logistics

MDPI

Article

Order Picking Problem: A Model for the Joint Optimisation of Order Batching, Batch Assignment Sequencing, and Picking Routing

Antonio Maria Coruzzolo [1,*], Francesco Lolli [1], Elia Balugani [1], Elisa Magnani [1] and Miguel Afonso Sellitto [2]

[1] Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy; francesco.lolli@unimore.it (F.L.); elia.balugani@unimore.it (E.B.); magnanielli@gmail.com (E.M.)
[2] Production and Systems Graduate Program, University of Vale do Rio dos Sinos, Avenida Unisinos 950, São Leopoldo 93022-180, Brazil; sellitto@unisinos.br
* Correspondence: antoniomaria.coruzzolo@unimore.it

Abstract: *Background:* Order picking is a critical activity in end-product warehouses, particularly using the picker-to-part system, entail substantial manual labor, representing approximately 60% of warehouse work. *Methods:* This study develops a new linear model to perform batching, which allows for defining, assigning, and sequencing batches and determining the best routing strategy. Its goal is to minimise the completion time and the weighted sum of tardiness and earliness of orders. We developed a second linear model without the constraints related to the picking routing to reduce complexity. This model searches for the best routing using the closest neighbour approach. As both models were too complex to test, the earliest due date constructive heuristic algorithm was developed. To improve the solution, we implemented various algorithms, from multi-start with random ordering to more complex like iterated local search. *Results:* The proposed models were tested on a real case study where the picking time was reduced by 57% compared to single-order strategy. *Conclusions:* The results showed that the iterated local search multiple perturbation algorithms could successfully identify the minimum solution and significantly improve the solution initially obtained with the heuristic earliest due date algorithm.

Keywords: order picking problem; order batching; batch assignment-sequencing; picking routing; heuristics

Citation: Coruzzolo, A.M.; Lolli, F.; Balugani, E.; Magnani, E.; Sellitto, M.A. Order Picking Problem: A Model for the Joint Optimisation of Order Batching, Batch Assignment Sequencing, and Picking Routing. *Logistics* **2023**, *7*, 61. https://doi.org/10.3390/logistics7030061

Academic Editors: Tomasz Nowakowski, Artur Kierzkowski, Agnieszka A. Tubis, Franciszek Restel, Tomasz Kisiel, Anna Jodejko-Pietruczuk, Mateusz Zając and Mladen Krstić

Received: 26 June 2023
Revised: 21 August 2023
Accepted: 29 August 2023
Published: 11 September 2023

1. Introduction

End-product warehouses play a fundamental role within the supply chain as platforms to temporarily store products before sending them to end customers. They are used in most industrial and commercial activities to maximise the service level and minimise storage costs simultaneously [1]. Although warehouses account for approximately 20% of the total cost for organisations, they are essential because they contribute to many company missions [2]. In a typical scenario, the following activities take place inside a warehouse: receiving and storing goods; preparation and order processing; shipping and exit of goods. In general, logistics companies deal with equipment of different origins that must operate according to different strategies, such as line-feed kitting (intracell) or milk-run (intercell) line feed [3]. In both cases, features such as modularity and scalability of the system can be helpful if demand changes over time [4].

Order picking is critical to manage and operate warehouses efficiently. Order picking aims to transform the load unit, with which a product is generally stored in the warehouse, into a format more suited to the customer's needs according to the service level at the minimum cost. Orders consist of order lines referring to a specific product and indicating the specific quantity requested by a customer [5]. When an order contains multiple lines, these must be accumulated and ordered before being transported to the shipping area. Therefore, the picking operation consists of planning, preparing, and releasing the customer's order by taking the products required from the various picking boxes in which they are stored [6].

As indicated in [7], the picking activity is characterised by a high intensity of manual labor. It is estimated to account for approximately 60% of warehouse work, with a consequent impact on both the overall logistic costs and the service level provided to the customer. By breaking down the warehouse costs, the picking activity is typically the most laborious and expensive, with a cost of up to 55% of the total operating costs of the warehouse [8].

The picking system must be organised and executed as effectively as possible because any malfunction of this phase can lead to unsatisfactory service, high operating costs, and, consequently, the deterioration of the entire supply chain. An efficient picking process must be designed in a robust and optimally controlled manner. The literature solves a related problem: how to determine a batch sequence and, at the same time, assign workers to batches according to pre-determined requirements [9].

This study focuses on one of the picking systems most used in warehouses, namely, the picker-to-part system. Over 80% of Western European organisations use this methodology, which is the primary system for the picking activity [7]. In this activity, each operator carries out a "picking mission" in the picking area guided by a paper or an RF/voice terminal. Pickers walk or drive along the corridors to pick up the required items and complete a single order or batch of orders, depending on the picking logic.

This study analyses the possibility of grouping customer orders in batches so that the items belonging to a batch are collected during a single picking tour. Each customer order is generally characterised by the quantity to be picked and a loading date; the latter indicates when the order must be completed and ready for shipment. A breach of the loading date entails additional costs and a lower service level. The feasibility of meeting the deadlines for a set of customer orders depends on the following factors:

1. How the customer orders are grouped in batches (order batching problem);
2. How the batches are assigned and sequenced to the operators (batch assignment sequencing problem);
3. How each order picker is routed to collect the items of each picking order (picking routing problem).

The three issues are highly interrelated and may involve trade-off considerations, necessitating an optimisation approach. If addressed independently, there is a possibility of encountering a local, suboptimal, or even unacceptable solution. Consequently, a collective, balanced resolution is required to address these challenges effectively. Therefore, the purpose of this article was to propose a joint optimisation model to reach an acceptable solution concurrently for the batching, assigning, sequencing, and routing problems. The objective function seeks to minimise the overall completion time while ensuring that total tardiness is not compromised and that orders are not excessively anticipated. The specific objectives are to theoretically formulate a model, simulate an application in a real case, and compare the application's result with other usual strategies. The primary novelty of this study lies in introducing a comprehensive model that addresses multiple correlated problems while simultaneously reducing the number of variables that need to be managed. To the best of our knowledge and supported by a search on the Scopus database relying on related keywords, this is the first study that tackles the abovementioned problems jointly.

The remaining content is organised as follows. Section 2 reviews some of the most relevant studies and highlights the novelties of this study. Section 3 describes the problem and introduces the model notation. Section 4 describes the mathematical model and explains the heuristic approaches used. Section 5 describes the experimental analysis. Section 6 outlines the conclusions along with some scope for further research.

2. Literature Review

Only recently, researchers have started to focus on topics related to the ordering, sequencing, and picking routing problems to identify a simultaneous, complete, and advantageous solution to these problems. By simultaneously considering the ordering, sequencing, and picking routing activities, the operating costs can be reduced, and the service level can be improved, e.g., by reducing the overall travel time by more than 35% [10]. Most studies dealt with these problems separately or focused on the simultaneous resolution of two or more subproblems of the joint order batching, assignment, sequencing, and routing problems (JOBASRPs). In order to position the study in front of others that are seminal and current, a brief survey in databases using the keywords "problem-solving", "warehousing", "batching orders", "assigning orders", "sequencing orders", and "routing" provided a sample of articles that, after being screened, produced the content in Table 1. Such content ensures the novelty of the proposed study.

Table 1. Comparison of this study with others in JOBASRP.

