140.2985	1.1672	1.1672*
13	160.5067	163.6433	0.9808	0.9808*
14	168.1433	163.6433	1.0275	1.0274*
15	158.1292	141.3762	1.1185	1.1184*
16	140.8108	165.1417	0.8527	0.8526*
17	139.8931	141.9400	0.9856	0.9855*
18	140.3515	145.1800	0.9667	0.9667*
19	140.6808	159.0908	0.8843	0.8842*
20	141.4408	160.6033	0.8807	0.8806*
21	160.9750	163.6433	0.9837	0.9836*
22	140.9315	141.0815	0.9989	0.9989*
23	140.9315	145.1800	0.9707	0.9707*
24	154.9875	140.8638	1.1003	1.1002*
25	161.1508	159.9783	1.0073	1.0073*
26	149.6954	144.6146	1.0351	1.0351*
27	155.8308	159.6458	0.9761	0.9761*
28	157.1100	141.5938	1.1096	1.1095*
29	166.9258	166.1675	1.0046	1.0045*
30	157.0233	149.9223	1.0474	1.0473*
31	145.3200	141.3746	1.0279	1.0279*
32	168.6950	150.4362	1.1214	1.1213*
33	163.5275	159.9783	1.0222	1.0221*
34	139.1008	161.1325	0.8633	0.8632*
35	161.5608	158.4783	1.0195	1.0194*
36	161.8700	145.1800	1.1150	1.1149*
37	148.6815	159.9783	0.9294	0.9293*
38	149.6554	149.9223	0.9982	0.9982*
39	148.8477	158.4225	0.9396	0.9395*
40	149.4577	160.5908	0.9307	0.9306*
41	149.7723	165.1417	0.9069	0.9069*
42	157.2467	140.8638	1.1163	1.1163*
43	141.1346	161.1325	0.8759	0.8758*
44	139.6231	162.2983	0.8603	0.8602*
45	161.0042	161.1325	0.9992	0.9992*
46	164.4075	165.7542	0.9919	0.9918*
47	165.0233	140.8754	1.1714	1.1714*
48	162.4342	140.2985	1.1578	1.1577*
49	150.6669	165.1417	0.9123	0.9123*
50	162.7550	141.0815	1.1536	1.1536*
	σ^2 the variance			50/50

*There is no statistically significant difference between samples

Table 17 Comparison of results for each average with n=60, K=45

Trial	TS μ_1	OBCEDA μ_2
1	0.1394	0.1467
2	0.0502	0.0000
3	0.1041	0.1554
4	0.1210	0.1084
5	0.1408	0.0875
6	0.0000	0.1454
7	0.2438	0.1804
8	0.1473	0.1647
9	0.1413	0.1128
10	0.1351	0.0202
11	0.1598	0.0838
12	0.0920	0.1217
13	0.1065	0.1255
14	0.2077	0.1772
15	0.0631	0.1232
16	0.1627	0.0817
17	0.1274	0.2565
18	0.1347	0.1058
19	0.0224	0.0539
20	0.0758	0.0059
21	0.0831	0.1173
22	0.0485	0.0341
23	0.1015	0.1589
24	0.0876	0.1540
25	0.0498	0.2203
26	0.0823	0.1478
27	0.1440	0.2163
28	0.0681	0.0949
29	0.0620	0.0247
30	0.1467	0.1750
31	0.1059	0.0763
32	0.0934	0.2096
33	0.1025	0.0523
34	0.1237	0.1440
35	0.1552	0.1260
36	0.0835	0.2310
37	0.0523	0.1358
38	0.0475	0.0645
39	0.0873	0.1109
40	0.1746	0.0439
41	0.1804	0.1954
42	0.0361	0.0957
43	0.1100	0.1063
44	0.1428	0.0524
45	0.0786	0.1617
46	0.2014	0.0610
47	0.0536	0.1448
48	0.1931	0.1817
49	0.0957	0.1860
50	0.1512	0.1285
μ the average	28/50	22/50

