Ciencia y Reflexión
Revista Científica Multidisciplinaria

Recibido: 06/01/2026
Aprobado: 18/03/2026

DOI: https://doi.org/10.70747/cr.v5i1.843

Un Algoritmo de Agrupamiento Radial para el Problema de Loteo de Pedidos en Línea

Ricardo Pérez Rodríguez
dr.ricardo.perez.rodriguez@gmail.com
https://orcid.org/0000-0001-7401-9669
SECIHTI - TECNM Instituto Tecnológico de Aguascalientes, Departamento Económico Administrativo
Aguascalientes - México

María de los Ángeles Silva Olvera
maria.so@aguascalientes.tecnm.mx
https://orcid.org/0000-0002-7771-7355
TECNM - Instituto Tecnológico de Aguascalientes, Departamento Económico Administrativo
Aguascalientes - México

Sergio Frausto Hernández
sergio.fh@aguascalientes.tecnm.mx
https://orcid.org/0000-0002-9935-1024
TECNM - Instituto Tecnológico de Aguascalientes, Departamento de Ingeniería Química y Bioquímica
Aguascalientes - México

RESUMEN

Una parte fundamental en cadenas de suministro suele tener lugar en un almacén, donde se reciben, procesan y almacenan diferentes materiales o productos para su uso posterior. Dentro de una bodega, las decisiones suelen estar relacionadas con aspectos como la gestión de los pedidos recibidos. El objetivo principal de este artículo es reducir el tiempo de estancia de los pedidos de los clientes mediante la solución del problema de loteo de pedidos en línea. Para solucionar este problema, se crea y sugiere un algoritmo radial de loteo de pedidos. Este enfoque contribuye a ampliar el repertorio de técnicas experimentales basadas en computación evolutiva para resolver problemas del mundo real. La contribución consiste en utilizar un nuevo esquema a partir una función de base radial del hidrógeno, con la que se construye una densidad que se discretiza dinámicamente a partir de la cantidad de pedidos pendientes y así establecer nuevos tiempos de ventana para el proceso de decisión de loteo de pedidos. Se emplean las condiciones necesarias del proceso de loteo de pedidos en línea para demostrar el buen funcionamiento de este método novedoso, así como otros métodos. Se realizaron pruebas estadísticas para confirmar la eficacia del esquema sugerido. Los resultados indican que el desempeño de la bodega mejora al minimizar el tiempo de estancia de los pedidos de los clientes.

Palabras claves: *algoritmo radial, loteo de órdenes, loteo en línea, recolección de materiales*

Los autores declaran no tener conflicto de intereses.
El Editor y los Revisores declararon no tener conflicto de intereses.

A Radial Clustering Algorithm for the Online Order Batching Problem

ABSTRACT

A fundamental part of supply chains typically takes place in a warehouse, where different materials or products are received, processed, and stored for later use. Within a warehouse, decisions are usually related to aspects such as the management of incoming orders. The main purpose of this article is the reduction of the turnover time for all customer orders by addressing the online order batching problem. To overcome the issue, a radial order batching algorithm is created and suggested. This approach helps to increase the repertoire of experimental techniques based on evolutionary computation to resolve real-world issues. The contribution is to use a new scheme based on a radial basis function of hydrogen, with which a density is constructed that is dynamically discretized from the number of pending orders and thus to set new time windows for the batching-decision process. The online order-picking process conditions are employed to demonstrate how well this novel method and other procedures work. Tests of statistics were employed to confirm the effectiveness of the suggested scheme. The findings indicate that the warehouse's performance is enhanced with a minimization on the turnover time of all customer orders.

Keywords: *radial algorithm, order batching, online batching, order picking*

The authors declare no conflict of interest.
The Editor and the Reviewers declared no conflict of interest.

INTRODUCTION

According to Blanchard (2010), the supply chain is a series of actions required to carry out a product or service's life cycle. Let us consider such events from conception to consumption into any market. The aforementioned events involve entities, resources, and activities. Among entities we can find manufacturers, suppliers, wholesalers, retailers among others. The main resources used in the supply chain are material, human, financial, and information. The main actions are distribution, storage, transformation, and acquisition.

Every entity involved in the supply chain makes a variety of decisions as part of its management. The main decisions can be classified into strategic, tactical, and operational (Misni & Lee, 2017).

A warehouse is where many crucial operations of supply chain take place. This location typically receives, processes, and stores materials or products (referred to as items). Subsequently such items are commonly picked up and shipped to the customers.

As with entity of the supply chain, warehouses have managers making strategic decisions. These decisions include long-term considerations like where to locate the warehouse, how to configure it up or automate it, how to choose the best network of suppliers and carriers, and how to choose and modify software systems, among other things. Also, the managers can make tactical decisions to enhance the warehouse performance. Such decisions address mid-term issues including deciding on the best place to keep the goods, establishing standards for quality and safety, deciding on the quantity and placement of depots, and more. Lastly, the managers decide on short-term tasks including predicting daily and weekly demand, planning production schedules, controlling incoming and outgoing inventory, controlling incoming orders, setting strategies to pick up and batch items, among others. Such operational decisions are detailed in Il-Choe & Sharp (1991); De Koster, Le-Duc, & Roodbergen (2007); Misni & Lee

(2017).

The importance of the operational tasks associated with picking orders in the warehouse is highlighted in this study. Specifically, we analyze the impact over the warehouse performance through batching orders.

On a workday into the warehouse, workers receive different orders, from various customers, that containing a collection of goods or materials that need to be taken out of the warehouse, packaged and shipped to their destinations. Some operational decisions into the warehouse consist to set the order to pick the articles, to decide which picker will pick them up, to specify the path the pickers should take, to identify in which batch each order should integrate, to decide when the picker can begin picking, among various other things.

In general, the picking process might consist up to 60% of the total time into the warehouse (Coyle et al 1996). Consequently, to cut down on the overall amount of time, a key operational decision is made when the managers identify which orders should be batched before picking. This is because multiple customer requirements might be collected simultaneously during the same tour.

Order batching is essentially determined by the information that is available about requests from clients. In wide and diverse real-world cases, the requests from clients stochastically get there at various times when the picking procedure might already be underway. The aforementioned feature is called a dynamic order picker system.

Henn (2010) detailed all the online order-batching problem (OOBP). Basically, the pickers walk through the warehouse and pick up articles from different storage locations. 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 about which orders should be processed directly must 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 of a batch is the point in time when a picker starts to process the 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 turnover time is the amount of time for which an order stays in the system. This study focuses on minimizing the average turnover time of all customer orders. The main idea is to form and release batches without having complete information on the types and the arrival times of future orders.

Based on Henn (2010), an optimization model for the OOBP is explained below. Let

n 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 $j \in \{1, \ldots, m\}$,

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

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

The goal is to minimize the average turnover time, i.e.,

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

Equation (2) ensures the assignment of each order to exactly one batch

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

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

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

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

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

In the situation of just one picker, Eq. (5) follows that a batch is started after the previous one is completed

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

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

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

In the OOBP, normally the managers set a time window to create batches. Commonly, setting a time window is done in either, i.e., window batching with a variable time or a fixed time. In the first one, pickers hold off until a specific quantity of items (from different client requests) has come and then gather the products of these client orders in one individual tour (Henn et al 2011). The quantity of objects to gather is determined by the picking device capacity. In the second one, a batch is carried out with the arrival of consumer orders within a specific time frame.

Performance metrics are commonly used in a dynamic picking process for orders. As an example, the turnover time, i.e., the length of time that the order stays in the system. The aim is to reduce the turnover time; it will result in improved service levels.

Van Nieuwenhuysen & de Koster (2009) showed through numerical studies that using predetermined time frame batching can result in shorter average turnover times than using dynamic time frame batching.

The goal of this study is to determine an improved method for setting the window time, i.e., we are interested in finding a better time to set up order batching. To set up order batching should consider the amount of customer orders available into the warehouse, i.e., we believe that the more orders have arrived, the sooner batching of orders should be.

We suggest a radial distribution to determine when order batching should be set up. Such radial distribution is used as input parameter in a genetic algorithm (GA), called Radial Order Batching Algorithm (ROBA) to resolve the (OOBP).

The contribution is to use a new scheme based on a radial basis function of hydrogen, with which a density is constructed that is dynamically discretized from the number of pending orders and thus to set new time windows for the batching-decision process.