Study	Batching	Assigning	Sequencing	Routing
[11]	x		x	
[12]	x		x	
[13]	x		x	x
[14]	x			x
[15]	x		x	
[16]	x			x
[17]	x	x	x	
[18]	x		x	x
[19]	x			x
[20]	x			x
[21]	x	x		x
[22]	x			x
[23]	x		x	
[24]	x	x	x	
[25]	x			
Our study	x	x	x	x

The order-picking problems needed to be considered jointly because the selection of orders composing the batches and the sequence in which the batches are assigned and processed influences both the planning of the picking routes end the fulfilment of loading dates [13]. Ref. [18] proposed a mathematical model for solving the joint order batching, sequencing, assignment, and routing problem (JOBSPRP). In this model, which includes only linear constraints, the number of variables and constraints increases polynomially with increasing customer orders. It is significantly more advantageous than traditional models.

The first subproblem is the order batching problem, which involves grouping a set of customer orders in batches to minimise the total length of a tour. From a mathematical point of view, the order batching problem can be seen as a travelling salesman problem (TSP) with unique properties derived from the warehouse layout [26]. Several approaches have been proposed to solve this problem (e.g., [7,27]).

The second subproblem is the picking routing problem, generally solved as a Steiner TSP, where some nodes must not be visited, and others can be visited more than once [7]. The Steiner TSP can also be formulated as a classic TSP if the minimum distance between each pair of storage locations is previously calculated [14,28]. The picking routing problem involves picking a given set of items stored in known storage locations to minimise the total distance travelled.

The last subproblem is the order batch assignment and sequencing problem, which involves the assignment of batches to the minimum number of pickers and sequencing the batches for each picker so that the overall tardiness is reduced. To tackle this problem, researchers have considered different factors, such as due dates [28], and scenarios, such as multiple pickers [29], heterogeneous pickers [17], and multiple pickers in the online environment [30].

Some mathematical formulations have also been proposed for the JOBASRP considering the due dates [31], 3D warehouses [18], and multiple pickers [31,32]. Ref. [32] published a series of mathematical models examining actual distribution centres, and considering the minimisation of the total distance, tardiness, and earliness. However, their models are highly computationally complex; in fact, the number of binary variables and constraints increases exponentially with the complexity of the problem, and in particular with an increase in warehouse locations. The JOBASRP is a recent problem that has been extensively studied and represents a promising path for research. The most accredited and complete formulations of this problem are those proposed in [30,33] as indicated in [34].

The proposed model only requires the definition of some feasible batches and their execution time, which is the main difference in front of other models already mentioned. As the number of feasible batches grows exponentially with the increase in the number of variables, this activity involves a significant amount of computational time and the risk of being unable to identify an acceptable solution, even for small instances, owing to memory restrictions. To limit the number of variables, in this study we developed an integer linear model in which the number of variables and constraints grows polynomially with the increasing complexity of the problem.

Furthermore, this study changes the constraints for defining the routing strategy. The proposed mathematical model assumes that a picker always starts the tour from an initial depot and concludes it in a final depot to make the model more flexible and realistic. It is worth noting that this assumption is not critical because the input data can be easily set to make these two depots physically coincident and separated only from a logical point of view.

3. Problem Description and Notation

This study considers a warehouse with a manual, low-level picker-to-parts order-picking system from which a given set of items must be retrieved. The items are stored on pallets or in bins directly accessible to the operators [27]. More operators work simultaneously in the picking area, and each travels on a picking device, allowing him/her to collect more items during a single tour. Each tour starts from the initial depot and ends at the final depot. The initial depot is where the operator starts the tour, and the pick list and picking cart are assigned to the picker. In the final depot, the operator delivers the picking orders to be prepared for shipment. In the proposed model, these depots are assumed to be separate to increase the flexibility of the proposed model; however, they can be considered coincident if necessary. The order picker is guided by a pick list containing a set of orders along with the storage positions of the requested items and their respective quantities to retrieve them on the same tour. The pick list also indicates the operator's path to reach the required locations. The maximum number of stops a picker can make during the same tour depends on the capacity of the picking vehicle.

A batch is an aggregation of a set of customer orders which is picked in a single picking tour. The time spent by an order picker to retrieve all the items of a batch can be divided as follows [7]:

1. The setup time, i.e., the time spent preparing a tour;
2. The search time, i.e., the time spent identifying the correct item;
3. The picking time, i.e., the time spent physically retrieving the items from their storage locations;
4. The travel time, i.e., the time spent to reach the picking locations of the batch items.

The execution time of a batch starts at the batch start time, i.e., when the picker to whom the batch has been assigned starts preparing the picking tour. The batch is completed when the operator has picked all the items needed for a batch and returns them to the final depot. The start time of an order corresponds to the start time of the batch to which it is assigned; similarly, the completion time of the customer order corresponds to the completion time of the batch to which it is assigned.

Customer orders must be picked before a specific due date to guarantee an adequate service level to the end customers. Whether such due dates are met depends on how the orders are assigned to the batches, their sequencing, and the defined routing strategy. More precisely, the tardiness ta_n of an order n is defined as the non-negative difference between the completion time v_n and its due date d_n [16], expressed as $ta_n = max\{v_n - d_n; 0\}$. The sum of the tardiness of all customer orders, called total tardiness, must be null to obtain an acceptable solution at the company level. The earliness ea_n of an order n is defined as the non-negative difference between its due date d_n and the completion time v_n, which can be expressed as $ea_n = max\{d_n - v_n; 0\}$; the sum of the earliness of all customer orders is called total earliness. The excessively early preparation of orders is not desirable because it leads to the unnecessary use of warehouse space.

The JOBASRP can be stated as follows. Given a non-empty set of orders, some items with known storage locations must be removed from the warehouse. Each order is characterised by a due date by which all requested items should be retrieved and brought to the final depot. Some order pickers are available for the necessary picking operations. Then, the following questions must be answered (simultaneously) in such a way that the total tardiness, earliness, and completion time are minimised [28]:

- How should the set of customer orders be grouped into picking orders? (Order Batching Problem);
- How and in which sequence should the set of picking orders be assigned to the order pickers? (Batch Assignment-Sequencing Problem);
- In which sequence should the respective pick locations be visited for each order? (Picking Routing Problem).

4. The Problem Formulation

4.1. Sets and Parameters

Before presenting the formulation of the model in detail, it is worth introducing the sets, parameters, and variables used.