Table 18 Comparison of results for each variance with n=60, K=45

Trial	TS σ^2	OBCEDA σ^2	σ^2/σ^2	H_0: $\sigma^2=\sigma^2$ α=0.01 F_c<0.545 or F_c>1.832
1	219.7588	221.3544	0.9928	0.9927*
2	202.5706	193.0393	1.0494	1.0493*
3	212.9659	223.0247	0.9549	0.9548*
4	216.2224	213.9613	1.0106	1.0105*
5	220.0441	209.9381	1.0481	1.0481*
6	192.8833	221.1100	0.8723	0.8723*
7	239.9144	227.8638	1.0529	1.0528*
8	221.2881	224.8240	0.9843	0.9842*
9	220.1321	214.8180	1.0247	1.0247*
10	218.9300	196.9327	1.1117	1.1116*
11	223.7067	209.2047	1.0693	1.0693*
12	210.6156	216.5413	0.9726	0.9726*
13	213.4180	217.2593	0.9823	0.9823*
14	232.9413	227.2387	1.0251	1.0250*
15	205.0565	216.8181	0.9458	0.9457*
16	224.2663	208.8065	1.0740	1.0740*
17	217.4535	242.5580	0.8965	0.8965*
18	218.8581	213.4687	1.0252	1.0252*
19	197.1947	203.4464	0.9693	0.9692*
20	207.5007	194.1700	1.0687	1.0686*
21	208.9041	215.6819	0.9686	0.9685*
22	202.2367	199.6353	1.0130	1.0130*
23	212.4500	223.7138	0.9497	0.9496*
24	209.7656	222.7594	0.9417	0.9416*
25	202.4812	235.5588	0.8596	0.8595*
26	208.7512	221.5687	0.9422	0.9421*
27	220.6520	234.7863	0.9398	0.9397*
28	206.0112	211.3600	0.9747	0.9746*
29	204.8400	197.7967	1.0356	1.0356*
30	221.1771	226.8257	0.9751	0.9750*
31	213.3100	207.7587	1.0267	1.0267*
32	210.9018	233.4947	0.9032	0.9032*
33	212.6527	203.1447	1.0468	1.0468*
34	216.7294	220.8292	0.9814	0.9814*
35	222.8157	217.3556	1.0251	1.0251*
36	208.9959	237.6294	0.8795	0.8795*
37	202.9682	219.2544	0.9257	0.9257*
38	202.0518	205.5053	0.9832	0.9831*
39	209.7100	214.4393	0.9779	0.9779*
40	226.5613	201.5120	1.1243	1.1243*
41	227.6687	230.7560	0.9866	0.9866*
42	199.8450	211.5143	0.9448	0.9448*
43	214.0856	213.5573	1.0025	1.0024*
44	220.4180	203.1520	1.0850	1.0849*
45	208.0464	224.2629	0.9277	0.9276*
46	231.7242	204.8180	1.1314	1.1313*
47	203.2059	220.9913	0.9195	0.9195*
48	230.1318	228.1219	1.0088	1.0088*
49	211.3350	228.9393	0.9231	0.9231*
50	222.0524	217.8429	1.0193	1.0193*
	σ^2 the variance			50/50

*There is no statistically significant difference between samples

Table 19 Comparison of results for each average with n=60, K=75

Trial	TS μ_1	OBCEDA μ_2
1	0.0751	0.0905
2	0.2030	0.1128
3	0.2006	0.0557
4	0.1997	0.1477
5	0.2218	0.1054
6	0.1429	0.1699
7	0.0575	0.1655
8	0.1525	0.1555
9	0.2020	0.1755
10	0.0868	0.1374
11	0.2182	0.1752
12	0.1716	0.1265
13	0.0987	0.1360
14	0.1180	0.1090
15	0.1764	0.1779
16	0.1200	0.1289
17	0.1716	0.1534
18	0.0408	0.0885
19	0.2313	0.0351
20	0.1906	0.0321
21	0.1495	0.1445
22	0.0000	0.0000
23	0.2234	0.1912
24	0.0869	0.1561
25	0.2483	0.1496
26	0.0899	0.1062
27	0.1000	0.1726
28	0.1497	0.2252
29	0.1370	0.1260
30	0.0780	0.1094
31	0.0644	0.1161
32	0.2077	0.1134
33	0.1035	0.1189
34	0.1045	0.1815
35	0.1551	0.1670
36	0.0544	0.1496
37	0.0538	0.1016
38	0.2424	0.1764
39	0.1603	0.1947
40	0.2154	0.0122
41	0.1521	0.0328
42	0.2089	0.1538
43	0.1485	0.0942
44	0.1003	0.1566
45	0.1984	0.1740
46	0.1318	0.1063
47	0.1467	0.1204
48	0.2020	0.0214
49	0.0667	0.0280
50	0.1838	0.0353
μ the average	22/50	28/50