The suggested ROBA uses a radial probability model to create new time windows. Basically, through a radial distribution function new time windows are generated.

Radial distribution functions are commonly used in the quantum chemistry perspective. In chemistry, these functions are suitable to explain how an electron behaves inside an atom. The chemical engineers know that the electron commonly has a randomly behavior. Therefore, it is more useful to estimate the electron behavior through a distance metric to identify how far it is from the core. Then, computing the radial distribution function of the electron permits to estimate where it can be in the atomic space at a specific time. This information is relevant in this research, i.e., the separation between the core and the electron, is utilized and transformed to produce new time windows for the OOBP with a single picker.

Figure 1 shows a contour surface of an atom. It makes the atomic space visible. The radial distribution of hydrogen (H) is employed in this study to build new time windows for the OOBP with a single picker. Hydrogen is selected for this research because the mathematical definition of its radial distribution is precise. It allows us to compute with clarity the radial distribution.

In Pérez-Rodríguez & Frausto-Hernández (2023) is described the hydrogen's circumferential dispersion form, $P(r)$.

$$P(r) = 4\left(\frac{Z}{a_0}\right)^3 e^{-\left(\frac{2Zr}{a_0}\right)} r^2 \quad (8)$$

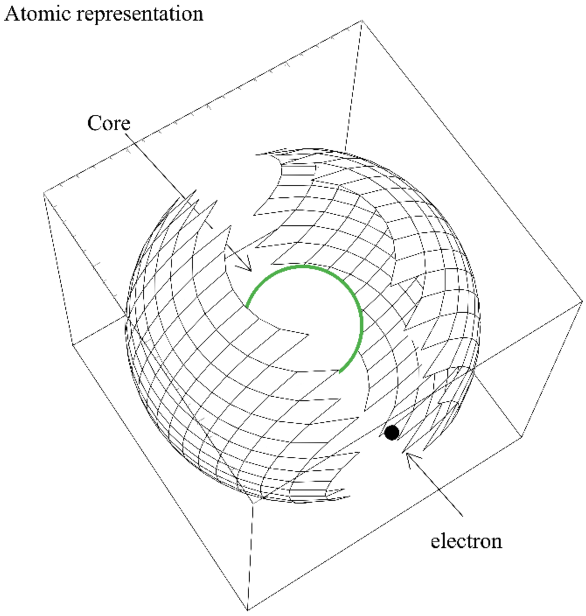
where Z represents the atomic number of the element (for instance, hydrogen has an atomic number of 1); a_0 represents the Bohr radius, while r signifies the distance (radius), measured in picometers (pm), from the electron and the nucleus.

The definition of the Bohr radius is

$a_0 = \frac{h^2}{(4\pi^2 me)} = 52.9 \text{ pm}$, the electron's mass is denoted by m, its charge by e, and the Planck constant by h.

This study's contribution is the application of such a radial probability function as input parameter in a GA scheme, and from this function to generate new time windows to resolve the OOBP. The suggested ROBA's performance is contrasted with those of the more recent algorithms that effectively solve the OOBP. By using the probability radial function' hydrogen in the OOBP with a single picker as a probability model to improve the GA's performance, this study advances the state of the art. Furthermore, it is a gap among the current studies from the online perspective, i.e., probability radial functions have not been the aim and scope of the previous studies. In this research, we defined an improved method for setting window time by a probability radial model. The findings of this study allow for the conclusion that radial probability distributions are a new area of study for creating effective GAs.

Figure 1. An illustration of the shape exterior of a hydrogen atom



Source: own design

Literature review

Despite, as objective functions, minimizing the completion time (or cost) have been previously studied for scenarios with dynamic arrival of orders, the most frequently cited objective functions in the articles for the OOBP are the turnover time (and the picking time), whether it is a singular picker or multiple pickers.

Heuristics, and metaheuristics algorithms are the most common used to tackle the OOBP featuring one picker and several pickers. Henn (2009) presented a discussion regarding the latest findings on the OOBP, where both a single picker, and several pickers are present. A part of that discussion, along with other contributions, is outlined below.

Tang & Chew (1997) minimize the average turnover time over a situation where the arrival order follows a Poisson process. The minimization is carried out by establishing a batch size n in a queuing system $E_n/G/c$.

A real-world batching problem involving the retrieval of greeting cards from a warehouse was examined by Kamin (1998). In this configuration, the products are collected by pickers using automated-guided vehicles. Like other articles, this one focuses on minimizing average turnover times, and like any OOBP, orders come in throughout the study horizon.

Chew & Tang (1999) also minimize the average turnover time by reducing the journey and service durations. Once more, a Poisson process governs the arrival order, and the minimization is carried out by establishing a batch size n in a queuing system $E_n/G/c$.

For the OOBP, Hsu et al (2005) created a GA method. The authors minimized the navigational distances in 2D, and 3D warehouses of different layouts as objective function.

In a two-block warehouse, Le-Duc & De Koster (2007) aim to reduce the average throughput time, or the duration an order remains in the system before being served. The problem was treated as a queuing system $M/G^k/1$, and orders arrive according to a Poisson process.

In 2009, Yu & de Koster described an order-picking procedure that used multiple zones of the same size. Each batch's articles are picked up by zones in a sequential manner. The average turnover times were estimated by the researchers.

Schleyer & Gue (2012) also minimize of the average throughput time. Here, the problem was treated as a queuing system $G/G/1$, and the receipt of orders does not have to follow a Poisson process; it can be any constant incoming flow of orders.

Other papers also consider **the arrival of orders follows a Poisson process** such as Xu et al (2014), and Pérez-Rodríguez et al (2015).

Henn (2012) resolved an OOBP in a walk-and-pick warehouse where the objective is to minimize the completion times of all (dynamically arriving) customer orders (or the makespan). To tackle the online situation, the author modified the offline order batching solution approaches.

Another example to resolve the order batching is found in Li et al (2022). The authors utilized a Tabu Search (TS) method for resolving a shuttle-based storage and retrieval system with low carbon emissions, with a multi-item order batching and retrieval system.

Although the contribution, of the aforementioned papers, falls in the OOBP category, there exist other relevant reported works to tackle situations between the OOBP, and other issues. An extensive discussion over the OOBP, and other issues can be found in Boz & Aras (2022). A part of that discussion, along with other contributions, is outlined below.

The online order batching, and waiting problem (OOBWP) has been examined by Bukchin et al. (2012). The average expenses related to the pickers' overtime and tardiness are minimized by the authors by a proposed heuristic based on Markov decision process (MDP-H).

Giannikas et al (2017) minimize the average completion time in a scenario with dynamic arrival of orders by a proactive order picking approach designed to enhance the agility of order fulfillment systems. In 2020, Gil-Borrás et al reduce the picking time by evaluating and comparing several time-window strategies.

The online order batching, and routing problem (OOBRP) has been tackled by Ene & Öztürk (2012). In a two-block warehouse, the authors minimize the travel expense as a function of the transit time by means of a GA.

A hybrid technique was described by Azadnia et al (2013). The authors clustered client orders using a weighted association rule mining technique. After that, to create batches, the authors employed a binary integer programming model. Lastly, a GA to address the picker routing issue is considered by the authors to minimize the total tardiness. Öncan (2013) introduced a GA for the order batching issue taking into account traversal and return routing strategies. The suggested GA is put to the test and contrasted against a popular cost-saving method to minimize the total tour length of all pickers. The author utilized a set of randomly generated instances thorough the comparison. The findings showed that the suggested GA succeeded better in acceptable computation times.

Li et al (2016) reduce the overall travel distance using an algorithm that relies on Ant Colony Optimization (ACO). The group of writers took into account facilities with more than 10,000 orders and several blocks.

Ardjmand et al (2019) proposed GAs, and simulated annealing (SA) algorithms to address the issue using randomly created data of varying sizes, both small and large.

Gil-Borrás et al (2021) simultaneously minimize the collecting time, finishing time, and the variations in the pickers' assignments by a Variable Neighborhood Method (VND).

Xie et al (2023) developed a novel VND to address the combined issue more effectively in multi-depot AGV-supported mixed-shelves warehouses.

The online order batching, and sequencing problem (OOBSP) has been addressed by Pinto & Nagano (2019). The author suggested a GA solution to reduce the collection method's overall duration and cost.

