Volume 39(1), pp. 45 – 64
http://orion.journals.ac.za

ORiON
ISSN 0259–191X (print)
ISSN 2224–0004 (online)

Route overlap metrics to batch orders

FM Hofmann[*] and SE Visagie[†]

Received: 8 August 2022; Revised: 31 January 2023; Accepted: 31 January 2023

Abstract

In this study different metrics to perform order batching for a unidirectional cyclical picking system are investigated. Batches must be formed to minimise walking distance that is measured as the total number of cycles traversed by pickers. Route overlap metrics are developed to approximate walking distance before picker routing. Thereby, information about route similarity is added to identify orders that are compatible for batching. A span is the length of a section of picking line that a picker walks to collect the items in an order. Three metrics based on the concept of a span are developed, namely the stops non-identical spans, the non-identical stops-spans and the stops-spans ratio metric. Numerical experiments are carried out to determine the best combination between route overlap metric and batching method. A prominent South African retailer's distribution centre provides 50 sample picking waves for an experimental set up with real-life data. The stops non-identical spans metric used together with greedy smallest entry heuristic to perform the batching consistently leads to a low number of cycles traversed to complete a picking wave relative to the other methods.

Key words: Order batching, route overlap metrics for batching, assignment problems, heuristics, cyclical picking line

1 Background and introduction

A distribution centre (DC) matches changing demand with reacting supply. Various factors such as product seasonality, shorter product life cycles, quicker response times to the market and transport networks with product consolidation can cause discrepancies between supply and demand [32]. DCs are unlikely to vanish from supply chains, even though they require space, labour, equipment and information resources. A DC renders a service to other logistical activities and thus depends on them. It plays an important role

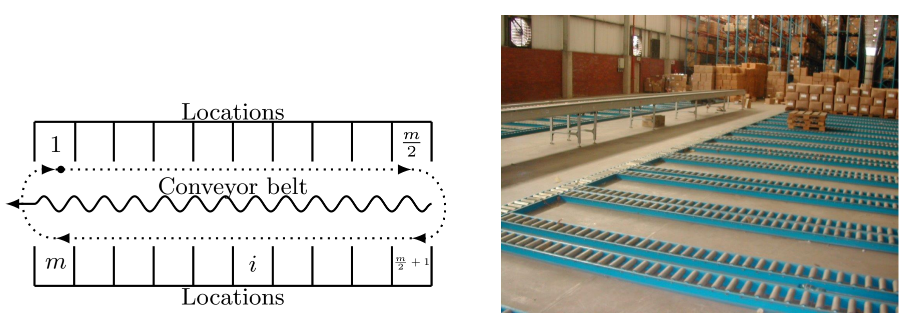
[*]Department of Logistics, Stellenbosch University, South Africa, email:20304269@sun.ac.za
[†]Corresponding author:(Fellow of the Operations Research Society of South Africa) Department of Logistics, Stellenbosch University, South Africa, email:svisagie@sun.ac.za

http://dx.doi.org/10.5784/39-1-745

with regards to increasing customer service and lowering transport cost [10]. It collects products from a number of suppliers for delivery to different end customers. At the DC bulk supplies are turned into customer orders [35].

According to Rouwenhorst *et al.* [32] the four main activities in a DC are receiving, storage, order picking, and shipping. Accounting for between 2% to 5% of a companies' cost of sales, warehousing is expensive. Therefore, DCs always strive to minimise cost. The extend to which this cost can be minimised is countered by the emphasis on customer service [10]. The cost of order picking, the basic service of a DC, contribute 50% to 65% of the cost of operating a DC [35]. The range in product portfolio and size diversity results in most DCs operating manual order picking systems. Manual picking results in high labour cost [14]. The process of responding to a specific customer request by retrieving items from buffer or storage areas describes order picking [7]. Setting up an order picking system includes the average size of an order and the number of orders in combination with the order items picked during a day [5].