Table 20 Comparison of results for each variance with n=60, K=75

Trial	TS σ^2	OBCEDA σ^2	σ^2/σ^2	H_0: σ^2=σ^2 α=0.01 F_c<0.545 or F_c>1.832
1	219.7550	234.7535	0.9361	0.9361*
2	245.9144	239.5519	1.0266	1.0265*
3	245.4080	227.2471	1.0799	1.0799*
4	245.2313	247.0506	0.9926	0.9926*
5	249.7476	237.9367	1.0496	1.0496*
6	233.6125	251.8288	0.9277	0.9276*
7	216.1725	250.8807	0.8617	0.8616*
8	235.5906	248.7325	0.9472	0.9471*
9	245.7139	253.0354	0.9711	0.9710*
10	222.1483	244.8375	0.9073	0.9073*
11	249.0075	252.9729	0.9843	0.9843*
12	239.4813	242.4987	0.9876	0.9875*
13	224.5888	244.5312	0.9184	0.9184*
14	228.5294	238.7175	0.9573	0.9573*
15	240.4706	253.5517	0.9484	0.9484*
16	228.9279	243.0071	0.9421	0.9420*
17	239.4840	248.2947	0.9645	0.9645*
18	212.7393	234.3031	0.9080	0.9079*
19	251.6788	222.8233	1.1295	1.1294*
20	243.3707	222.1713	1.0954	1.0954*
21	234.9580	246.3744	0.9537	0.9536*
22	204.4093	215.2647	0.9496	0.9495*
23	250.0800	256.4115	0.9753	0.9753*
24	222.1719	248.8706	0.8927	0.8927*
25	255.1706	247.4550	1.0312	1.0311*
26	222.7888	238.1247	0.9356	0.9355*
27	224.8587	252.4229	0.8908	0.8908*
28	235.0143	263.7438	0.8911	0.8910*
29	232.4186	242.3867	0.9589	0.9588*
30	220.3600	238.8241	0.9227	0.9226*
31	217.5811	240.2627	0.9056	0.9055*
32	246.8707	239.6736	1.0300	1.0300*
33	225.5575	240.8625	0.9365	0.9364*
34	225.7744	254.3293	0.8877	0.8877*
35	236.1064	251.2061	0.9399	0.9398*
36	215.5456	247.4618	0.8710	0.8710*
37	215.4050	237.1412	0.9083	0.9083*
38	253.9553	253.2400	1.0028	1.0028*
39	237.1731	257.1827	0.9222	0.9221*
40	248.4383	217.8756	1.1403	1.1402*
41	235.5013	222.3267	1.0593	1.0592*
42	247.1053	248.3556	0.9950	0.9949*
43	234.7694	235.5318	0.9968	0.9967*
44	224.9179	248.9771	0.9034	0.9033*
45	244.9567	252.7206	0.9693	0.9692*
46	231.3594	238.1365	0.9715	0.9715*
47	234.4006	241.1676	0.9719	0.9719*
48	245.6943	219.8744	1.1174	1.1174*
49	218.0353	221.2920	0.9853	0.9852*
50	241.9773	222.8600	1.0858	1.0857*
	σ^2 the variance			50/50

*There is no statistically significant difference between samples

6 Discussion

On robustness, the algorithms utilized in this research are not able to handle invalid or unexpected inputs. These have not been encoded for specific users. This topic has not been considered in this research because it is not the main objective. However, the OBCEDA algorithm proposed can be modified in order to get a useful module for specific users in industry.