Finally, the online order batching, sequencing, and routing problem (OOBSRP) has been analyzed by Won & Olafsson (2005). The picking and turnover times are reduced by the writers. Chen et al (2015) use a hybrid algorithm. The authors develop a nonlinear mathematical model that seeks to reduce client order delays. To resolve the model, the authors employed mixed genetic and ACO algorithms. Table 1 summarizes the most common techniques to resolve the OOBP. Also, the Table 1 includes the work's significance.

Based on the previous literature review, there is a gap at present by not considering radial distributions in the solution. Heuristics, and metaheuristics, well-known benchmarking techniques, should enhance their performance if any radial function is incorporated in them. Basically, this research incorporates the radial function of hydrogen to enhance the performance of the GA to resolve the OOBP.

Table 1. Literature review summarization

Author	Problem					Technique	Objective function													
	OOBP	OOBWP	OOBRP	OOBSP	OOBSRP		ATT	ND	ATHT	MK	CER	ATO	PT	TC	TT	TTL	TTD	TTC	PTT	DT
Tang & Chew (1997)	•					by queuing system $E_n/G/c$	•													
Kamin (1998)	•					by experimental techniques	•													
Chew & Tang (1999)	•					by queuing system $E_n/G/c$	•													
Hsu et al (2005)	•					GA		•												
Le-Duc & De Koster (2007)	•					by queuing system $M/G^k/1$			•											
u & de Koster (2009)	•					by experimental techniques	•													
Schleyer & Gue (2012)	•					by queuing system $G/G/1$			•											
Pérez-Rodríguez et al (2015)	•					A continuous estimation of distribution algorithm (CEDA)	•													
Henn (2012)	•					by experimental techniques				•										
Li et al (2022)	•					TS					•									
Bukchin et al. (2012)		•				MDP-H						•								
Giannikas et al (2017)		•				by an interventionist order picking strategy				•										
Gil-Borrás et al (2020)		•				by experimental techniques							•							
Ene & Öztürk (2012).			•			GA								•						
Azadnia et al (2013)			•			GA									•					
Öncan (2013)			•			GA										•				
Li et al (2016)			•			ACO											•			
Ardjmand et al (2019)			•			GA & SA											•			
Gil-Borrás et al (2021)			•			VND				•			•							
Xie et al (2023).			•			VND											•			
Pinto & Nagano (2019)				•		GA												•		
Won & Olafsson (2005)					•	by heuristics													•	
Chen et al (2015)					•	GA & ACO														•
This research	•					*Radial GA*	•													

ATT: the average turnover time, ND: the navigational distances, ATHT: the average throughput time, MK: the makespan, CER: the carbon emission reduction, ATO: the average tardiness and overtime, PT: the picking time, TC: the travel cost, TT: the total tardiness, TTL: the total tour length of all pickers, TTD: the total travel distance, TTC: the time and total cost, PTT: the picking and turnover time, DT: the delay of customer orders

METHODOLOGY

ROBA for the OOBP

Characterization of the OOBP

A simple queue process is created to manage the customer orders arriving into the warehouse. Three situations should be considered in the queue. The first one, customer orders arriving, and waiting to be attended. The second one, customer orders being batched by the ROBA. Also, it includes defining the time the orders will leave the warehouse. The third one, customer orders leaving the warehouse. Then, the turnover time can be computed for each order.

In this study, client requests flow via a Poisson process through a horizon time. A time window is defined when the batching is carried out with the available orders until that time. The time window is established via hydrogen's radial dispersion. A detailed process to set the time window is outline below

Radial distribution of hydrogen computing

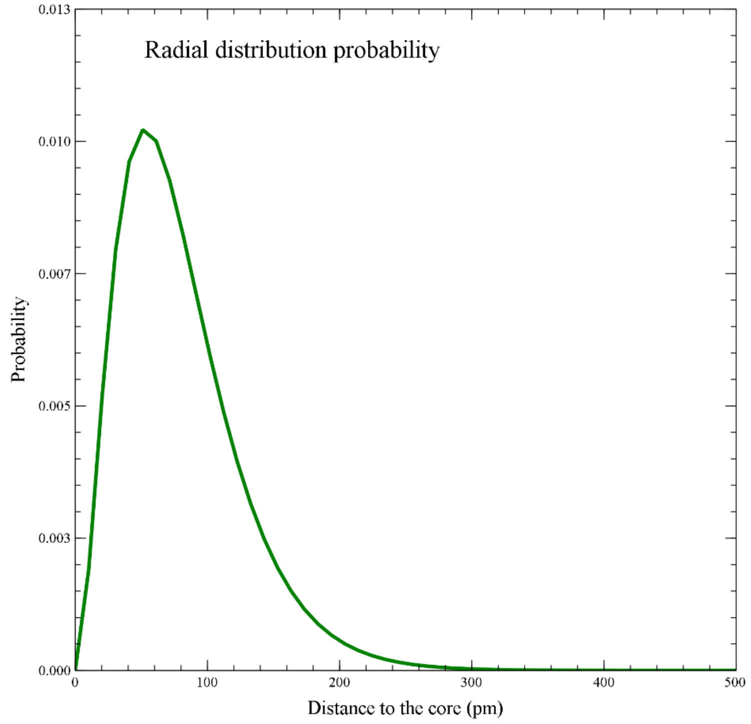
Using Eq. (8), the likelihood of radial dispersion is depicted in Figure 2. It is evident that the function quickly deteriorates in relation to its distance from the core.

A cumulative distribution ought to be constructed using this radial distribution function. Consequently, new time frames for the OOBP are created using the cumulative distribution.

Figure 3 depicts the cumulative distribution. The density is dynamically discretized, and conditionally depends on the quantity of unfulfilled client orders to be batched. The greater the number of pending orders, the smaller the interval size for establishing the next time window.

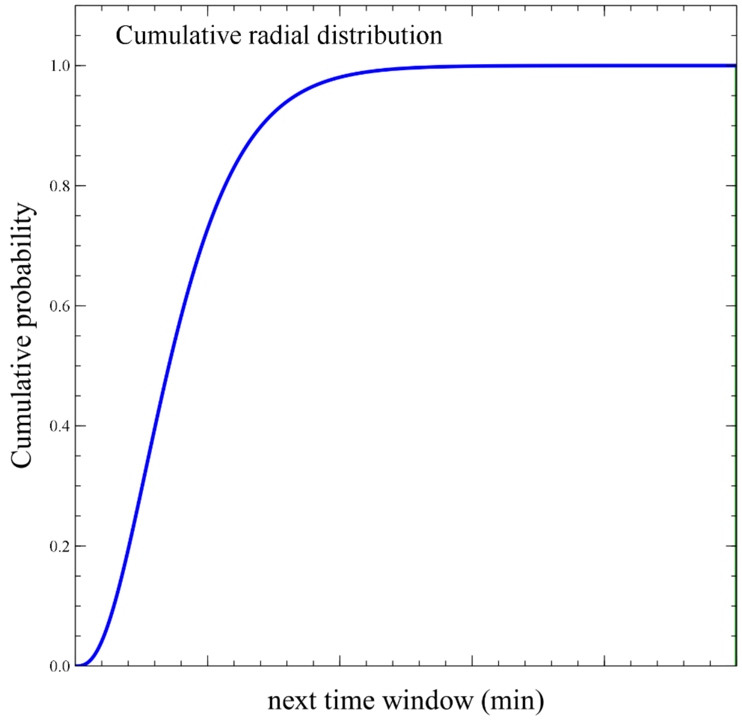
Normally, in the previous studies, the fixed time window is established in advance, and the variable time window requires the information of the picking device capacity as input parameter. In this research, the time window is established considering the pending customer orders to be batched. In this way, we believe that the more orders have arrived, the sooner batching of orders should be.

Figure 2. A radial distribution probability



Source: own design

Figure 3. A cumulative radial distribution probability



Source: own design

Solution representation

After computing the cumulative radial distribution, and setting the next time window for the pending customer orders, we initialize the initial population of the ROBA.

As almost any GA for the OOBP, a solution vector can be defined as a set of numbers each one represents a batch number where the corresponding customer order will be batched. A solution vector will have n elements, where each element represents a specific batch. Figure 4 depicts an example with 10 pending customer orders.

