

# An Estimation of Distribution Algorithm based Approach for the Order Batching Problem

Ricardo Pérez-Rodríguez, Arturo Hernández-Aguirre

CIMAT A.C., Center for Research in Mathematics,  
Guanajuato, Guanajuato, Mexico

ricardo.perez@cimat.mx, artha@cimat.mx

**Abstract.** Up to 50 per cent of all warehouse operations costs can be attributed to order picking [1]. It is an aspect critical to each supply chain and the planning and control of warehouse processes. According to experience from logistics environments, a reduction of picking times is an integral part of the delivery lead-time and may result in an improvement of the customer service provided by a warehouse. Order batching consists of combining customer orders into picking orders to minimize the picking time. Different evolutionary algorithms have been proposed to build better batches for the order picking. These algorithms try to preserve batches or sequences of batches that appear frequently in all members of the population in order to keep track and inherit these characteristics exhibited by the parents to the next generation. However, the traditional evolutionary operators used in current research may lose the characteristics mentioned. To address this situation, an estimation of distribution algorithm - based approach is proposed for the order batching problem in an order-picking warehouse. A probability model is built to describe the characteristics exhibited by the parents as a distribution of the solution space. In a series of extensive numerical experiments, an acceptable performance using the algorithm mentioned is shown against different evolutionary algorithms known in the literature.

**Keywords:** estimation of distribution algorithm, warehouse management, order picking, order batching.

## 1 Introduction

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 [2]. Order picking systems involving operators are very common in industry. Such manual order picking systems can be divided into two categories [3]: parts-to-picker, where automated storage and retrieval systems deliver the articles to a transfer site where stationary pickers are located; picker-to-parts, where pickers normally walk or ride (e.g. on an automated guided vehicle) through the warehouse and collect articles. If the number of arriving orders is too large for processing each cus-

customer order separately in an appropriate amount of time, customer orders must be combined into batches. The process of grouping a set of customer orders into picking orders is referred to as order batching. It means that different customer orders can be simultaneously released for picking. 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.

On the tours through the warehouse, order pickers are guided by pick lists. A pick list comprises a set of order lines, each one identifying a particular article, the quantity of the article requested and the respective storage location. The order lines are already sorted into the sequence according to which the order picker is meant to collect the items.

The order batching problem is a combinatorial issue. It features some similarities with the capacitated vehicle routing problem, however differs from that with respect to the customer order integrity condition, i.e. all items of a customer order must be picked on the same tour. Thus, traditional solutions approaches to the capacitated vehicle routing problem cannot be applied directly to the order batching problem [4].

The order batching has been studied and solved through different evolutionary algorithms such as Genetic Algorithms (GA), Memetic Algorithms (MA), and Tabu Search (TS). A main characteristic of these algorithms is to preserve batches or sequences of batches that appear frequently in all members of the population in order to keep track and inherit these characteristics exhibited by the parents to the next generation. However, the traditional evolutionary operators used in current research may lose the characteristics mentioned. An example can be found in [5].

Another approach consists of using Estimation of Distribution Algorithms (EDA), introduced by [6]. 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 to describe the characteristics exhibited by the parents 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. 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 [7], the 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, 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 under-

stand 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.