On convergence and diversity, the algorithms used in this research keep diversity to incorporate specific operators such as the mutation operator in GA and TS in the evolutionary progress. Those operators are useful on permutation-based problems. In addition, any trial returns the optimal vector until the difference between the average fitness of the trial and the best is <5 %.

On stopping criteria, we changed the stopping criterion used previously where a number of generations had to be reached because the disadvantage is that the number of function evaluations necessary for convergence is unknown a priori. Therefore, we used the difference between the average fitness of the trial and the best as a stopping criterion.

On computational time and cost, these were not considered in this research because the algorithm proposed is currently in the prototype phase. Future research work would consider a module for users, and it should include computational time and cost aspects.

On advantages and disadvantages of the proposed method, we can consider that it takes into account the relationship or interactions among variables of the problem as an advantage. For each generation, we know the probability that batch j was used for the i customer order. However, the probabilistic model used could be basic; it may be a disadvantage if we need to model higher interactions.

On global optimum, note that for this problem, there are no known effective precise techniques and a comparison with an optimum solution is not possible. It is a characteristic of the online optimization topic.

On computational complexity, the online order-batching problem is as *NP* hard as the offline problem type based on Gademann and van de Velde [6], if the number of orders per batch is greater than two.

On feasibility and flexibility, on the one hand, all the algorithms used in this research were able to produce feasible solutions according to different constraints detailed in the problem statement section of this paper. It was not necessary to repair the solutions as other algorithms used for permutation-based problems. The proposed method considers the previous results in order to avoid unfeasible solutions. On the other hand, the algorithms utilized in this research are not flexible to handle new and unexpected customer orders. The proposed method is currently in the prototype phase for users.

On efficiency and effectiveness, in this research, the amount of resources used by all the algorithms was not considered, e.g., the requirement for high speed or for minimum memory usage were of no interest.

On reliability and user friendliness, all the algorithms were tested in order to get reliability according to the online order-batching characteristics. However, these are not industry-ready yet.

On exploitative and exploration capability, all the algorithms used in this research keep exploitative and exploration capability to incorporate specific operators such as cross and mutation operator in GA and TS in the evolutionary progress.

7 Conclusions

Based on the experimental results shown, we confirmed that an appropriate modeling of the most important variables that affect the performance of the picking process should be considered in the proposed solution. We reach the conclusion that the order-picking performance can be improved if we take into account the relationship or interaction among variables of the problem. Using a continuous EDA was not necessary to make any modifications in the sampling process in the processing sequence of customer orders on the batches, as is generally required by other algorithms. It allowed for better trust in the data against the GA. The OBCEDA considers the previous assignments like an updating mechanism to detect the relationship between customer orders and was able to tackle the individual's inadequate representation related to combinatorial problems used in GAs. We conclude that the OBCEDA can be an efficient mechanism to handle different order-picking conditions where there are diverse variable interactions such as the online batching problem.

Although the difference between the TS and OBCEDA performances was not statistically significant, future research work will use higher probabilistic models in order to model higher interactions or relations between variables of the order-picking performance. Finally, this research contributes using an EDA as an optimization method for any order-picking process.

Acknowledgments We would like to express our gratitude to Elizabeth O'Shaughnessy for reviewing the manuscript.

References

1. Bean J, Norman B (1993) Random keys for job shop scheduling problem. The University of Michigan, Ann Arbor
2. Chen S-H, Chen M-C, Chang P-C, Zhang Q, Chen Y-M (2010) Guidelines for developing effective estimation of distribution algorithms in solving single machine scheduling problems. Expert Syst Appl 37:6441–6451