Figure 4. A solution representation

Pending customer orders	1	2	3	4	5	6	7	8	9	10
Batch	A	A	B	B	A	C	C	C	D	D

Source: own design

Table 1. Initial Population

K $\leftarrow$ Get the number of pending customer orders

P $\leftarrow$ Create a set with the pending customer orders

D $\leftarrow$ Generate M individuals each of length K

Input: K, P, D , picking device capacity

for each $-i-$ member from D **do**

P' $\leftarrow$ Create an auxiliary set from P

capacity_used $\leftarrow 0$

batch $\leftarrow 1$

orders $\leftarrow 0$

while orders $< K$ **do**

P'_k $\leftarrow$ randomly choose a $-k-$ pending customer order from P'

capacity_used $\leftarrow$ capacity_used + articles from P'_k

if capacity_used $\leq$ picking device capacity **then**

at P'_k position of the $-i-$ member in $D \leftarrow$ batch

else

capacity_used $\leftarrow$ articles from P'_k

batch $\leftarrow$ batch + 1

at P'_k position of the $-i-$ member in $D \leftarrow$ batch

endif

orders $\leftarrow$ orders + 1

$P' \leftarrow$ remove P'_k from P'

endwhile

endfor

Output: D

Source: own design

It is important to mention that the batching process only can be made with available customer orders that already arrived into the warehouse. Therefore, the necessary information about orders must be known at the time of the batching process.

The initial population of solution vectors is totally random, i.e., to set a batch for a specific customer order is only considering the picking device capacity used for the picking process. As a size of population, we consider $M = 100$ members per generation.

Fitness

At this elemental level, the ROBA computes, and sets the aptitude, for each member of the population, as the duration of the entire tour needed to retrieve the items for all the corresponding batches. To determine the overall duration of the tour, "S" shape routing is established into the warehouse as a routing policy.

Fitness procedure

$K \leftarrow$ Get the number of pending customer orders

$A \leftarrow$ Create a set with the amount of articles in each pending customer order

$I \leftarrow$ Create a matrix with each required item in each pending customer order

$FitD \leftarrow$ Generate a vector for the fitness population of length M

```
D′ ← Create a copy of D
Input: D , D′ , FitD, K, A, I
    for each − i − member from D   do
        B ← Get the number of batches from the member
        batch ← 1
        fitness ← 0
        while batch ≤ B   do
            parts ← 0
            order ← 0
            while order < K   do
                if at ′order′ position of the − i − member in D′ <> 0   then
                    lot ← at ′order′ position of the − i − member in D′
                    if lot == batch   then
                        parts ← parts + articles at ′order′ position from A
                    endif
                endif
                order ← order + 1
            endwhile
            picklist ← Create a empty set of length ′parts′
            order ← 0
            while order < K   do
                if at ′order′ position of the − i − member in D′ <> 0   then
                    lot ← at ′order′ position of the − i − member in D′
                    if lot == batch   then
                        for each − item − of the corresponding ′order′ in I   do
                            picklist ← add the required − item − in the picklist
                        endfor
                    endif
                    D′ ← set 0 at ′order′ position of the − i − member in D′
```

endif

order $\leftarrow$ order + 1

endwhile

picklist $\leftarrow$ Sort the picklist

fitness $\leftarrow$ CALL SUBFUNCTION < APTITUDE > AND CARRY THE PICKLIST

$FitD \leftarrow FitD\ +$ fitness for the $-\ i -$ member in D

picklist $\leftarrow$ Destroy the picklist

batch $\leftarrow$ batch + 1

endwhile

endfor

Output: $FitD$

SUBFUNCTION < APTITUDE >

Input: picklist

W $\leftarrow$ Set the warehouse configuration

aptitude $\leftarrow$ Compute the tour length required to pick the items up in W

Output: aptitude

Basically, the fitness detailed in procedure above is computed only using a single picker. In the current state of the investigation, it is the main methodological limitation to apply the ROBA scheme in other picking environments. To obtain a generalization to multi-picker or parallel picking systems, an event-discrete simulation model must be built to handle each completion time of any batch made for each picker. Thus, the precise remaining of pending orders will be used to discretize the density from the cumulative distribution of the hydrogen.

Offspring

Descendants are obtained by uniform crossing. A batch es elected from two parents, previously randomly selected. Then, the batch is assigned for the descendant confirming the picking device capacity is not violated. Until every outstanding client order has a batch assigned to it, the procedure is carried out. Once the offspring have been produced, their fitness must be computed.

Offspring production

K ← Get the quantity of unfulfilled orders from customers

P ← Create an auxiliary set of length equivalent to the greatest quantity of batches from D

A ← Create a set with the amount of articles in each pending customer order

S ← Generate M individuals each of length K

Input: D, S, K, P, A, picking device capacity

```
for each − i − member from S  do
    l ← randomly choose a member from D
    m ← randomly choose another member from D
    for each − p − pending customer order  do
        flag ← 0
        while flag == 0  do
            batch ← randomly choose a batch between l and m
            if P_batch + A_p ≤ picking device capacity  then
                P_batch ← P_batch + A_p
                at − i − position of the member in S ← batch
                flag ← 1
            else
                k ← search any empty position in P
                batch ← k
                P_batch ← P_batch + A_p
                at − i − position of the member in S ← batch
                flag ← 1
            endif
        endwhile
    endfor
endfor
```

Output: S

Replacement

The best candidates are chosen using the bubble approach, i.e., a single population is constituted, with parents and offspring, to execute the bubble process.

Replacement procedure

Input: $D, S, FitD, FitS$

$G \leftarrow D \cup S$

$G \leftarrow$ Sort G based on $FitD$ & $FitS$

$D_t \leftarrow$ Set the best members from G

$FitD \leftarrow$ Set the corresponding fitness from the best members from G

Output: $D, FitD$

Every step in the suggested algorithm has been specified. The ROBA framework is provided below

ROBA framework

$R \leftarrow$ Compute a cumulative radial distribution from Eq. (1)

Setting next time window from R for pending customer orders

$D \leftarrow$ Generate M individuals

$FitD \leftarrow$ Assess each individual's fitness from D

$Best \leftarrow$ Keep the top candidate from D.

Do

$S \leftarrow$ Sampling from D

$FitS \leftarrow$ Assess each individual's fitness from S

$Best \leftarrow$ If you apply, update the most suitable candidate from $FitS$

$D \leftarrow$ Substituted by bubble method using D & S

Until the requirement for stopping is satisfied

Output: $Best$

RESULTS AND COMPARISON

To show the advantages of dynamically adjusting the time window using the radial distribution, Table 2 depicts a comparison using the fixed time window against the radial time window. The comparison is made with the parameters describe below

30 customer orders arriving through a period of 8 hours, 900 storage spaces divided into 10 hallways each including 90 storage spaces, only one available picker, picking device capacity; 45 articles, routing strategy; S-shape, fixed time window; each hour, 10 generations as a stopping criterion, 50 runs for the experiment.

Table 2. Comparison between time window approaches

	Fixed time window	Radial time window
Average turnover time	166.94 min	145.50 min
Average processing time for each order	31.55 min	27.50 min

Source: own design

The following algorithms are suggested for comparison with the ROBA scheme to confirm the scientific relevance of this study.

The VND scheme proposed by Gil-Borrás et al (2021)

The VND approach shown by Xie et al (2023)

The TS method discussed by Li et al (2022)

The relative percentage increase (RPI), for the turnover time, is calculated to assess the effectiveness of every algorithm. Each of the previously listed algorithms' outputs was acquired by the direct implementation of the authors.

$$RPI(tt_i) = \frac{tt_i - tt_*}{tt_*} \quad (2)$$

tt_* is the best turnover time, and tt_i is the turnover time obtained in the i^{th} trial.

We conducted 50 trials to take into consideration the order-picking warehouse's stochastic nature, where $M = 100$ individuals form the population in each generation. When the gap between the trial's average fitness and the optimum is under 5%, the trials are terminated. Then, each trial returns the best solution.