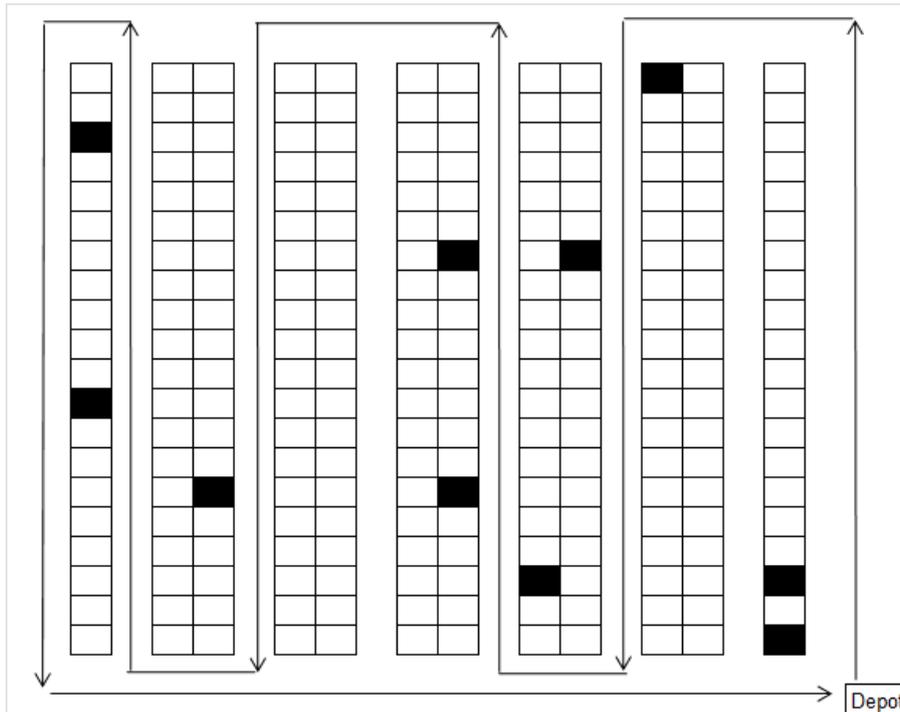


Fig. 1. Example of “S-Shape” route.

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 research a single order picker is considered, i.e., all batches must be processed one after another. We focus on minimizing the total completion time of all customer 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

$v_{travel}$  : travel velocity, i.e. number of length units the picker can cover in the warehouse per time unit.

$v_{pick}$  : pick velocity, i.e. number of items the picker can search and pick per unit time.  
 $n$  : 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  $i \in \{1, \dots, n\}$ ,  
 $k_i$  : the number of articles of order  $i$  for all  $i \in \{1, \dots, n\}$ ,  
 $K$  : 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  $j \in \{1, \dots, m\}$   
 $x_{ji} = \{1 \text{ if order } i \text{ is assigned to batch } j, 0 \text{ otherwise}\}$   
 $\mathbf{x}_j = (x_{j1}, \dots, x_{jn})^T$  is a vector where some customer orders are associated to the batch  $j$ , for all  $j \in \{1, \dots, m\}$

An optimization model can be formulated as follows:

The total length of the picking tour for a particular batch involving the S-shape routing method is determined by  $d$  as a function for determining the length of tour given a batch. The problem can be modeled as follows:

$$\text{Min } \max_{j \in \{1, \dots, m\}} \left\{ S_j + \frac{d(\mathbf{x}_j)}{v_{travel}} + \frac{\sum_{i=1}^n k_i x_{ji}}{v_{pick}} \right\} \quad (1)$$

$$\sum_{j=1}^m x_{ji} = 1, \forall i \in \{1, \dots, n\} \quad (2)$$

$$\sum_{i=1}^n k_i x_{ji} \leq K, \forall j \in \{1, \dots, m\} \quad (3)$$

$$S_j \geq \max_{i \in \{1, \dots, n\}} \{a_i \cdot x_{ji}\}, \forall j \in \{1, \dots, m\} \quad (4)$$

$$S_j \geq S_{j-1} + \frac{d(\mathbf{x}_{j-1})}{v_{travel}} + \frac{\sum_{i=1}^n k_i x_{j-1i}}{v_{pick}}, \forall j \in \{2, \dots, m\} \quad (5)$$

$$S_j \geq 0, \forall j \in \{1, \dots, m\} \quad (6)$$

$$x_{ji} \in \{0, 1\} \quad (7)$$

In the objective function (1) the expression after  $S_j$  represents the service time of a batch. This sum is composed of the time the picker needs to travel and the time he/she needs to pick the items. Equations (2) ensure the assignment of each order to exactly one batch. Inequalities (3) guarantee that the capacity of the picking device is not violated. The conditions (4) indicate that a batch is started after all customer orders assigned to that batch are known. From (5) follows that a batch is started after the previous one is completed. Finally (6) and (7) indicate that the start times are non-negative and the variables are binary.

### 3 Literature Review

A discussion about the most current research on the order batching process is outlined below.

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

[9] 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 FCFS rule (First Come First Serve). 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 [10]. 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 [11].

[12] 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.

[7] describes an on-line order batching problem in a walk-and-pick ware-house 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 off-line order batching in order to deal with the on-line situation.