3. Chew E, Tang L (1999) Travel time analysis for general item location assignment in a rectangular warehouse. Eur J Oper Res 112(3):582–597
4. De Bonet J, Isbell C, Viola P (1997) MIMIC: finding optima by estimation probability densities. Advances in Neural Information Processing Systems 9
5. de Koster M, Le-Duc T, Roodbergen K (2007) Design and control of warehouse order picking: a literature review. Eur J Oper Res 182(2): 481–501
6. Gademann N, van de Velde S (2005) Order batching to minimize total travel time in a parallel-aisle warehouse. IIE Trans 37(1):63–75
7. Ganesan T, Elamvazuthi I, Shaari KK, Vasant P (2013) Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production. Appl Energy 103:368–374
8. Goldberg D, Korb B, Deb K (1989) Messy genetic algorithms: motivation, analysis and first results. Compl Syst 3:493–530
9. Greenwood A, Vanguri S, Eksioglu B, Jain P, Hill T, Miller J (2005) Simulation optimization decision support system for ship panel shop operations. In Kuhl ME, Steiger NM, Armstrong F, Joines J (eds) Proceedings of the 2005 Winter Simulation Conference, (pp. 2078–2086)
10. G Harik (1997) Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms. PhD. Thesis
11. Henn S (2009) Metaheuristics for the order batching problem in manual order picking systems. Ottovon-Guericke-University Magdeburg. Faculty of Economics and Management
12. Henn, S. (2010). Algorithms for on-line order batching in an order-picking warehouse. Proceedings of the 3rd International Conference on Information Systems, Logistics and Supply Chain ILS 2010. Casablanca: Business Process Consulting
13. Kamin N (1998) On-line optimization of order picking in an automated warehouse. Shaker, Aachen
14. H Kargupta (1996) The gene expression messy genetic algorithm. Proceedings of the 1996 I.E. International Conference on Evolutionary Computation (pp. 631–636)
15. Larrañaga P, Exteberria R, Lozano J, Peña J (2000) Optimization in continuous domains by learning and simulation of Gaussian networks. Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program
16. Larrañaga P, Lozano J (2002) Estimation of distribution algorithms: a new tool for evolutionary computation. Kluwer Academic Publishers, Boston/Dordrecht/London
17. Le Dinh L, Vo Ngoc D, Vasant P (2013) Artificial bee colony algorithm for solving optimal power flow problem. Sci World J 2013
18. Le-Duc T, de Koster R (2007) Travel time estimation and order batching in a 2-block warehouse. Eur J Oper Res 176(1):374–388
19. Liu H, Gao L, Pan Q (2011) A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem. Expert Syst Appl 38:4348–4360
20. Mühlenbein H, Paaß G (1996) From recombination of genes to the estimation of distributions: I. binary parameters. In: Voigt H, Ebeling W, Rechenberg I, Schwefel H (eds) Parallel problem solving from nature PPSN IV. Springer, Berlin, pp 178–187
21. Öncan T (2013) A Genetic Algorithm for the order batching problem in low-level picker-to-part warehouse systems. Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013. Hong Kong
22. Pan Q-K, Ruiz R (2012) An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. Omega 40:166–180
23. Pérez R, Jöns S, Hernández A, Ochoa C (2014) Simulation optimization for a flexible jobshop scheduling problem using an estimation of distribution algorithm. Int J Adv Manuf Technol 73(1–4):3–21
24. Rudolph G (1991) Global optimization by means of distributed evolution strategies. Parallel problem solving from nature PPSN I. Lectures Notes in Computer Science
25. van Nieuwenhuyse I, de Koster R (2009) Evaluating order throughput time in 2-block warehouses with time window batching. Int J Prod Econ 121:654–664
26. Wang L, Wang S, Xu Y, Zhou G, Liu M (2012) A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. Comput Ind Eng 62:917–926
27. Wäscher G (2004) Order picking: a survey of planning problems and methods. In: Dyckhoff H, Lackes R, Reese J (eds) Supply Chain Management and Reverse Logistics. Springer, Berlin, pp 323–347
28. Whitley D, Starweather T, Shaner D (1990) The traveling salesman and sequence scheduling: quality solutions using genetic edge recombination. In: Davis L (ed) Handbook of genetic algorithms. Van Nostrand Reinhold, New York, pp 350–372
29. Yu M, de Koster R (2009) The impact of order batching and picking area zoning on order picking system performance. Eur J Oper Res 198(2):480–490