Sets:

- Set of costumer orders: $N = 1..num\ order$.
- Set of locations to visit: $V = 1..num\ location$.
- Set of missions that can be performed: $H = 1..num\ mission$.
- Set of periods, for each picker, in which a batch can be collected: $K = 1..num\ period$.
- Set of pickers: $P = 1..num\ picker$
- Set of feasible batches: $B = 1..num\ batch$

Parameters:

- $w_1 = Weight\ assigned\ to\ completion\ time$
- $w_2 = Weight\ assigned\ to\ the\ weighted\ sum\ of\ earlinesses\ and\ tardinesses$
- $\gamma = Weight\ assigned\ to\ the\ total\ earliness$
- $\partial = Weight\ assigned\ to\ the\ total\ tardiness$
- $d_{ij} = distance\ between\ picking\ locations\ i \in V\ and\ j \in V$
- $cap_n = capacity\ request\ by\ order\ n \in N$
- $Cap_{vehicle} = capacity\ of\ the\ picking\ vehicle$
- $a_{ni} = \begin{cases} 1\ If\ the\ order\ n\ \in N\ must\ visit\ the\ picking\ location\ i\ \in V \\ 0\ otherwise \end{cases}$
- $\beta^{time} = time\ required\ to\ travel\ by\ a\ unit\ of\ space$
- $\beta^{setup} = setup\ time\ required\ for\ each\ batch$
- $\beta^{picking} = time\ required\ for\ searching\ and\ picking\ items$
- $due\_carico_n = due\ date\ for\ the\ order\ n \in N$
- $M = sufficiently\ large\ positive\ number$
- $dis_i = distance\ travelled\ to\ complete\ the\ batch\ i \in B$

- $capbatch_i = capacity\ request\ by\ batch\ i \in B$
- $b_{in} = \begin{cases} 1\ If\ the\ batch\ i\ \in B\ contains\ the\ order\ i\ \in N \\ 0\ otherwise \end{cases}$

Variables:

- $ta_n = tardiness\ of\ the\ order\ n \in N$
- $ea_n = earliness\ of\ the\ order\ n \in N$
- $t\_exec_h = execution\ time\ of\ the\ mission\ h \in H$
- $u_{pk} = processing\ time\ for\ the\ batch\ assigned\ to\ the\ position\ k \in K\ of\ picker\ p \in P$
- $v_{pk} = completion\ time\ for\ the\ batch\ assigned\ to\ the\ position\ k \in K\ of\ picker\ p \in P$
- $x_{pkh} = \begin{cases} 1\ If\ the\ mission\ h \in H\ is\ assigned\ to\ the\ position\ k \in K\ of\ picker\ p \in P \\ 0\ otherwise \end{cases}$
- $y_{nh} = \begin{cases} 1\ If\ the\ order\ n\ \in N\ is\ assigned\ to\ mission\ h\ \in H \\ 0\ otherwise \end{cases}$
- $z_{ijh} = \begin{cases} 1\ if\ the\ arc\ that\ connects\ the\ picking\ locations \\ i \in V\ and\ j \in V\ are\ included\ in\ the\ mission\ h \in H \\ \\ 0\ otherwise \end{cases}$
- $f_{ijh} = Number\ of\ units\ passing\ through\ the\ arc\ that\ connects\ the\ picking\ locations\ i \in V\ and\ j \in V\ in\ the\ mission\ h \in H$
- $t\_exec_i = execution\ time\ of\ the\ batch\ i \in B$
- $x_{ikh} = \begin{cases} 1\ If\ the\ batch\ i \in B\ is\ assigned\ to\ the\ position\ k \in K\ of\ picker\ p \in P \\ 0\ otherwise \end{cases}$

The proposed model generates a solution that minimises the completion time and the weighted sum of the total tardiness and earliness. The aim is to generate a solution to increase the picking area's productivity by reducing the acceptable preparation time at the organisational level. To make the definition of the solution more flexible, each of these factors is assigned a weight, which can be modified according to organisational needs.

For the formulation of the model, the capacity of the picking vehicle is measured in several items, as in [28,29,35]. Note that this is not a critical assumption because the number of items can be easily modified to consider other capacity constraints (e.g., the maximum number of customer orders or the maximum total weight of items).

The mathematical model formulation can be divided into two parts. The first part is related to the joint order batching problem and picking routing problem; here, the customer orders are grouped into batches, and the corresponding tours are constructed. The second part is related to the batch assignment and sequencing problem. Here, the batches are assigned to an operator and arranged in a specific sequence. The picker starts processing the batch assigned to the first position, and after reaching the final depot and returning to the initial one, he/she processes the batch assigned to the next position. This process is repeated until the sequence is completed. This procedure is related to the total completion time and the completion time of each batch, which are necessary to calculate the tardiness and earliness of each customer order.

The model is defined using Equations (1)–(32).

$$min\left(w_1 \times \left(\left(\sum_{p=1}^{N}\sum_{k=1}^{K} v_{pk}\right)\right) + w_2 \times \left(\gamma \times \sum_{n=1}^{N} ea_n + \partial \times \sum_{n=1}^{N} ta_n\right)\right) \tag{1}$$

$$\sum_{h \in H} x_{pkh} \leq 1; \qquad \forall p \in P, k \in K \tag{2}$$

$$\sum_{p \in P}\sum_{k \in K} x_{pkh} = 1; \quad \forall h \in H \tag{3}$$

$$\sum_{h \in H} y_{nh} = 1; \qquad \forall n \in N \tag{4}$$

$$\sum_{n \in N} cap_n \times y_{nh} \leq Cap_{Vehicle}; \qquad \forall h \in H \tag{5}$$

$$\sum_{j \in J | j > 1} z_{1jh} = 1; \qquad \forall h \in H \tag{6}$$

$$\sum_{j \in J | j < num\_location} z_{inum\_locationh} = 1; \quad \forall h \in H \tag{7}$$

$$\sum_{i \in V | i > 1} \sum_{h \in H} z_{i1h} = 0 \tag{8}$$

$$\sum_{j \in V | j < num\_location} \sum_{h \in H} z_{num\_locationjh} = 0 \tag{9}$$

$$\sum_{i \in V | i < num\_location} z_{ijh} \geq a_{nj} \times y_{nh}; \quad \forall h \in H, n \in N, j \in V | j \geq 2 \tag{10}$$

$$\sum_{j \in V | j > 1} z_{ijh} \geq a_{ni} \times y_{nh}; \qquad \forall h \in H, n \in N, i \in V | i \leq num\ location - 1 \tag{11}$$

$$f_{ijh} \leq Cap_{Vehicle} \times z_{ijh}; \quad \forall h \in H, i \in V | i \leq num\ location - 1, j \in V | j \geq 2 \tag{12}$$

$$\sum_{g \in V | g < num\_location} z_{gih} = \sum_{j \in J | j > 2} z_{ijh}; \qquad \forall h \in H, i \in V | 2 \leq i \leq num\ location - 1 \tag{13}$$

$$\sum_{g \in V | g < num\_location} z_{gih} \leq 1; \qquad \forall h \in H, i \in V | 2 \leq i \leq num\ location - 1 \tag{14}$$

$$\sum_{j \in J | j > 2} z_{ijh} \leq 1; \qquad \forall h \in H, i \in V | 2 \leq i \leq num\ location - 1 \tag{15}$$

$$\sum_{g \in V | g < num\_location} f_{gih} = \sum_{j \in J | j > 2} f_{ijh} \geq a_{ni} \times y_{nh} \qquad \forall h \in H, n \in N, i \in V | 2 \leq i \leq num\ location - 1 \tag{16}$$

$$z_{iih} = 0; \qquad \forall h \in H, i \in V | 2 \leq num\ location - 1 \tag{17}$$

$$\beta^{setup} + \beta^{picking} \times \left(\sum_{n \in N} cap_n \times y_{nh} \right) + \beta^{time} \times \left(\sum_{i \in V | i < num\_location} \sum_{j \in J | j > 2} d_{ij} \times z_{ijh} \right) \leq t_{exech}; \qquad \forall h \in H \tag{18}$$

$$t_{exech} - M \times \left(1 - x_{pkh} \right) \leq u_{pk} \qquad \forall p \in P, k \in K, h \in H \tag{19}$$

$$u_{p1} \leq v_{p1}; \ \forall p \in P \tag{20}$$

$$u_{pk} + v_{pk-1} \leq v_{pk}; \ \forall p \in P, k \in K, |k \geq 2 \tag{21}$$