In this study the picking system in a real-life distribution centre (DC) is investigated. A large South African retailer (the Retailer) owns this DC. On a daily basis its DC handles an variety of orders for a large number of stores located across South Africa. The Retailer's operations' key characteristic is its central inventory planning at store level. Store managers do not request stock directly, but the central planners assign stock keeping units (SKUs) that are available in the DC to the stores. After SKUs arrived at the DC, planners determine the number of items per SKU needed at each store. These instructions are then issued to the DC. A number of SKUs is selected by a picking line manager. These are arranged on a single picking line for picking. The process of selecting SKUs, putting them on a picking line, performing the picking and removing excess stock from the picking line is called a picking wave. The number of items that should be picked for a single store over all SKUs that are in a wave constitutes an *order*. For each wave, one unique SKU is located at one unique location in the picking line [23]. This set up implies that all orders are known when a wave of picking starst and only exist for that wave of picking.



(a) A picking line is represented with m locations. Source: De Villiers [8].

(b) Empty picking line at the DC.

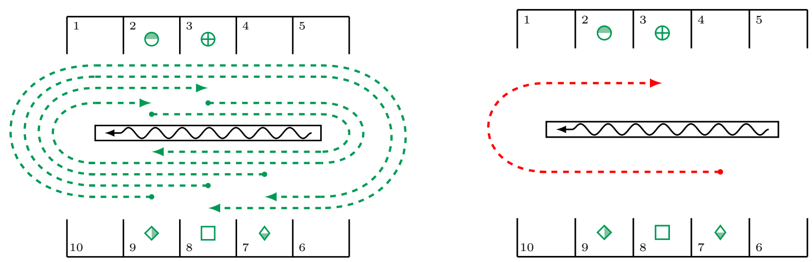
Figure 1: *The layout of a unidirectional cyclical picking line.*

In Figure 1(a) the layout of a picking line containing m locations is presented, and in Figure 1(b) a photo of a physical picking line without any SKUs is shown. At the DC of the Retailer, a picking line is made up of 56 locations. Between the two rows of locations runs a conveyor belt which has two gates for access to the other side. During the picking wave each SKU from the storage area has a unique picking line location. Before the picking wave starts the number of items per SKU that should go to each store is known. Up to five pallets of items of an identical SKU can be stored at each location (on the rollers depicted in Figure b). On the floor between picking lines additional SKUs can be kept in easy supply preventing stock outs during a wave of picking. Walking clockwise around the conveyor belt, pickers pick the orders for that wave. A voice recognition system (VRS) guides them through the process and sends them to the next (closest) required SKU. Pickers walk clockwise and pick the next SKU for the current order. This picking system can be classified as a picker-to-parts system. An empty carton is registered with the VRS before an order starts. When the carton is full or the order is completed, the cartons are moved to further processing areas via the conveyor belt [21, 22].

Matthews & Visagie [23] proposed that the *total number of cycles* that all pickers combined traverse during a wave of picking as an appropriate and useful travel distance measure. These cycles are formed by the distance travelled during picking individual orders and distance travelled to link these orders - it is the distance travelled between the last SKU of one order and the first SKU of a next order. The measurement can compare different methodologies that will be introduced to reduce walking distance in terms of their effectiveness.

Matthews & Visagie [23] defined a *span* as the distance needed to collect all order items, given a location starting point. For order i, with closest ending position e_i^a, given starting location a, $P_i^a = \langle a, e_i^a \rangle$ describes the span. The shortest span for every order, also called the minimum span, could be identified by finding the largest opening (or gap) between any two items within an order. The minimum span starts at the end of this gap, while the end of the minimum span is the gap's starting location. An example is illustrated in Figure 2 with a picking line consisting of 10 locations. In this example the SKUs placed in locations $2, 3, 7, 8$ and 9 are requested by Order 1. The spans of this order can start at any location containing one of the SKUs in the order. This is depicted in Figure 2(a). For example, there is a gap that covers locations 4 to 6, while another gap goes from location 10 to 1. The first gap, covering 3 location is the largest gap in this example. The minimum span to collect all of the needed SKUs corresponds with this biggest gap and starts at location 7 and runs all the way to location 3. In Figure 2(b) the minimum span is illustrated by the red dashed arrow. This minimum span's length is 7 locations, denoted by $|P_1^{\min}| = \langle 7, 3 \rangle$.