[5] introduces a GA for the order batching problem considering tra-versal 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. We explained in the introduction section above that these algorithms try to preserve batches or sequences of batches that appear frequently in all members of the population in order to keep track and inherit these characteristics exhibited by the parents to the next generation. However, the traditional evolutionary operators used in current research may lose the characteristics mentioned.

## **4 EDA for the Order Batching Problem**

Instead of using crossover and mutation operators of GA, the EDA produces offspring implicitly by using a probability model according to the knowledge of the obtained solution space during the search process. At the beginning of each generation, a subset of the population is determined and the obtained best solution is updated. Then,

the probability model is updated based on the sub-set mentioned and the offspring is generated by sampling the probability model. The EDA repeats the above process until a stopping condition is met.

#### 4.1 Solution Representation

A solution to the order 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 an integer value. This representation is shown in Figure 2 with 10 orders. Observe that the first batch consists of orders {5; 7; 9}; the second batch includes orders {1; 3; 4; 6} and the third batch incorporates order {2; 8; 10}.

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

Fig. 2. Solution vector.

#### 4.2 Probability Model Proposed

The probability model is designed as a probability matrix, i.e., batch probability matrix  $P$ . The element  $p_{ji}$  of the probability matrix represents the probability that the batch  $j$  be used for the  $i$  customer order, where  $j = \{1, \dots, m\}$  and  $i = \{1, \dots, n\}$ . The value of  $p_{ji}$  indicates the opportunity of a customer order on a certain batch. To build the probabilistic model or probability matrix  $P$  we need to compute the number of times that batch  $j$  appears for the customer  $i$  in the selected population. The result obtained previously is divided by the number of individuals considered in the selected population. Via sampling according to the probability matrix new promising individuals may be generated. 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  $p_{j1}, p_{j2}, \dots, p_{jn}$  of probability matrix will be set as zero. This updating mechanism considers the previous assignments.

The probability model describes the characteristics exhibited by the parents as a distribution of the solution space. Therefore, the algorithm proposed does not need to hold members of the population to keep track and inherit the characteristics mentioned in each generation. The characteristics persist for all members of the population in the evolutionary progress by the probability model.

The major procedure of the EDA 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 the probabilistic model  $P$  which describes the distribution characteristics of  $D$ .

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

Step 5. A binary tournament is done to 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  $g < GN$ , where  $GN$  is the maximum number of generations return to step 2. Otherwise, output the best solution in  $S(g)$ .

To describe the distribution of the solution space, the EDA tries to determine the relationship or interaction among variables of the problem as primary objective. The traditional evolutionary operators such as crossover and mutation are substitute by the probability model which is built with information on relationships and interactions among variables. The probability model on step 3 plays an important role in the EDA. Therefore, it should be well designed according to the characteristics of the considered problem. Step 4 makes another main difference between the EDA and other evolutionary algorithms. Sampling offspring from the probability matrix instead of using classical operators is the critical feature.

## **5 Results and Comparison**

Three evolutionary algorithms have been proposed as benchmark for comparison with the EDA scheme. A GA based on [13] works with tournament selection. The “edge recombination operator” is used as a cross operator and a mutation operator changes batches among different positions. A TS based on [1] works with the operator “swap move”, meaning the interchanging of two customer orders from different batches. A tabu list is built on ten neighbors. A MA base on [14] works with the same operator for the local search phase.

To account for the stochastic nature of the order-picking warehouse, we ran 50 trials for all algorithms. We established a workload that contains different customer orders and differing types of articles required in a workday, replicating the order-picking process conditions to evaluate and find the minimum total completion time. 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 [1]. In this layout, the picking area consists of 900 articles and storage locations. The storage locations are arranged into 10 aisles. We further assume that an order picker walks through 10 storage locations in 30 seconds and they need 10 seconds to search and collect an article from a storage location. For the capacity of the picking device  $K$ , we assume three different values, namely 45, 75 and 100 articles. For the total number of orders  $n$  we consider 30, 60 and 90. 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 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. In our numerical experiments we choose  $\lambda$  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  $\lambda$ : for  $n = 30$ :  $\lambda = 0.0625$ , for  $n = 60$ :  $\lambda = 0.125$ , for  $n = 90$ :  $\lambda = 0.1875$ .