$$v_{pk} - due\_date_n - M \times \left(2 - x_{pkh} - y_{nh}\right) \leq ta_n;\ \forall p \in P, k \in K, h \in H, n \in N \quad (22)$$

$$due\_date_n - v_{pk} - M \times \left(2 - x_{pkh} - y_{nh}\right) \leq ea_n;\ \forall p \in P, k \in K, h \in H, n \in N \quad (23)$$

$$x_{pkh} \in \{0,1\};\forall\ p \in P, k \in K, h \in H \quad (24)$$

$$y_{nh} \in \{0,1\};\ \forall h \in H, n \in N \quad (25)$$

$$z_{ijh} \in \{0,1\};\ \forall h \in H, i \in V | i \leq 1..num\ location - 1, j \in V | j \geq 2 \quad (26)$$

$$t\_exec_h \geq 0;\ \forall h \in H \quad (27)$$

$$ta_n \geq 0; \quad \forall n \in N \quad (28)$$

$$ea_n \geq 0; \quad \forall n \in N \quad (29)$$

$$u_{pk} \geq 0; \quad \forall p \in P, k \in K \quad (30)$$

$$v_{pk} \geq 0; \quad \forall p \in P, k \in K \quad (31)$$

$$f_{ijh} \geq 0 \quad \forall h \in H, i \in V | i \leq 1..num\ location - 1, j \in V | j \geq 2 \quad (32)$$

Equation (1) is a mono-objective function. The first term minimises the total completion time, and the second term is the weighted sum of the total tardiness and earliness. The weights can give more relevance to either term; in this way, the solution that best fits the specific needs of each operative environment can be determined.

Equations (2)–(17) show the model's constraints. Equation (2) ensures that a maximum of one mission is assigned to each picker at each position, and Equation (3) guarantees that each mission is performed. Equations (4) and (5) ensure that each customer order is executed on a specific picking mission and that the capacity of a picking vehicle is not exceeded, respectively.

Equations (6)–(17) show the routing constraints. Equations (6) and (7) ensure that each mission starts from the initial depot and ends in the final one, respectively; Equations (8) and (9) ensure that the initial depot has no inputs and the final one has no outputs, respectively. Equations (10) and (11) ensure that each picking location has only one entrance and one exit, respectively. Equation (12) represents the capacity constraint on the arc, Equation (13) ensures that the number of incoming arcs in a vertex is equal to the number of outgoing arcs, and Equations (14) and (15) guarantee that each vertex has only one incoming arc and one outgoing arc, respectively. Equations (16) and (17) prevent the formation of cycles in the path and itineraries that end without crossing all the required locations, respectively. These constraints force each mission to pass only once from each compartment, with the consideration that the initial and final deposits cannot have entries and exits, respectively.

Equation (18) determines the execution time for each mission, composed of the setup time, the time to search and pick up the items, and the travel time. The execution time of a mission assigned to a particular position of a given operator is calculated with Equation (19). Equation (20) determines the completion time in the first period, which corresponds precisely to the execution time of the mission, and Equation (21) determines the completion time of the missions assigned to successive periods. In this case, it is necessary to consider the mission assigned to the period considered plus the completion time of the previous periods. Finally, Equations (22) and (23) determine the tardiness and earliness of each order, respectively. The third value subtracted allows the consideration that it is necessary to calculate the delay or advance of an order to evaluate the batch's completion time. Equations (24)–(25) define the domains of the variables.

4.2. Simplified Model

This study modified the mathematical model proposed in the previous section for defining and sequencing batches. Based on the model proposed in [36], this study eliminated the constraints and variables related to the picking routing problem to reduce the complexity of the model, particularly that of the picking routing problem.

However, eliminating the constraints and variables related to the picking routing problem complicates the inputs. In particular, the proposed modified model requires the input of all the executable batches as initial data. Therefore, as the number of orders increases, the computational times increase significantly because the number of batches also increases. Hence, it may not be possible to identify an improved solution even for small batches due to memory restrictions. As the model does not deal with the definition of routing strategies, this study implemented a heuristic algorithm that defines the path and distances of each possible feasible batch before the execution of the model. The closest neighbour is a standard heuristic algorithm for the TSP that optimises the route by joining the nearest nodes each time. The proposed modified model can be mathematically expressed using Equations (33) to (48).

$$min\left(w_1 \times \left(\sum_{p \in P}\sum_{k \in K} v_{pk}\right) + w_2 \times \left(\gamma \times \sum_{n \in N} ea_n + \partial \times \sum_{n \in N} ta_n\right)\right) \tag{33}$$

$$\sum_{i \in B} x_{ipk} \leq 1; \quad \forall p \in P, k \in K \tag{34}$$

$$\sum_{i \in B}\sum_{p \in P}\sum_{k \in K} x_{ipk} \times b_{in} = 1; \quad \forall n \in N \tag{35}$$

$$Cap\ batch_i \times x_{ipk} \leq Cap_{vehicle}; \quad \forall p \in P, k \in K, i \in B \tag{36}$$

$$\beta^{setup} + \beta^{picking} \times \left(Cap\ batch_i \times x_{ipk}\right) + \beta^{time} \times \left(\sum_{i \in B} dis_i \times x_{ipk}\right) \leq t\_exec_i\ i \in B \tag{37}$$

$$t\_exec_i - M \times \left(1 - x_{ipk}\right) \leq u_{pk}; \quad \forall p \in P, k \in K, i \in B \tag{38}$$

$$u_{p1} \leq v_{p1}; \quad \forall p \in P \tag{39}$$

$$u_{pk} + v_{pk-1} \leq v_{pk}; \qquad \forall p \in P, k \in K, |k \geq 2 \tag{40}$$

$$v_{pk} - due_{caricon} - M \times \left(2 - x_{ipk} - b_{in}\right) \leq ta_n; \quad \forall p \in P, k \in K, h \in H, n \in N \tag{41}$$

$$due\_carico_n - v_{pk} - M \times \left(2 - x_{ipk} - b_{in}\right) \leq ea_n; \ \forall p \in P, k \in K, h \in H, n \in N \tag{42}$$

$$x_{ipk} \in \{0, 1\}; \ \forall p \in P, k \in K, i \in B \tag{43}$$

$$t\_exec_i \geq 0; \forall i \in B \tag{44}$$

$$ta_n \geq 0; \quad \forall n \in N \tag{45}$$

$$ea_n \geq 0; \quad \forall n \in N \tag{46}$$

$$u_{pk} \geq 0; \quad \forall p \in P, k \in K \tag{47}$$

$$v_{pk} \geq 0; \quad \forall p \in P, k \in K \tag{48}$$

Equation (33) is the objective function to be minimised and is the same as that used in the previous model. However, compared to the previous model, this model does not contain the picking routing constraints to reduce its complexity.

4.3. Heuristic Solving Approach

Ref. [37] showed that the problem of minimising the total tardiness for a set of independent tasks on a single machine is NP-hard. This problem can be interpreted as a special case of the order batching and sequencing problem, in which the capacity of the picking device is equal to only one customer's order. Therefore, it is possible to conclude that the order batching and sequencing problem is also NP-hard and using a heuristics approach is advantageous [28].