The layout is comparable to a unidirectional carousel system if the picker remains static while the SKUs are moving. Usually small and medium products are picked using automated carousel warehouse systems. A rotatable shelving circuit which holds multiple SKUs comprises a carousel. Throughout the picking process, a picker remains at one location while operating the system. This picking system may be categorised as a parts-to-picker system. A bidirectional carousel is a carousel that can rotate in both directions, while a unidirectional carousel only rotates one way [1]. The unidirectional cyclical picking line resembles a unidirectional carousel, but a major difference between the two systems



(a) Order 1's potential spans.

(b) Order 1's minimum span, described by $P_1^{\min} = \langle 7, 3 \rangle$.

Figure 2: *All spans of an order are schematically represented on a picking line. A SKU is specified by its shape.*

is the existence of wave picking in the one that results in deterministic orders for a wave of picking.

Optimisation approaches in carousel systems in literature derive the mix of SKUs from historical data [20]. Therefore, the aspect that orders are deterministic in the unidirectional cyclical picking system is unique in literature [23]. Bidirectional movement, as it is frequently implemented in industry for carousel systems, cannot be used in the unidirectional picking line because the minimum span between orders may overlap and thus increase congestion. Furthermore, known optimal picking strategies for bidirectional picking lines are not optimal in the unidirectional case [20, 21].

Planning a wave of order picking includes the order batching problem (OBP). In a picking wave, the SKUs in that wave determine the orders. All orders (defined by that wave) must be picked during the wave defining them [13]. Orders can be picked as single orders or can be batched. In single order picking a picker picks one order at a time, while in batch picking a picker picks multiple orders simultaneously. Individual orders within a batch may be sorted after the batch picking process, or orders may be sorted-while-picked. The total picking time is reduced by batching, since multiple orders are picked simultaneously. A reduction in labour cost can be achieved by this decrease in picking time [6]. In some instances the capacity of the picking device limits the number of orders that can be batched [36]. An additional sorting effort prohibits the splitting of orders when batched [15].

According to Nicolas *et al.* [27] few articles have been published on the order batching problem in automated carousel systems as compared to single-block warehouses. Order batching was introduced by Hofmann & Visagie [19] to the layout of a unidirectional cyclical picking line. Walking distance was approximated before picker routing by the development of picking location metrics. However, the locations a picker passes to get to the next required SKU is excluded in these metrics that are based on picking locations. Including more information about each order by adding spans to the metrics aims at overcoming this limitation. The aim of this paper is thus to identify the best span-based metric that reduces the total distance travelled by pickers during a picking wave.

Section 2 contains a review of the literature on order batching in single-block parallel aisle and automated storage and retrieval warehouses. The route overlap metrics are developed by combining batches using greedy heuristics in Section 3. In Section 4 real-life data provided by the Retailer is used to test the route overlap metrics' performance. It is compared with the best performing picking location metric suggested by Hofmann & Visagie [19] to identify the best method. The study concludes in Section 5. It summarises the findings and gives possible ideas for further research opportunities.

2 Literature review

The order batching problem for the layout of a parallel aisle single-block warehouse was formulated by Gademann & Velde [11], who also proved it to be NP-hard in the strong sense. Nevertheless, for batches containing not more than two orders, the problem can be solved in polynomial time. A time-consuming preprocessing step determining all batches that are feasible is a requirement of the solution approach of Henn & Wäscher [16] solving test instances of up to 40 orders. Muter & Öncan [26] solved batching problems with a maximum of 100 orders by means of a column-generation model that was specially tailored for their problem formulation.