This dynamic case is characterized as an order picking environment in which the customer orders arrive stochastically over time, and only when an order has arrived the information becomes available of which articles and respective quantities the order is composed of. Under such conditions time window batching is prevalent. We set a fixed time window batching, i.e., all customer orders arriving during a particular time interval are assigned to batches by means of the probability model described above. The fixed time window batching for this tested scenario is set hourly. The unprocessed customer orders are input for building the initial population and the probabilistic model proposed in periods of one hour. The solution representation, i.e., the solution vector gets its dimension on the number of unprocessed customer orders confirmed previously. Therefore the batching decision is based on the known orders only. Once the batches are built by the sampling process, these are released one by one because we only have one picker to collect the parts in this tested scenario.

Table 1 details the minimum total completion time obtained for each case among all the trials.

**Table 1.** Comparison of results.

Capacity	Orders	Algorithms			
		GA	MA	TS	EDA
45	30	659	659	659	665
	60	840	784	834	<b>784</b>
	90	1926	1738	1818	<b>1716</b>
75	30	694	694	694	694
	60	1422	1329	1391	<b>1328</b>
	90	2294	2109	2231	<b>2096</b>
100	30	711	711	712	712
	60	1316	1214	1259	<b>1214</b>
	90	2011	1892	1981	<b>1891</b>

## 6 Discussion and Conclusions

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 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 prob-

lems. 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 on-line optimization topic. On computational complexity, the on-line order batching problem is as NP-hard. On feasibility and flexibility, on 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 on-line 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.

Based on the experimental results shown, we deduce that the EDA can be an efficient algorithm to handle different order-picking conditions such as capacity and customer orders. When the amount of orders increase the EDA has an acceptable performance according to the results detailed above. The probability model is useful in order to preserve batches or sequences of batches that appear frequently in all members of the population. The EDA is able to describe the characteristics exhibited by the parents in each generation. Finally, this research contributes by using an EDA as an optimization method for any order-picking process. Future research work will use higher probabilistic models in order to model higher interactions or relationships between variables of the order-picking performance.

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

## References

1. Henn, S.: 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 (2009)

2. de Koster, M., Le-Duc, T., Roodbergen, K.: Design and Control of Warehouse Order Picking: A Literature Review. *European Journal of Operational Research*, 182(2), 481–501 (2007)
3. Wäscher, G.: Order Picking: A survey of planning problems and methods. In Dyckhoff, H., Lackes, R., Reese J. (eds.) *Supply Chain Management and Reverse Logistics*, pp. 323–347. Springer, Berlin (2004)
4. Bozer, Y., Kile, J.: Order Batching in Walk-and-Pick Order Picking Systems. *International Journal of Production Research*, 46(7), 1887–1909 (2008)
5. Öncan, T.: 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 IMECS 2013*, vol. I. Hong Kong (2013)
6. Mühlenbein, H., Paaß, G.: 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*, pp. 178–187, Springer, Berlin (1996)
7. Henn, S.: 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: Business Process Consulting (2010).
8. Kamin, N.: On-line optimization of order picking in an automated warehouse. Ph.D. Dissertation, Shaker Verlag, Aachen (1998)
9. Chew, E., Tang, L.: Travel time analysis for general item location assignment in a rectangular warehouse. *European Journal of Operations Research*, 112(3), 582–597 (1999).
10. Le-Duc, T., de Koster, R.: Travel time estimation and order batching in a 2-block warehouse. *European Journal of Operations Research*, 176(1), 374–388 (2007)
11. van Nieuwenhuysse, I., de Koster, R.: Evaluating order throughput time in 2-block warehouses with time window batching. *International Journal of Production Economics*, 121, 654–664 (2009)
12. Yu, M., de Koster, R.: The impact of order batching and picking area zoning on order picking system performance. *European Journal of Operational Research*, 198(2), 480–490 (2009)
13. Whitley, D., Starweather, T., Shaner, D.: The traveling salesman and sequence scheduling: quality solutions using genetic edge recombination. In: Davis L. (ed.) *Handbook of genetic algorithms*, pp. 350–372, New York (1990)
14. Moscato, P.: On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Caltech Concurrent Computation Program (report 826) (1989)