As both previously implemented models are complex and, therefore, were not solvable within acceptable time limits, this study developed a constructive heuristic algorithm. Constructive algorithms use common sense rules to solve the problem quickly but without the certainty of reaching the globally optimal solution. Various heuristic solution algorithms have been proposed for the order batching and sequencing problem. Such algorithms can be categorized into four groups: priority rule-based, seed, saving, and metaheuristic algorithms. The first three groups use constructive approaches, whereas metaheuristic algorithms focus on improving given solutions. Metaheuristic algorithms have been shown to significantly improve the results obtained using constructive algorithms through several numerical experiments [38].

This study used an approach based on order sequencing according to the earliest due date rule to generate the initial solution, i.e., the orders with the nearest due date are assigned first. Therefore, the proposed algorithm is priority-based, i.e., it assigns orders to the batches and the operator's position one at a time. Priority rule-based algorithms have a two-step procedure: first, priorities are assigned to customer orders, and second, these orders are assigned to different batches. This procedure ensures that the capacity constraint is not violated.

In the pseudo-code, U represents the set of orders not yet assigned to a picker; k_p is the sequence position of the operator p to whom the next batch can be assigned to; B_{pk_p} represents the set of orders included in the batch assigned to the position k_p of operator p; C_p and v_p are the numbers of items contained in the batch under consideration and the completion time of the batch, respectively. The execution time of the current batch is calculated as $\beta^{time} \times \left(TSP\left(B_{pk_p}\right)\right)$, where TSP indicates the call to the heuristic method for the calculation of the routing strategy of the batch under consideration and β^{time} is the time required to travel a unit of space. Initially, all the orders belong to the vector U because they have not yet been assigned. Then, the orders are successively assigned to specific batches, starting from the order n^* not yet assigned with the earliest due date. First, this study assigns the current order to each operator p and calculates the completion time $\widetilde{v}_p$ for each. The assignment consists of adding the order n^* to the batch B_{pk_p} if the capacity constraint is not violated; otherwise, the assignment includes opening a new batch containing n^*. Second, the algorithm decides which operator p^* the order under consideration is assigned to. This order is assigned to the operator who has generated the minimum completion time in the previous step. Third, the algorithm evaluates the completion times obtained for each picker after the assignment of n^* and selects the one with the smallest value. Fourth, the order n^* is assigned to the corresponding picker considering the capacity constraint. The algorithm ends when all the orders have been assigned to a batch. Algorithm 1 shows the procedure's pseudo-code, algorithm principle, input, and output.

Algorithm 1 Pseudo-code.

Algorithm principle: Assign the orders individually, starting with the ones with the closest due date. The current order is assigned to the operator who is least loaded.
Input: set of N customer orders sorted in ascending order of due dates; the number of items requested c_n $(n \in N)$ for each order $n \in N$; set of operators P; capacity of the pickup vehicle $Cap_{vehicle}$.
Output: heuristic value z^*, which represents the objective function to be minimised, and the corresponding values of total travel time, tardiness, and earliness.
$U := set\ of\ N\ customer\ orders\ sorted\ in\ ascending\ order\ of\ due\ dates$;
for p ∈ P **do**
    $k_p = 1;\ B_{pk_p} = \varnothing; C_p := 0; v_p = 0;$
end for
while $U \neq \varnothing$ **do**
    $n^* = argmin\{due\_carico_n \mid n \in U\};$
    for $p \in P$ **do**
        if $C_p + c_{n^*} \leq Cap_{vehicle}$
            then $\widetilde{v}_p = v_p + \beta^{time} \times \left(TSP\left(B_{pk_p} \cup \{n^*\}\right)\right);$
        else
$\widetilde{v}_p = v_p + \beta^{time} \times (TSP(\{n^*\}));$
        end if
    end for
$p^* = argmin\ \{\widetilde{v}_p \mid p \in P\};$
$U = U \backslash \{n^*\}; v_{p^*} = \widetilde{v}_{p^*};$
if $C_{p^*} + c_{n^*} \leq Cap_{vehicle}$
then $B_{p^*k_{p^*}} = B_{p^*k_{p^*}} \cup \{n^*\}; C_{p^*} = C_{p^*} + c_{n^*};$
else
$k_{p^*} = k_{p^*} + 1; B_{p^*k_{p^*}} = \{n^*\}; C_{p^*} = c_{n^*};$
end if
end while

The heuristic algorithm proposed does not guarantee the identification of the optimal solution but runs the lowest possible risk of having delays. The initial solution was improved by applying different types of local search. A multi-start approach was implemented to evaluate the largest number of possible solutions. The goal was to generate a random sequence at each iteration and, based on this sequence, find the solution; for each execution, a different objective function value is obtained. By comparing the different results, the best order, i.e., the order that generates the lowest value of z, was identified. This study implemented a local search swap and insert move approach, which improved the heuristic solution produced by applying small changes that led to local improvements. Two types of local search exist, namely, the first improvement and best improvement, which define the end criterion of the algorithm. With the first improvement, when the algorithm reaches a local minimum, it stops and returns this minimum as the output value. With the best improvement, the algorithm does not stop when it reaches a local minimum but continues the search until the end criterion is satisfied.

In this study, the following approaches were developed:

1. Swap—first improvement: exchanges two orders of the input sequence and verifies the solution. Once the algorithm identifies a sequence that generates a better objective function, it stops and releases the sequence as the output.
2. Insert move—best improvement: inserts each order in each position, and once it finds an improvement, it returns the sequence as the output.

This study implemented the iterated local search algorithm to increase the objective function's value generated by the heuristic algorithm. The basic principle of this approach was to originate a sequence of possible solutions starting from a generic one. The solutions were obtained by perturbing the current solution and implementing a local search procedure. An acceptance criterion was applied to decide which solution among the candidates to consider in the next step. This study used a Markovian acceptance criterion, which allows an extreme intensification because it accepts s'^* if, and only if, $f(s'^*) < f(s^*)$ [37]. Algorithm 2 shows the local search procedure.

Algorithm 2 The local search procedure.

procedure *IteratedLocalSearch*
  $s_0 = GenerateInitialSolution$
  $s^* = LocalSearch(s_0)$ %*Optional*
  repeat
    $s' = Perturbation\ (s^*)$
    $s^{*'} = LocalSearch\ (s')$
    $s^* = AcceptanceCriterion\ \left(s^*,\ s^{*'}\right)$
  until *the termination condition met*
end

The heuristic earliest due date generated the initial solution in this approach, and a multi-start algorithm performed the first local search. The first solution was perturbed through a swap move perturbation algorithm and improved through a local search insert move. As the acceptance criterion was Markovian, the solution generated was accepted only if it was better than that generated with the multi-start algorithm. If this was the case, the value of the best result found was inserted, and the number of iterations without improvement was reset. At the beginning of the procedure, it was necessary to set the maximum number of cycles without improvement. The cycle stopped when no more solutions existed for the consecutive number of cycles set. Algorithm 3 shows the earliest due date procedure.

Algorithm 3 The earliest due date procedure.

procedure *IteratedLocalSearch*
  $s_0 = Earliest\ Due\ Date\ Algorithm$
  $s^* = Multi - Start\ (s_0)$
  repeat
    $s' = Swap\ Move\ Perturbation\ (s^*)$
    $s^{*'} = Local\ Search\ Insert\ Move\ (s')$
    $s^* = AcceptanceCriterion\ \left(s^*,\ s^{*'}\right)$
  until $Iteration\ without\ improvement > Max\ Interaction\ without\ improvement$
end

Finally, this study implemented an iterated local search multiple perturbation. This approach had the same acceptance characteristics as the previous one, with the only difference being that it chooses whether to perform an insert move perturbation or a swap perturbation randomly.