Nicolas *et al.* [27] introduced the OBP for a vertical carousel system by formulating it as a mixed integer linear programming problem (MILP). However, the MILP could not solve test instances over 50 orders. They imposed a maximum solution time of 30 minutes on the MILP.

In this study the number of orders to be batched far exceeds these maximum sizes of about 100 orders. At the Retailer, a picking line can serve more than 1 500 orders in a single picking wave. This leads to the exploration of faster approaches to determine good batches of orders. Seed and saving algorithms have been suggested to help solving large OBP instances.

In the seed approach, batches are first initiated, then orders are allocated to batches, and finally terminated by fulfilling a stopping criterion. The overall objective remains the minimisation of the total walking distance for pickers to collect all orders. In the standard single-block parallel aisle warehouse, De Koster *et al.* [6] evaluated various seed algorithms and proposed the farthest storage location seed selection rule. Furthermore, Ho & Tseng [18] investigated several location- and aisle-based seed algorithms. Extending this study, they proposed the smallest number of common picking locations, the smallest number of overlapping picking aisles as well as the smallest rectangular covering area by the batched orders as selection rules [17]. Pan & Liu [28] evaluated four rules to select an initial seed as well as four rules to add orders. This OBP was analysed in an automated storage and retrieval system.

Saving algorithms, that are related to the Clarke and Wright-algorithm (CW) [4], are also used to solve the OBP. This approach compares collecting orders together in one route to collecting orders individually. It attempts to save picking time [6]. The minimum additional aisles heuristic begins by computing a savings score for each order pair's additional aisles and was proposed by Rosenwein [31]. Compared to Gibson & Sharp [12] their algorithm generates fewer and shorter picking tours.

The first OBP version of the Clarke and Wright-algorithm computing savings for each combination of orders is described by De Koster *et al.* [6]. In the second version, the savings are recalculated every time a novel batch combination has been identified. Bozer & Kile [2] further improved on this approach by including a normalised value of time saving for every order pair. In the third version, the initial savings matrix is adapted every time an order has been allocated [6]. Elsayed & Unal [9] proposed a small and large algorithm which depending on values that are predefined classifies orders as small or large before batch allocation.

In general, saving algorithms are more complex and require more computational time, because of the dynamic batch comparison. The unidirectional cyclical picking implemented by the Retailer differs from the layouts investigated in literature and thus these methods cannot be used directly.

The order-to-route closeness metric is central to almost all order batching heuristics [13]. Hofmann & Visagie [19] introduced the OBP to the unidirectional cyclical picking line layout of the Retailer. They developed three picking location-based (also called stop-based) order-to-route closeness metrics with the objective of minimising the walking distance of pickers. The stops, non-identical stops and stops ratio metrics generate batches using exact solution approaches, greedy heuristics and metaheuristics. Using the stops ratio metric together with the greedy random heuristic produces the shortest total walking distance and computational time. However, the information about which locations are passed without picking from them is not included. This study extends on location based metrics to include more picker route information in determining good batches.

3 Batching metrics

In this paper the same picking system that was used in Matthews & Visagie [23] and Hofmann & Visagie [19] is investigated. The main characteristics are re-emphasised for completeness. The proposed batching metrics incorporate information about spans of orders and combine the different pieces of information to arrive at new heuristics to find good batches.

The Retailer's DC has 12 independent picking lines that are picking in a unidirectional cyclical fashion. All are operated in the same way. Therefore, one line is examined in this studies' remainder as it can be applied to all picking lines in the DC.

Modelling the Retailer's picking system incorporates the following assumptions.

1. An order's SKUs and locations picked during a wave are fixed *a priori*.
2. Due to the trolley's capacity and the size of the SKUs that must be picked, the size of a batch is limited to two orders per batch.
3. Before a next batch can commence, a picker must finish the entire batch being processed at the moment.
4. Pickers can pass each other because there is enough space to do so. Therefore, picker speed is assumed to be constant.