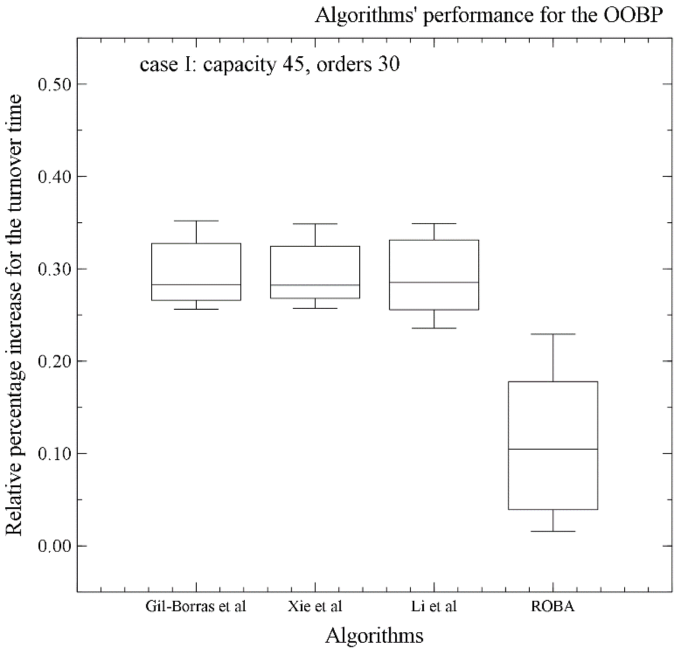
We used a Lenovo® ideapad 330 computer, AMD® A9-9425 Radeon R5, 5 Compute Cores 2C+3G processor, 3.1 GHZ, 8 GB of RAM, Windows® 10 for 64 bits to run each algorithm. The algorithms are encoded within C++®

In order to obtain a concise assessment, a workload was made for the comparison among algorithms. The workload includes a series of varying client orders, and the kinds of items needed for each order, and also incorporates different numbers of articles per order, emulating the online order-picking process conditions and environment. It is unknown beforehand when the client requests will arrive. Our experiments focused on the warehouse layout used by Henn (2009). The warehouse has two cross aisles and is one block long. Ten hallways, each with 90 storage places, create the 900 storage spots in the picking zone. We likewise presume that a picker takes 30 seconds to navigate among 10 storage locations and that it takes them 10 seconds to locate and select an item. Furthermore, our experiments take into account the parameters utilized by Pérez-Rodríguez et al (2015), i.e., the picking device capacity K, we used two capacities, 45 and 75 articles, the "S-Shape" route was used as the routing strategy, and for the number of orders n, we considered 30 and 60 for a workday. Within the allotted eight hours for planning, the orders should arrive. With a value λ known as the arrival rate, the times between arrivals—specifically, the duration from the arrival of order i to order $i + 1$—follows an exponential distribution. Denote the count of incoming orders during the time interval $[0, t]$ as $X(t)$. When the inter-arrival periods $E[X(t)] = \lambda$ are exponentially distributed, then t is true. We select λ in our numerical trials so that, for $t = 8[\text{hours}]$, the expectation $E[X(t)]$ equals n. In conclusion, we set λ to the subsequent values: $\lambda = 0.08625$ for $n = 30$ and $\lambda = 0.125$ for $n =$

60.

Figure 5 shows the distribution of the experimental outcomes in detail. The performance attained for every trial when there are 30 orders and 45 articles in the capacity. The results of the other algorithms fall between 0.25 and 0.35, but the majority of ROBA results are concentrated between 0 and 0.25. The performance of the ROBA was superior.

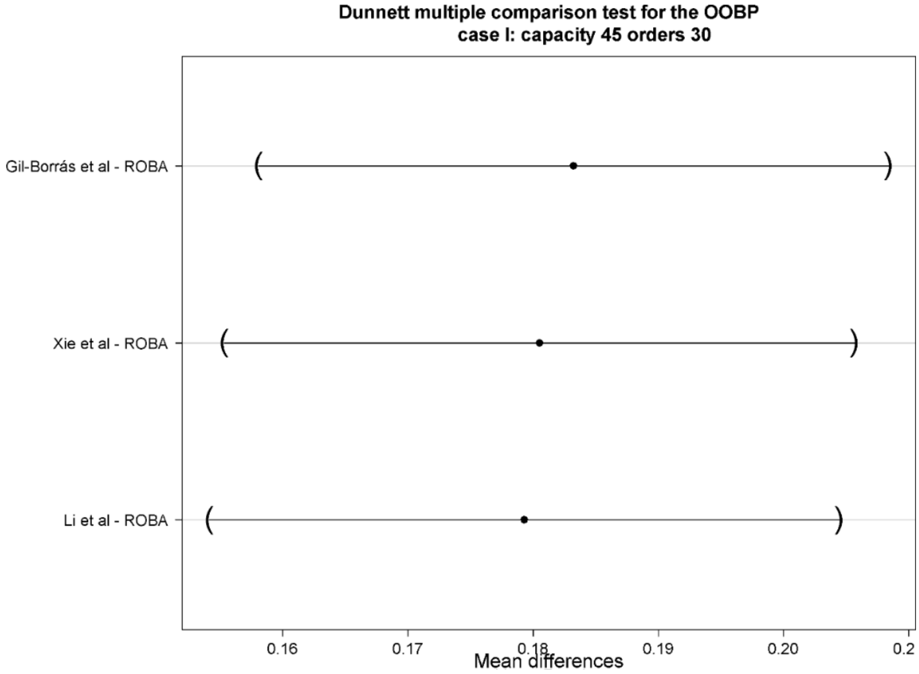
Figure 5. Performance of the ROBA, with $n = 30$ and $K = 45$



Source: own design

Figure 6 illustrates the ROBA performance using a Dunnett statistical test. The difference between the ROBA and the comparison algorithms is statistically significant.

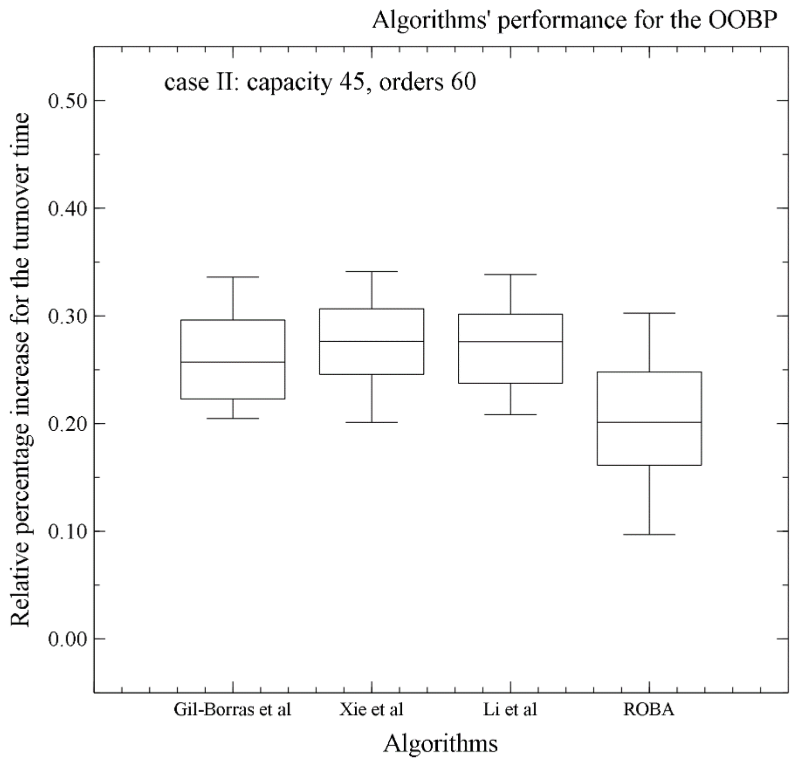
Figure 6. Dunnett test, with $n = 30$ and $K = 45$



Source: own design

Figure 7 depicts the distributionn of the experimental outcomes. The results achieved for every trial when the capacity is 45 articles and the number of orders is 60. The results of the other algorithms fall between 0.20 and 0.35, but the majority of ROBA results are concentrated between 0.10 and 0.30. The performance of the ROBA was superior.