5. Numerical Experiments

5.1. Implementation of Models

This study designed a simple problem that could be solved within acceptable time limits and allowed us to understand how the results were generated. As mentioned above, the order picking time comprises the items' travel time, search and pickup time, and setup time. This study used the same parameters of the test phase of the algorithm presented in [29] to assign realistic values to the parameters βtime, βsetup, and βpicking. It was estimated that an operator travels 20 units of length per minute, which corresponds to 3 s per unit of length. Furthermore, it was estimated that it takes 10 s to search and retrieve an item from the storage location and that each batch requires a setup time of 3 min (180 s). The capacity of the picking vehicle was assumed to be 20 items. As inputs, both models need the characteristics of each order: the due date, the number of items requested, and the picking locations that need to be visited.

It is important to note that all the time and distance values input had units of seconds and meters, respectively. However, the specific units used can be changed if they are consistent. These examples demonstrate the model's effectiveness and show how consistent solutions can be generated by assigning more or less priority to the different factors. Another emergent factor is that even slight variations can lead to significant changes to the solution in the case of more critical input data.

5.2. Heuristic Implementation

This study considered empirical data to test the proposed heuristic algorithm's correctness. For clarity, it was decided to initially test only a tiny portion of this data. The steps taken to manage and implement the data were as follows. First, to simplify the extensive shipping database, this study selected only the necessary data, i.e., the name of the order, picking locations, due date, and quantities needed in each order. It was essential to know the distances that separate all the storage locations to minimise the completion time and evaluate the distances travelled. As this information is not currently available within the company considered and the number of picking locations was too high to allow manual measurement (there are approximately 550 total spaces in the picking warehouse), this study implemented an algorithm on the resolution software IVE Xpress to calculate the distance matrix automatically. As the code assigned to each location was not random, it can be used to identify the actual position of the location in the warehouse. The basic idea of the algorithm was to exploit this knowledge to calculate distances: the first three letters of the code represent the specific aisle, and the following two numbers identify the position of the location inside the aisle. All this information allows the determination of the precise picking location.

The warehouse layout was used to calculate the distances between the picking locations. As the warehouse was developed from successive enlargements and in compliance with strong spatial constraints, its layout is complex and non-linear. The layout can be described as the union of two different blocks with a longitudinal layout. It consists of several parallel picking aisles with storage locations on both sides and a single central aisle (transverse aisle) without storage locations that enable order pickers to enter or exit a picking aisle.

Another data input not directly available is the matrix a, which is related to the picking location that each order must visit; it is defined as a $num\_order \times num\_location$ matrix with elements a_{ij} equal to 1 if order I visits compartment j and 0 otherwise. An algorithm was developed to calculate this matrix automatically, starting from orders and the corresponding locations. The algorithm used the data derived from the database as the input. However, one order was repeated several times. Each replica was associated with the value of a picking location where in the storage a different article was requested. Starting from the procedure just described, the matrix was constructed iteratively: given two vectors containing the orders and the non-duplicated spaces, the matrix was defined by setting a_{ij} to 1 if the order i required a visit to location j and 0 otherwise.

This study evaluated the bulk orders in the first week of July 2019, which represents a critical period for the organisation owing to the seasonality of the products. To evaluate the implementation of the batching strategy, this study compared the times obtained with the algorithms with those obtained with the strategy commonly implemented in the warehouse, i.e., the single order strategy. With this objective, it was decided to compare the time required to pick orders, assuming that only one operator was involved in picking individual orders. Table 2 lists the results of the case study.

Table 2. Travel time with different strategies.

Date	Travel Time (min)		Difference
	Single Order Strategy	Batching Strategy	
1 July	118.38	88.77	29.61
2 July	152.74	102.39	50.35
3 July	168.07	127.90	40.17
4 July	160.97	144.73	16.24
5 July	202.17	144.68	57.49
Average	160.47	121.69	38.77 (24%)

With the batching strategy, the average travel time was 38.77 min/d. The variability in the different days was due to the different levels of order overlap, which strongly affected the time savings obtained through the batching.

In the evaluation, this study only considered the travel time component, i.e., the time required to reach each picking location. Next, this study evaluated the setup times expected to equal 180 s. As the setup time is involved every time the picker starts a new pick list, the overall setup time in a day under the single order strategy could be calculated as the fixed setup time for the number of orders. In contrast, in the batching strategy, the time required for this activity could be calculated as the product of the fixed time and the batch number. Therefore, with the single order strategy, the average setup time was 201.00 min/d, while with batching strategy, it decreased to 31.80 min/d, which represents an average daily time saving of 169.20 min. Table 3 shows the setup time for different strategies. Since the earliest due date heuristic sequence is in ascending order according to the due date (i.e., it considers one order at a time from the one with the closest due date up to the one that represents the least criticality, and already takes into account the delays), we chose to assign the maximum weight (1) to the completion time. However, the proposed model can be applied with different weights.

Table 3. Setup time with different strategies.

Date	Total Orders	Total Batches	Setup Time (min)		Difference
			Single Order Strategy	Batching Strategy	
1 July	50	8	150	24	126
2 July	71	7	213	21	192
3 July	74	12	222	36	186
4 July	71	12	213	36	177
5 July	69	14	207	42	165
Average	67.0	10.6	201.0	31.8	169.2 (84%)

The picking time was constant on average because the number of items to be picked did not change. Therefore, the average time saved by applying the batching strategy was 207.97 min/d or approximately 3.47 h/d. Table 4 shows the total time for different strategies.

However, the batching strategy requires a final sorting activity, dividing the batch into individual customer orders. Some choices were made in the model definition phase to reduce the time and space required for this activity. First, each order was assumed to belong to only one batch, i.e., there was no fractioning; moreover, the capacity constraint guaranteed that, on average, each mission contained few orders. However, this activity was still necessary and introduces additional time not present within the single order strategy. As the time required to sort the batch is not known, this study calculated the sorting threshold time that guaranteed a gain in terms of time; if this value was exceeded, there were no time gains. Table 5 shows the sorting time with the batching strategy.

Table 4. Picking total time with different strategies.

Date	Picking Total Time (min)		Difference
	Single Order Strategy	Batching Strategy	
1 July	268.38	112.77	155.61
2 July	365.74	123.39	242.35
3 July	390.07	163.90	226.17
4 July	373.97	180.73	193.24
5 July	409.17	186.68	222.49
Average	361.47	153.49	207.97 (57%)

Table 5. Sorting time with the batching strategy.

Date	Total Difference [min]	Total Batches	Sorting Time for Batch (min)	Average Order/Batch (min)	Sorting Time for Order (min)
1 July	155.61	8	19.45	6.25	3.11
2 July	242.35	7	34.62	8.88	3.90
3 July	226.17	12	18.85	9.25	2.04
4 July	193.24	12	16.10	8.88	1.81
5 July	222.49	14	15.89	8.63	1.84
Average	207.97	10.60	20.98	8.38	2.54

5.3. Discussion on the Implementation

An alternative to the batching strategy is the sort while picking method. In this method, an operator uses a picking cart containing several pallets to place objects in the same order at the same point. The advantage of this strategy is that it does not need a sorting activity after the picking is concluded; instead, the sorting activity takes place simultaneously with the picking. This picking strategy is preferred when handling orders consisting of small items and a relatively small number of pieces.