5. Each batch starts and ends at the locations defined by the orders within that batch.

An order is completed by a picker. Each picker picks one order at a time. Matthews & Visagie [23] proposed the nearest end heuristic to form an order sequence that minimises pickers' travel distance over a complete wave. The heuristic selects the order with the nearest or closest ending location from the pickers current location to pick next in the order pick sequence.

Consider the example picking line in Table 1. It consists of 10 locations, 10 SKUs and 4 orders. Each order's locations that should be visited by a picker is given per row.

Locations:	1	2	3	4	5	6	7	8	9	10
Order 1		◓	⊕				◈	□	◈	
Order 2				⊗	△		◈	□		
Order 3			⊕		△	◇	◈	□		⬠
Order 4	○	◓			△	◇	◈			

Table 1: *Order locations on the small picking line example.*

The green, blue, red and yellow colours depict four different orders or stores. SKUs in a location are represented by the different shapes. In Figure 3 a solution by the nearest end heuristic is given for this example. A picker would start with Order 4 (yellow), which is picked from Locations 1 to 7. The next order to pick is Order 1 (green). This is followed by Order 2 (blue). The picker would finish with Order 3 (red). Thus using the nearest end heuristic, a picker would cover four cycles to complete this wave. The effectiveness with regards to cycles traversed of various batching ideas is tested using the nearest end heuristic, since it is shown to be fast and provide good solutions [23].

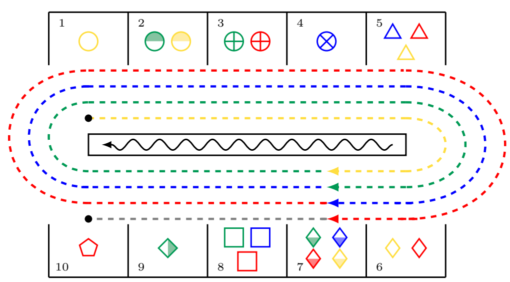


Figure 3: *A solution to the small example given in Table 1. Unique SKUs are represented by shapes, while colours correspond with the orders.*

The easiest way to introduce order batching would be to employ a first-in-first-out approach (FIFO). The first entries in an order list are grouped until a predetermined maximum in batch size is met [12]. The FIFO rule serves as a benchmark to compare various batching metrics. Order 1 and 2 would form Batch 1 and the last two orders would make up Batch 2, if the FIFO rule with a maximum batch size of two would be applied. In Figure 4 the two batches are illustrated in orange and purple respectively. The picker movement is represented by dashed lines. For this example, only two cycles are needed to pick the four orders if batching is allowed – a distance saving of 50%. Various batching approaches to further decrease walking distance will thus be introduced and evaluated.

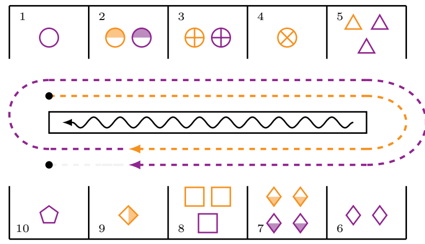


Figure 4: *Small example of order batching using FIFO on a unidirectional picking line.*

Including minimum span information into metrics that approximate distance before routing to minimise pickers' walking distance is the focus of this study. Comparing the minimum spans can be used to evaluate similarities between orders, since the minimum span of an order is specific to that order providing the shortest route to collect all items. In Figure 3 for example, Order 4's minimum span is $P_4^1 = \langle 1, 7\rangle$. The span's length is $|P_4^{\min}| = 7$ locations – this is the same span as obtained by the nearest end heuristic. The next order now has to begin at Location 8. This gives a span of $P_4^8 = \langle 8, 7\rangle$ with a length of $|P_4^8| = 10$ locations. This is not the minimum span, which is $P_1^7 = \langle 7, 3\rangle$ with a length of 7 locations. Depending on their starting locations, orders can be collected in different spans, but the span's length never exceeds one cycle. Unfortunately this information (the most favourable span to pick) will only become known after a picker's route has been established, and batching needs to be solved before the routing problem can be solved. Therefore, the proposed order-to-route closeness metrics incorporates the locations that have to be passed to arrive at a closeness measure between orders for batching. This information is derived from the minimum span of an order.