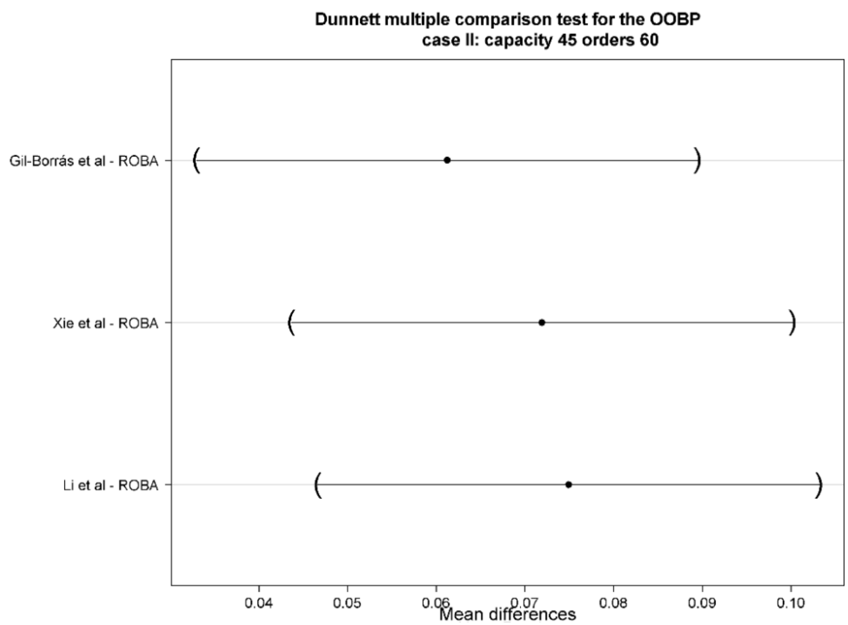
Figure 7. Performance of the ROBA, with $n = 60$ and $K = 45$



Source: own design

The ROBA performance is shown in Figure 8 using a Dunnett statistical test. The difference between the ROBA and the comparison algorithms is statistically significant.

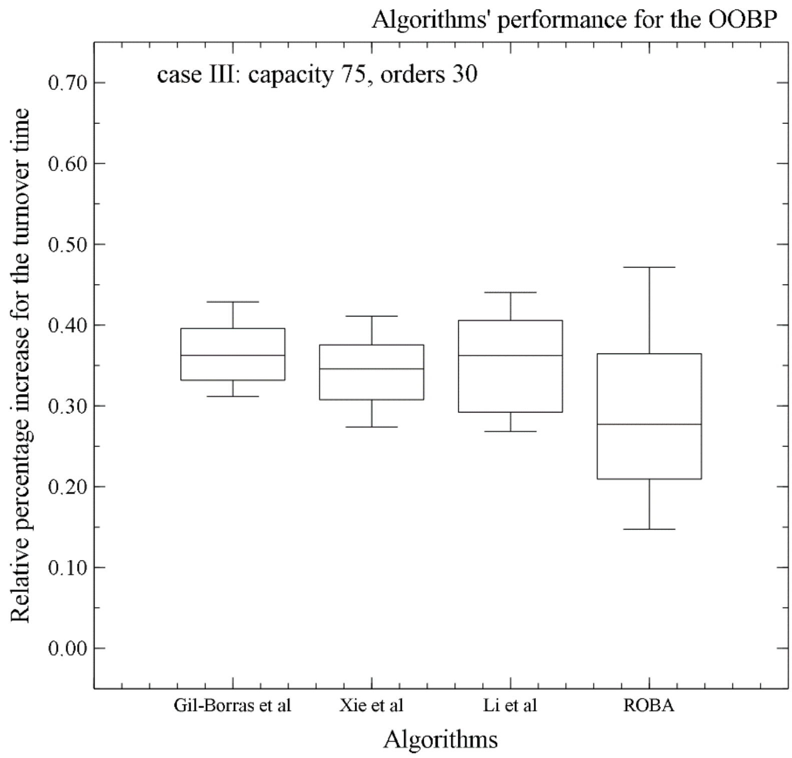
Figure 8. Dunnett test, with $n = 60$ and $K = 45$



Source: own design

Figure 9 displays the distribution of the experimental findings. The results achieved for every trial when the capacity is 75 articles and the number of orders is 30. The other algorithms' median is situated near 0.35, while the ROBA's median is situated near 0.30. The performance of the ROBA was superior.

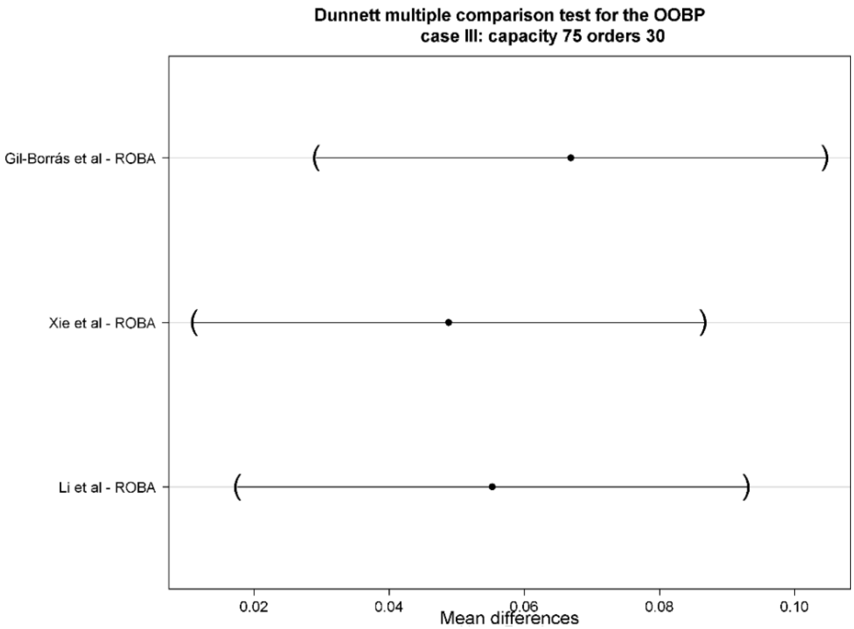
Figure 9. Performance of the ROBA, with $n = 30$ and $K = 75$



Source: own design

Figure 10 displays the results of another Dunnett statistical test to reveal the ROBA performance. Once more, the difference between the ROBA and the comparison algorithms is statistically significant.

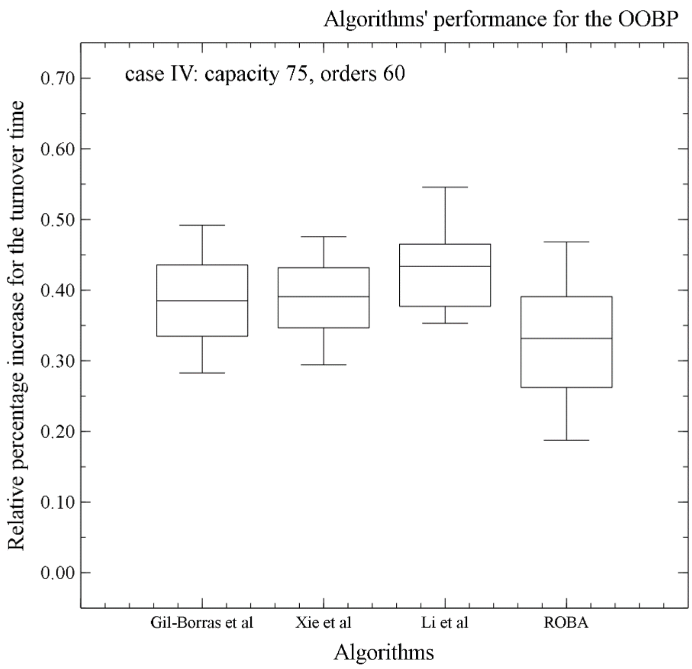
Figure 10. Dunnett test, with $n = 30$ and $K = 75$



Source: own design

Figure 11 presents the distribution of the experimental outcomes. The results of each experiment when the capacity is 75 articles and the number of orders is 60. The other algorithms' median is situated near 0.40, while the ROBA's median is assigned near 0.33. The performance of the ROBA was superior.