The results showed that the traditional pick-and-sort batching strategy is advantageous only if the sorting activity requires an average time of less than 20.98 min per batch or less than 2.54 min per order. In this case, the overall time savings strongly depend on the time required for the final sorting activity. On the other hand, a sort while picking batching strategy might be more convenient because the batches contain a limited number of orders and items. Therefore, the traditional pick-and-sort batching strategy might be convenient for picking bulk orders. This way, the sorting time would be zero, with a consequent average daily saving of 207.97 min (57%). Establishing a comparison, refs. [23,24] reported savings between 10 and 25%.

Further research should also focus on a more accurate and structured analysis of the different possible sorting systems, identifying the respective characteristics, advantages, and disadvantages with particular attention to their sorting times. Furthermore, a more accurate feasibility study of sort while picking systems could be carried out, focusing on the need to use picking vehicles of greater length, with particular attention to possible consequences on traffic congestion.

All the heuristic approaches developed were tested to define the solution for each day examined. It was found that the iterated local search algorithm and iterated local search multiple perturbation generated the best solution by four and one times, respectively. Therefore, the local search effectively leads to an improvement in the initial solution. However, specific challenges might emerge during the algorithm's implementation. The primary obstacle involves the uncertainty surrounding the measurement of parameters such as order-related timings. Another complexity lies in the necessity for on-site equipment within the factory premises, accessible to operators. Lastly, should the operation's evolution surpass the algorithm's predefined limits, it would necessitate a fresh implementation.

The overall results of the study have implications.

The advances highlighted by the heuristic can be leveraged by companies developing solutions for logistics companies, either as open innovation initiatives or as software licenses. Stakeholders such as logistics companies, distribution centers, freight forwarders, and users of logistics services can benefit from such an investment by reducing their storage costs, increasing the reliability of delivery times, reducing errors in the quantity and identification of materials, or increasing the flexibility of their operations. All these effects can increase the competitiveness of logistics companies and their customers, usually supply chain managers, in their industries. In particular, due to mathematical requirements, supply chain managers should use the resource as software add-ons to their usual warehouse management systems, which are already operating.

6. Conclusions

This study considered JOBSPRP and tried adapting it to the needs of an organisation that provides logistics services. Two depots were assumed in the definition of the routing strategy that were logically separated but may be physically coincident in practice. Unlike most previous models, with more than proportional growth, the size of the proposed model grew polynomially with the number of orders. To limit the amount of input data, the study developed another model to solve the picking routing problem. This model only needed some of the picking locations of the warehouse as the input, namely, the locations that required at least one visit on a day. Both proposed models could solve minor problems with acceptable computational times. To solve real, large problems, the study developed approaches based on the local search algorithm. The results showed that, for large problems, the iterated local search and iterated local search multiple perturbation algorithms could successfully identify the minimum solution and significantly improve the solution initially obtained with the heuristic earliest due date algorithm.

Future research should focus on picker blocking in real warehouses, which generally need optimal layouts. Picker blocking represents an essential factor that can cause significant inconvenience and slowdowns during picking articles, increasing the picking times and reducing productivity. When traffic congestion is considered, the tours of different operators can no longer be independently determined, making the problem significantly more complex. Furthermore, cases where customer orders are not known in advance but are made dynamically during the day could be considered [31]. Further research should also focus on the impact of picking and batching efficiency on the other steps of the supply chain, such as road and urban transportation [39].

The findings provide room for future advances. Regarding the involved technology, the next step is to develop a software add-on to be implemented in existing warehouse management systems. Furthermore, a parallel decision support system could increase the likelihood of logistics operators meeting order requirements, such as due date, quantity, and quality of the loads, while reducing costs associated with a lack of efficiency. Regarding the implications, a further step is to lead various real-world implementations to measure the gains such an add-on could provide to customers interested in managing their orders' dispatch operations.