The solution approaches suggested by Hofmann & Visagie [19] to combine orders into batches before routing will be applied to the route overlap metrics proposed. The best results with focus on solution quality and computational time was provided by two greedy heuristics, namely the greedy random heuristic (GR) and the greedy smallest entry heuristic (GS) [19]. Both these heuristics use a matrix M. The general element m_{ij} contains the selected closeness value for orders i and j as determined by the batching metric.

The only difference between the GR and GS heuristic is the working of their search mechanisms as shown in Algorithm 1. GR uses a random permutation of the elements in M to batch orders. On the other hand, GS searches the smallest and second smallest entry in each row of M. The smallest entry of the row with the biggest difference between the smallest and second smallest entry is then selected. The two orders corresponding to this entry is then used to be batched next. If this minimum entry is m_{kj}, the tuple (k, j) is appended to set $\mathcal{G}$ specifying the batching of orders k and j. Row k and column j are then both deleted from M. This process of batching two orders is repeated until set $\mathcal{G}$ has the cardinality of $n/2$, where n is the number of orders to be batched.

Algorithm 1 (Greedy heuristic)

Input: *M with entries m_{ij} calculated by a batching metric for all orders i and j, a solution set $\mathcal{G}$ that is empty*

Output: *Solution set $\mathcal{G}$ presented as batched orders list*

1: $\mathcal{G} \leftarrow \emptyset$

2: **while** $|\,\mathcal{G}\,| < n/2$ **do**

  $k \leftarrow$ *GR or GS's search mechanism*
  $m_{kq} = \min\limits_{j}[m_{kj}]$
  $\mathcal{G} \leftarrow \mathcal{G} \cup (k, q)$
  Remove row and column k and q from M
end while
Return $\mathcal{G}$

The aim of the proposed metrics to batch orders is to minimise incompatibility between orders. This is achieved by minimising walking distance before picker routing. A representative test set of historical data from the Retailer's DC that is publicly available [24] is used to identify the best combination of batching metric and solution approach with regards to total number of cycles traversed in the system. The results are reported on 50 sample picking waves that have been grouped into (a) more than 1 000 orders – the large set, (b) 100 – 1 000 orders – the medium set, and (c) less than 100 orders – the small set. Daily, the seasonal product portfolio and size variety leads to a set of non-uniform orders that the DC has to process. Each batching metric is tested in an experimental set up. It measures the total number of cycles that pickers have to walk over 50 picking waves to determine the best batching metric.

The experiments were implemented in Python 3.6 [29]. It was executed on a Dell Optiplex 5050 running the Microsoft Windows 10 Enterprise 2016 LTSB operating system [25]. The computer has 3.6 GHz Intel Core i7-7700 CPU with a 1x8GB 2400MHz DDR4 RAM. Results were analysed in R [30].

Based on the minimum span definition, batching metrics are developed and compared to the best performing picking location metric proposed by Hofmann & Visagie [19] in the next sections. According to Hofmann & Visagie [19] the best performing picking location metric is the stops ratio and greedy random (R-GR) combination. This combination will be used for evaluation/comparison purposes in this study. Developing spans metrics is the logical extension from stops metrics. Spans metrics lead to combination metrics which result in route overlap addition metrics.

3.1 Spans metrics

Three span-based metrics are developed first. Denote the minimum span of order i as $P_i^{\min}$. Let $P_i^{\min} \Cap P_j^{\min}$ be the overlap in number of locations between the minimum spans of orders i and j. Additionally, the biggest gap of order i is denoted by $\widetilde{P}_i^{\min}$. This implies that $P_i^{\min} + \widetilde{P}_i^{\min}$ is precisely the length of one cycle.