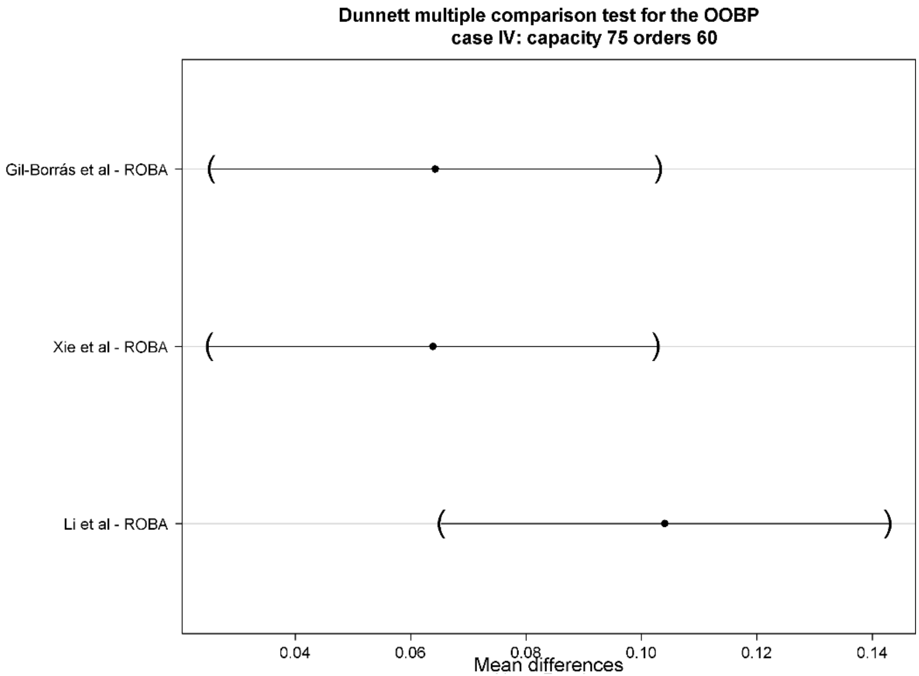
Figure 11. Performance of the ROBA, with $n = 60$ and $K = 75$



Source: own design

Figure 12 describes the ROBA performance using a different Dunnett statistical test. Once more, the difference between the ROBA and the comparison algorithms is statistically significant.

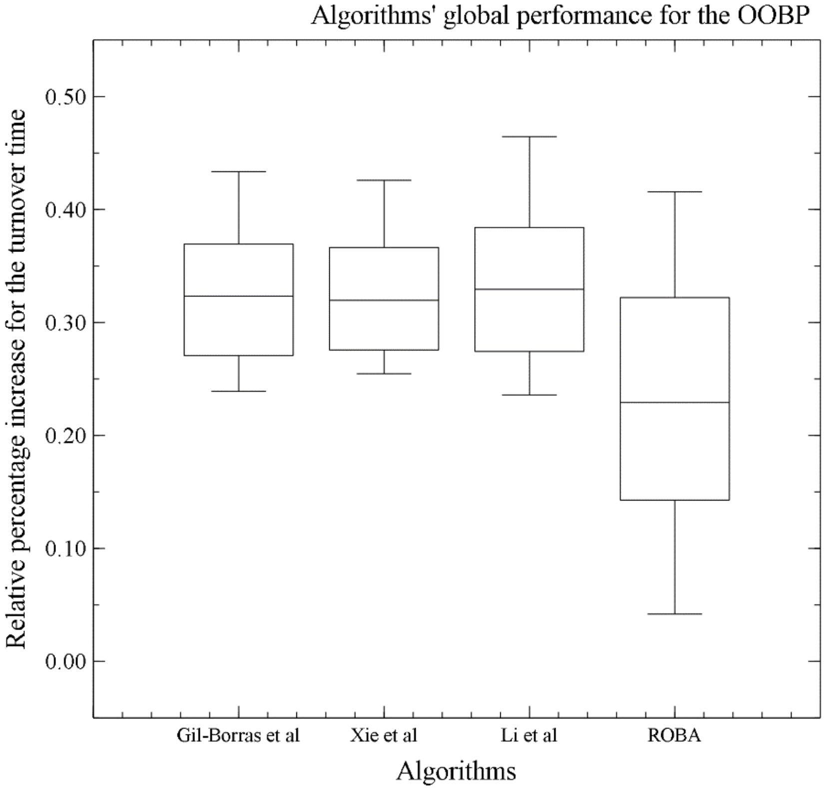
Figure 12. Dunnett test, with $n = 60$ and $K = 75$



Source: own design

Lastly, Figure 13 illustrates how the experimental data were distributed. the overall performance attained for every experiment. The other algorithms' median is situated near 0.33, while the ROBA's median is assigned near 0.22. The performance of the ROBA was superior.

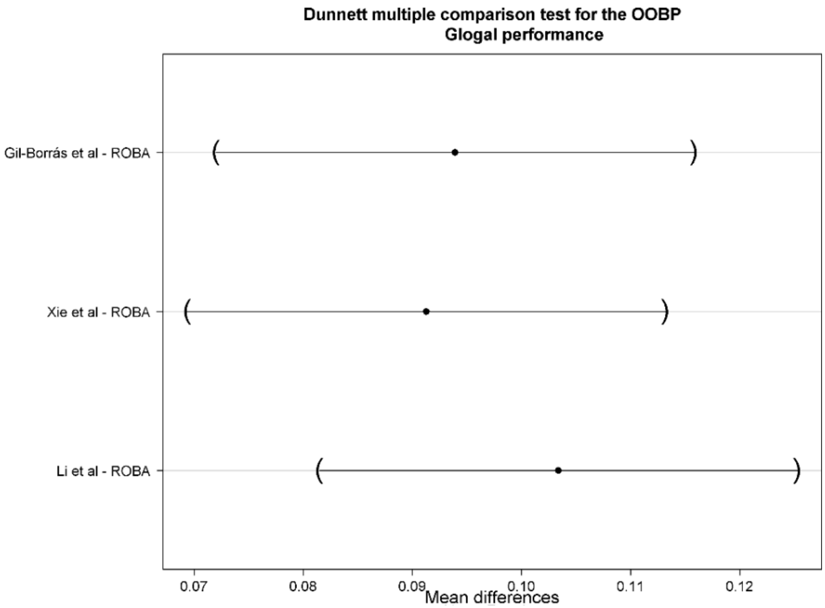
Figure 13. Global performance of the ROBA



Source: own design

Figure 14 provides an illustration of the final Dunnett statistical test used to analyze the ROBA performance. Once more, the difference between the ROBA and the comparison algorithms is statistically significant.

Figure 14. Dunnett test for the global performance



Source: own design

In all the comparisons, the ROBA scheme outperforms the algorithms used to obtain the best performance. Four different scenarios have been considered to determine the best performing ones. Also, a global comparison was made to show the effectiveness of the proposed approach. The radial distribution used in this research was able to identify the best solutions for the experimental design proposed in this section.

CONCLUSIONS AND FUTURE RESEARCH

The contribution of this research was shown by diverse experimental runs. To use a radial probability function as input parameter in a GA scheme, should be considered in future research.

This paper contributes to a new way of generating new time windows to outperform the order batching decision-process for the OOBP.

Based on the previous results, the ROBA scheme is competitive to resolve the OOBP. A gap in the literature exists, and it is a good opportunity to show the benefits of using the radial distributions to tackle the OOBP. The online order-picking process conditions were well incorporated in this research. The ROBA scheme was successfully implemented to minimize the turnover time.

When adding certain operators to the GA and the TS in the evolutionary process, the algorithms utilized in the comparison maintain diversity in terms of convergence and diversity. Additionally, the trials end when there is a 5% or less discrepancy between the trial's average and optimal fitness.

Since the number of function evaluations required for convergence is unknown a priori, we modified the original halting criterion, which called for a predetermined number of generations. The suggested ROBA is presently in the prototype stage in terms of computation time and expense. Consequently, these factors have not yet been taken into account in our study.

Regarding the ROBA's benefits and limitations, we may say that it considers the radial distribution function of the hydrogen to set the next time window as an advantage. However, it is unknown which radial distribution is best for the OOBP.

There are currently no reliable, accurate methods for locating a global optimum, and it is also impossible to compare the results with an optimal solution. It is a prominent feature of the problem regarding online optimization.

Regarding computational complexity, the online order-batching issue is NP hard, similar to the offline problem type identified by Gademann and van de Velde (2005), when the number of orders in each batch exceeds two.

The main drawback, of the ROBA scheme, it hasn't been encoded for particular users, and therefore can't handle unexpected or invalid inputs. Nonetheless, the ROBA system can be altered to provide a practical module for particular industry users.

As future research work, we are going to deem a mobile application, in order to be utilized for users in industry.