Author Contributions: Conceptualization, E.M., F.L. and E.B.; methodology, A.M.C., E.M., F.L. and M.A.S.; software, E.M. and E.B.; validation, F.L. and M.A.S.; formal analysis, E.M. and A.M.C.; investigation, F.L. and E.M.; data curation, A.M.C. and E.B.; writing—original draft preparation A.M.C., F.L. and M.A.S.; writing—review and editing A.M.C. and M.A.S.; supervision F.L.; project administration F.L. and E.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Song, J.S.; van Houtum, G.J.; van Mieghem, J.A. Capacity and Inventory Management: Review, Trends, and Projections. *Manuf. Serv. Oper. Manag.* **2020**, *22*, 36–46. [CrossRef]
2. Zheng, B.; Sun, C. Research on Logistics Warehousing Cost of Z Delivery Enterprise and Its Optimization Strategy. *Front. Bus. Econ. Manag.* **2023**, *8*, 12–22. [CrossRef]
3. Yilmaz, O.F.; Cevikcan, E.; Durmusoglu, M.B. Scheduling Batches in Multi Hybrid Cell Manufacturing System Considering Worker Resources: A Case Study from Pipeline Industry. *Adv. Prod. Eng. Manag.* **2016**, *11*, 192–206. [CrossRef]
4. Gauss, L.; Lacerda, D.P.; Sellitto, M.A. Module-Based Machinery Design: A Method to Support the Design of Modular Machine Families for Reconfigurable Manufacturing Systems. *Int. J. Adv. Manuf. Technol.* **2019**, *102*, 3911–3936. [CrossRef]
5. Lolli, F.; Lodi, F.; Giberti, C.; Coruzzolo, A.M.; Marinello, S. Order Picking Systems: A Queue Model for Dimensioning the Storage Capacity, the Crew of Pickers, and the AGV Fleet. *Math. Probl. Eng.* **2022**, *2022*, 6318659. [CrossRef]
6. Custodio, L.; Machado, R. Flexible Automated Warehouse: A Literature Review and an Innovative Framework. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 533–558. [CrossRef]
7. de Koster, R.; Le-Duc, T.; Roodbergen, K.J. Design and Control of Warehouse Order Picking: A Literature Review. *Eur. J. Oper. Res.* **2007**, *182*, 481–501. [CrossRef]
8. Tompkins, J.A.; White, J.A.; Bozer, Y.A.; Tanchoco, J.M.A. *Facilities Planning*; Wiley: Hoboken, NJ, USA, 2003; ISBN 9780470444047.
9. Yilmaz, O.F.; Oztaysi, B.; Durmusoglu, M.B.; Oner, S.C. Determination of Material Handling Equipment for Lean In-Plant Logistics Using Fuzzy Analytical Network Process Considering Risk Attitudes of the Experts. *Int. J. Ind. Eng. Theory Appl. Pract.* **2017**, *24*. [CrossRef]
10. Lee, M.K.; Scherer, E.; Elsayed, E.A.; Kim, S. Sequencing and Batching Procedures for Minimizing Earliness and Tardiness Penalty of Order Retrievals. *Int. J. Prod. Res.* **1993**, *31*, 727–738. [CrossRef]
11. Elsayed, E.A.; Lee, M.K. Order Processing in Automated Storage/Retrieval Systems with Due Dates. *IIE Trans.* **1996**, *28*, 567–577. [CrossRef]
12. Tsai, C.Y.; Liou, J.J.H.; Huang, T.M. Using a Multiple-GA Method to Solve the Batch Picking Problem: Considering Travel Distance and Order Due Time. *Int. J. Prod. Res.* **2008**, *46*, 6533–6555. [CrossRef]
13. Kulak, O.; Sahin, Y.; Taner, M.E. Joint Order Batching and Picker Routing in Single and Multiple-Cross-Aisle Warehouses Using Cluster-Based Tabu Search Algorithms. *Flex. Serv. Manuf. J.* **2012**, *24*, 52–80. [CrossRef]
14. Henn, S.; Schmid, V. Metaheuristics for Order Batching and Sequencing in Manual Order Picking Systems. *Comput. Ind. Eng.* **2013**, *66*, 338–351. [CrossRef]
15. Grosse, E.H.; Glock, C.H.; Ballester-Ripoll, R. A Simulated Annealing Approach for the Joint Order Batching and Order Picker Routing Problem with Weight Restrictions. *Int. J. Oper. Quant. Manag.* **2014**, *20*, 65–83.
16. Henn, S. Order Batching and Sequencing for the Minimization of the Total Tardiness in Picker-to-Part Warehouses. *Flex. Serv. Manuf. J.* **2015**, *27*, 86–114. [CrossRef]
17. Chen, T.L.; Cheng, C.Y.; Chen, Y.Y.; Chan, L.K. An Efficient Hybrid Algorithm for Integrated Order Batching, Sequencing and Routing Problem. *Int. J. Prod. Econ.* **2015**, *159*, 158–167. [CrossRef]
18. Scholz, A.; Schubert, D.; Wäscher, G. Order Picking with Multiple Pickers and Due Dates—Simultaneous Solution of Order Batching, Batch Assignment and Sequencing, and Picker Routing Problems. *Eur. J. Oper. Res.* **2017**, *263*, 461–478. [CrossRef]
19. Ardjmand, E.; Sanei Bajgiran, O.; Youssef, E. Using List-Based Simulated Annealing and Genetic Algorithm for Order Batching and Picker Routing in Put Wall Based Picking Systems. *Appl. Soft Comput.* **2019**, *75*, 106–119. [CrossRef]
20. Kübler, P.; Glock, C.H.; Bauernhansl, T. A New Iterative Method for Solving the Joint Dynamic Storage Location Assignment, Order Batching and Picker Routing Problem in Manual Picker-to-Parts Warehouses. *Comput. Ind. Eng.* **2020**, *147*, 106645. [CrossRef]
21. Attari, M.Y.N.; Torkayesh, A.E.; Malmir, B.; Jami, E.N. Robust Possibilistic Programming for Joint Order Batching and Picker Routing Problem in Warehouse Management. *Int. J. Prod. Res.* **2020**, *59*, 4434–4452. [CrossRef]
22. Cano, J.A.; Cortés, P.; Campo, E.A.; Correa-Espinal, A.A. Solving the Order Batching and Sequencing Problem with Multiple Pickers: A Grouped Genetic Algorithm. *Int. J. Electr. Comput. Eng. (IJECE)* **2021**, *11*, 2516–2524. [CrossRef]
23. Cano, J.A.; Cortés, P.; Campo, E.A.; Correa-Espinal, A.A. Multi-Objective Grouping Genetic Algorithm for the Joint Order Batching, Batch Assignment, and Sequencing Problem. *Int. J. Manag. Sci. Eng. Manag.* **2021**, *17*, 188–204. [CrossRef]
24. Cergibozan, Ç.; Tasan, A.S. Genetic Algorithm Based Approaches to Solve the Order Batching Problem and a Case Study in a Distribution Center. *J. Intell. Manuf.* **2022**, *33*, 137–149. [CrossRef]
25. Lin, C.C.; Kang, J.R.; Hou, C.C.; Cheng, C.Y. Joint Order Batching and Picker Manhattan Routing Problem. *Comput. Ind. Eng.* **2016**, *95*, 164–174. [CrossRef]
26. Henn, S.; Koch, S.; Wäscher, G. Order Batching in Order Picking Warehouses: A Survey of Solution Approaches. In *Warehousing in the Global Supply Chain*; Springer: London, UK, 2012; pp. 105–137. ISBN 978-1-4471-2273-9.
27. Lu, W.; McFarlane, D.; Giannikas, V.; Zhang, Q. An Algorithm for Dynamic Order-Picking in Warehouse Operations. *Eur. J. Oper. Res.* **2016**, *248*, 107–122. [CrossRef]
28. Valle, C.A.; Beasley, J.E.; da Cunha, A.S. Optimally Solving the Joint Order Batching and Picker Routing Problem. *Eur. J. Oper. Res.* **2017**, *262*, 817–834. [CrossRef]

29. Matusiak, M.; de Koster, R.; Saarinen, J. Utilizing Individual Picker Skills to Improve Order Batching in a Warehouse. *Eur. J. Oper. Res.* **2017**, *263*, 888–899. [CrossRef]
30. Cano, J.A.; Correa-Espinal, A.A.; Gómez-Montoya, R.A. An Evaluation of Picking Routing Policies to Improve Warehouse Efficiency. *Int. J. Ind. Eng. Manag.* **2017**, *8*, 229–238. [CrossRef]
31. Azadnia, A.H.; Taheri, S.; Ghadimi, P.; Mat Saman, M.Z.; Wong, K.Y. Order Batching in Warehouses by Minimizing Total Tardiness: A Hybrid Approach of Weighted Association Rule Mining and Genetic Algorithms. *Sci. World J.* **2013**, *2013*, 246578. [CrossRef]
32. Scholz, A.; Wäscher, G. Order Batching and Picker Routing in Manual Order Picking Systems: The Benefits of Integrated Routing. *Cent. Eur. J. Oper. Res.* **2017**, *25*, 491–520. [CrossRef]
33. Bozer, Y.A.; Kile, J.W. Order Batching in Walk-and-Pick Order Picking Systems. *Int. J. Prod. Res.* **2008**, *46*, 1887–1909. [CrossRef]
34. Cano, J.A.; Correa-Espinal, A.A.; Gómez-Montoya, R.A. Mathematical Programming Modeling for Joint Order Batching, Sequencing and Picker Routing Problems in Manual Order Picking Systems. *J. King Saud Univ. Eng. Sci.* **2020**, *32*, 219–228. [CrossRef]
35. Gademann, N.; van de Velde, S. Order Batching to Minimize Total Travel Time in a Parallel-Aisle Warehouse. *IIE Trans.* **2007**, *37*, 63–75. [CrossRef]
36. Du, J.; Leung, J.Y.-T. Minimizing Total Tardiness on One Machine Is NP-Hard. *Math. Oper. Res.* **1990**, *15*, 483–495. [CrossRef]
37. Henn, S.; Koch, S.; Doerner, K.F.; Strauss, C.; Wäscher, G. Metaheuristics for the Order Batching Problem in Manual Order Picking Systems. *Bus. Res.* **2010**, *3*, 82–105. [CrossRef]
38. Lourenço, H.R.; Martin, O.C.; Stützle, T. Iterated Local Search. In *Handbook of Metaheuristics*; Springer: Boston, MA, USA, 2003; pp. 320–353. [CrossRef]
39. Sellitto, M.A.; Borchardt, M.; Pereira, G.M.; Gomes, L.P. Environmental performance assessment of a provider of logistical services in an industrial supply chain. *Theor. Found. Chem. Eng.* **2012**, *46*, 691–703. [CrossRef]