The first spans based metric is the minimum spans metric. Every element s_{ij} in the minimum spans metric matrix S, is calculated by the formula

$$s_{ij} = \begin{cases} (|P_i^{\min}| + |P_j^{\min}| - |P_i^{\min} \Cap P_j^{\min}|) & i < j \\ - & i \geq j. \end{cases} \tag{1}$$

There are two alternatives for a non-identical spans metric. The first option is the non-identical minimum spans metric which calculates the length of spans between orders i and

j that are used by both minimum spans. The matrix D with element d_{ij} for this metric can be computed by

$$d_{ij} = \begin{cases} (|P_i^{\min}| + |P_j^{\min}| - 2|P_i^{\min} \Cap P_j^{\min}|) & i < j \\ - & i \geq j. \end{cases} \tag{2}$$

The second option to calculate non-identical spans is to use the size of the overlap between the biggest gap of orders i and j in the calculation. In this case, the matrix E with element e_{ij} can be computed by

$$e_{ij} = \begin{cases} (|\widetilde{P}_i^{\min}| + |\widetilde{P}_j^{\min}| - |\widetilde{P}_i^{\min} \Cap \widetilde{P}_j^{\min}|), & i < j \\ - & i \geq j. \end{cases} \tag{3}$$

The fourth metric using a ratio could also be applied. This metric is defined as the ratio between the non-identical spans metric or the minimum span metric. The matrix A with element a_{ij} can be computed by

$$a_{ij} = \begin{cases} \frac{d_{ij}}{s_{ij}}, & i < j \\ - & i \geq j. \end{cases} \tag{4}$$

In Table 2(a) the minimum spans metric is applied to the small picking line example resulting in matrix S. As an example $s_{23} = (5 + 8 - 5) = 8$ results from batching the first order $P_2^4 = \langle 4, 8 \rangle$ and the second order $P_3^3 = \langle 3, 10 \rangle$ and these two spans overlap for 5 locations. Using S, a batch would be formed between orders 1 and 3 (and order 2 and 4). In Table 2(b) the non-identical minimum span metric leads to matrix D with first and fourth order (and second and third order) being batched. In Table 2(c) the non-identical span metric is applied and results in matrix E with the first and third order (and the second and fourth order) being batched. The spans ratio metric results in matrix A as described in Table 2(d). A batch would be formed by the first and fourth order (and second and third order).

$$S = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{matrix} 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 10 & \boxed{10} & 10 \\ - & 8 & \boxed{8} \\ - & - & 10 \end{bmatrix} \end{array}$$
(a)

$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{matrix} 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 8 & 5 & \boxed{6} \\ - & \boxed{3} & 4 \\ - & - & 5 \end{bmatrix} \end{array}$$
(b)

$$E = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{matrix} 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 8 & \boxed{5} & 6 \\ - & 5 & \boxed{6} \\ - & - & 5 \end{bmatrix} \end{array}$$
(c)

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{matrix} 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 0.8 & 0.5 & \boxed{0.6} \\ - & \boxed{0.4} & 0.5 \\ - & - & 0.5 \end{bmatrix} \end{array}$$
(d)

Table 2: *Metrics for the small example in Table 1. In all matrices the best batching is indicated by the framed numbers.*

For the 50 sample picking lines, combining the s_{ij} metric and GR heuristic resulted in the lowest number of cycles traversed. A combination of GS heuristic with D and GR heuristic with E generated the shortest walking distance for pickers over these metrics. Combining GS with A produced the shortest distance for A.

Nevertheless, compared to the picking location metric stops ratio none of the span-based metrics was able to produce a lower numbers of total cycles traversed. This is illustrated in

Figure 5. Longer individual spans could merge orders to result in a shorter combined span. Therefore, only using the minimum span does not produce better results. Using picker stops allows for various spans (not only the minimum span) to be taken into account, which leads to a lower total number of cycles traversed. Conclusions by Matthews & Visagie [23] on picker routing are confirmed by this result. The shortest spanning interval linking is outperformed by their travelling salesman approach, because various span options are allowed. Therefore, an additional approach of combining stops and spans in one batching metric is investigated.