In light of these results, we consider to taste other radial functions to improve the GA scheme on the OOBP as future research.

Acknowledgments

We would like to thank each and every reviewer for their insightful criticism that helped us improve the manuscript. Additionally, we would like to express our gratitude to Prof. Dr. Luis Urban-Rivero for his insightful remarks.

Declaration of interest: The writers have declared that they have no conflicts of interest. The funders were not involved in the study's design, data collection, analysis, or interpretation, manuscript writing, or the choice to publish the findings.

Author contributions: conceptualization, Pérez-Rodríguez; methodology, Pérez-Rodríguez; investigation, all authors; formal analysis, all authors; writing—all authors prepared the first draft; all authors reviewed and edited the writing; all authors created the visualization.

The submitted version of the work has been read and approved by all authors.

Funding: no outside funding was obtained for this study.

Data availability: the corresponding author can provide the data from this study upon request, Pérez-Rodríguez (dr.ricardo.perez.rodriguez@gmail.com).

BIBLIOGRAPHIC REFERENCES

Ardjmand, E., Bajgiran, O.S., & Youssef, E. (2019). Using list based simulated annealing and genetic algorithm for order batching and picker routing in put wall based picking systems. *Appl Soft Comput*, 75, 106–119. DOI: 10.1016/J.ASOC.2018.11.019

Azadnia, A.H., Taheri, S., Ghadimi, P., Mat Saman, M.Z., & Wong, K.Y. (2013). Order batching in warehouses by minimizing total tardiness: A hybrid approach of weighted association rule mining and genetic algorithms. *Sci World J*, 1, 246578. DOI: 10.1155/2013/246578

Blanchard, D. (2010). *Supply chain management best practices*. John Wiley & Sons. DOI: 10.1002/9781119202912

Boz, E., & Aras, N. (2022). The order batching problem: A state of the art review. *Sigma Journal of Engineering and Natural Sciences*, 40(2), 402-420. DOI: 10.14744/sigma.2022.00018

Bukchin, Y., Khmelnitsky, E., & Yakuel, P. (2012). Optimizing a dynamic order-picking process. *European Journal of Operational Research*, 219, 335–346. DOI: 10.1016/j.ejor.2011.12.041

Chen, T.L., Cheng, C.Y., Chen, Y.Y., Chan, L.K. (2015). An efficient hybrid algorithm for integrated order batching, sequencing and routing problem. *Int J Prod Econ*, 159, 158–167. DOI: 10.1016/j.ijpe.2014.09.029

Chew, E.P., & Tang, L.C. (1999). Travel time analysis for general item location assignment in a rectangular warehouse. *European Journal of Operational Research*, 112, 582–597.

Coyle, J.J., Bardi, E.J., & Langley, C.J. (1996). *The management of business logistics: A supply chain perspective*. West Publishing Company Minneapolis/St. Paul.

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

Ene, S., & Öztürk, N. (2012). Storage location assignment and order picking optimization in the automotive industry. *The International Journal of Advanced Manufacturing Technology*, 60, 787–797. DOI: 10.1007/s00170-011-3593-y

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. DOI: 10.1080/07408170590516917

Giannikas, V., Lu, W., Robertson, B., & McFarlane, D. (2017). An interventionist strategy for warehouse order picking: Evidence from two case studies. *Inter. Journal of Production Economics*, 189, 63–76. DOI: 10.1016/j.ijpe.2017.04.002

Gil-Borrás, S., Pardo, E.G., Alonso-Ayuso, A., & Duarte, A. (2021). A heuristic approach for the online order batching problem with multiple pickers. *Computers & Industrial Engineering*, 160, DOI: 10.1016/j.cie.2021.107517

Gil-Borrás, S., Pardo, E.G., Alonso-Ayuso, A., & Duarte, A. (2020). Fixed versus variable time window warehousing strategies in real time. *Progress in Artificial Intelligence*, 9, 315–324. DOI: 10.1007/s13748-020-00215-1

Henn, S. (2012). Algorithms for on-line order batching in an order picking warehouse. *Computers and Operations Research*, 39, 2549–2563. DOI: 10.1016/j.cor.2011.12.019

Henn, S., Koch, S., & Wäsher, G. (2011). *Order batching in order picking warehouses: a survey of solution approaches*. Working paper 01/2011, Ottovon-Guericke-University. Faculty of Economics and Management. Magdeburg, Germany.

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.

Henn, S. (2009). *Metaheuristics for the order batching problem in manual order picking systems*. Ottovon-Guericke-University. Faculty of Economics and Management. Magdeburg, Germany

Hsu, C.M., Chen, K.Y., & Chen, M.C. (2005). Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Comput Ind*, 56, 169–178. DOI: 10.1016/j.compind.2004.06.001

Il-Choe, K., & Sharp, G.P. (1991). *Small parts order picking: design and operation*. Technical report, Atlanta, EEUU. School of Industrial and Systems Engineering. Georgia Institute of Technology.

Kamin, N. (1998). *On-line optimization of order picking in an automated warehouse*. Shaker Verlag, Aachen.

Le-Duc, T., & De Koster, R.B.M. (2007). Travel time estimation and order batching in a 2-block warehouse. *European Journal of Operational Research*, 176, 374–388. DOI: 10.1016/j.ejor.2005.03.052

Li, H., Lyu, J., Zhen, L., & Zhuge, D. (2022). A joint optimisation of multi-item order batching and retrieving problem for low-carbon shuttle-based storage and retrieval system. *Cleaner Logistics and Supply Chain*, 4, DOI: 10.1016/j.clscn.2022.100042

Li, J., Huang, R., & Dai, J.B. (2016). Joint optimisation of order batching and picker routing in the online retailer's warehouse in China. *International Journal of Production Research*, 55, 447–461. DOI: 10.1080/00207543.2016.1187313

Misni, F., & Lee, L.S. (2017). A review on strategic, tactical and operational decision planning in reverse logistics of green supply chain network design. *Journal of Computer and Communications*, 5, 83–104. DOI: 10.4236/jcc.2017.58007

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.

Pérez-Rodríguez, R., & Frausto-Hernández, S. (2023). A radial hybrid estimation of distribution algorithm for the truck and trailer routing problem. *Mathematical and Computational Applications*, 28, 27, DOI: 10.3390/mca28010027

Pérez-Rodríguez, R., Hernández-Aguirre, A., & Jöns, S. (2015). A continuous estimation of distribution algorithm for the online order-batching problem. *The International Journal of Advanced Manufacturing Technology*, 79, 569–588. DOI: 10.1007/s00170-015-6835-6

Pinto, A.R.F., & Nagano, M.S. (2019). An approach for the solution to order batching and sequencing in picking systems. *Prod Eng*, 13, 325–341. DOI:10.1007/s11740-019-00904-4

van Nieuwenhuysen, I., & de Koster, R. (2009). Evaluating order throughput time in 2-block warehouses with time window batching. *International Journal of Production Economics*, 121, 654-664.

Schleyer, M., & Gue, K.R. (2012). Throughput time distribution analysis for a one-block warehouse. *Transportation Research Part E: Logistics and Transportation Review*, 48, 652–666. DOI: 10.1016/j.tre.2011.10.010

Tang, L.C., & Chew, E.P. (1997). Order picking systems: Batching and storage assignment strategies. *Computers & Industrial Engineering*, 33, 817– 820. DOI: 10.1016/S0360-8352(97)00245-3

Won, J., & Olafsson, S. (2005). Joint order batching and order picking in warehouse operations. *International Journal of Production Research*, 43, 1427–1442. DOI: 10.1080/00207540410001733896

Xie, L., Li, H., & Luttmann, L. (2023). Formulating and solving integrated order batching and routing in multi-depot AGV-assisted mixed-shelves warehouses. *European Journal of Operational Research*, 307, 713-730. DOI: 10.1016/j.ejor.2022.08.047

Xu, X., Liu, T., Li, K., & Dong, W. (2014). Evaluating order throughput time with variable time window batching. *International Journal of Production Research*, 52, 2232–2242. DOI: 10.1080/00207543.2013.849009

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. DOI: 10.1016/j.ejor.2008.09.011

La Revista utiliza códigos para identificar al Revisor y al equipo de PARES REVISORES. Si tiene dudas o consultas, contacte a: contacto@cienciayreflexion.org

Código de Editor: 103

Código de Revisores: 843