3.2 Combining metrics

As an attempt to improve the walking distance, the two metrics based on stops and spans are merged into one metric. In the first step, orders are sorted in a list l based on some sorting criterion. Orders from the top of this list are then batched with another order based on a spans matrix M. In general, a lower number of total cycles traversed is generated by batching short orders first. In Algorithm 2 the process of this combination heuristic (CO) is presented.

Algorithm 2 (Combination heuristic)

Input: *Ascending list l, batching metric M that consists of a $n \times n$ matrix with entries m_{ij}, a solution set $\mathcal{G}$ that is empty*

Output: *Solution set $\mathcal{G}$ presented as a batched orders list*

```
𝒢 ← ∅
while | 𝒢 | < n/2 do
    k ← first order in l
    m_kq = min_j [m_kj]
    𝒢 ← 𝒢 ∪ (k, q)
    Remove row k and column q from M
    Remove the orders that correspond to k and q from l
end while
Return 𝒢
```

Calculating the number of picking locations per order is the start of the first combination configuration. Starting with the smallest number of stops, the orders are sorted to the largest in list l_s. Therefore, the first order for batching is selected by Algorithm 2 and the second order is added with regards to a matrix generated by a span metric from Section 3.1. The combination configurations makes four combinations possible (F-CO, P-CO, Q-CO, H-CO). They are listed in the first column of Table 3(a).

Orders can also be sequenced using the minimum span. List l_p sorts orders from shortest minimum span to longest. The first order to batch is then taken from the top of list p and the other order is added using one of the span metrics. This results in the four spans-spans combinations (K-CO, L-CO, O-CO, G-CO). These combinations are given in Table 3(b).

In the last option the first order is also chosen in the sequence of list p. A matrix provided by the picking location metrics developed by Hofmann & Visagie [19] (stops (T), non-identical stops (N), stops ratio (R)) selects the second order. This provides three possible combinations (U-CO, V-CO, W-CO). They are summarised in Table 3(c) as the spans-stops combinations.

Combination	Sort by	Batch by
F-CO	l_s	S
P-CO	l_s	D
Q-CO	l_s	E
H-CO	l_s	A

(a) Combining stops with spans.

Combination	Sort by	Batch by
K-CO	l_p	S
L-CO	l_p	D
O-CO	l_p	E
G-CO	l_p	A

(b) Combining spans with spans.

Combination	Sort by	Batch by
U-CO	l_p	T
V-CO	l_p	N
W-CO	l_p	R

(c) Combining spans with stops.

Table 3: *Metric configurations for the route overlap combination.*

In general, combinations that are based on the shortest or minimum span lead to better results as more specific location information is included in the metric calculation. The lowest number of total cycles traversed, evaluated over all span combination metrics, is produced by combining spans with spans. It selects the first order based on the shortest minimum span and then selects a batching partner from a picking location matrix.

The combination metrics reduce the cycles traversed when compared to the spans metrics (see Figure 5). However, the best solutions found be stops ratio metric is still not matched. Multiple orders can be included in the next batch, since stops or spans are sorted from smallest to largest leading to multiple orders with the same stops number or the same span length. The results are not changed significantly by changing the sequence within the ordered lists. Therefore, the next batching metrics try to overcome this limitation by incorporating information about picking locations and passed locations.

3.3 Addition metrics

Picking location metrics are added to span metrics in the third extension. Therefore, spans and stops information becomes available before routing. All potential combinations of stops and spans metrics introduced can be evaluated.

The first addition metric uses the matrix Z with general element z_{ij}. This stops non-identical spans metric is calculated by adding the stops metric to the non-identical spans. For the two orders the picking locations number is added to the combined span that is not part of a minimum span.

The metric's first part is adapted from Hofmann & Visagie [19]. The second part adjusts the smallest additional covering area proposed by Ho *et al.* [17] for this system. According to equation (3) both parts are added together combining orders that have both the shortest non-identical spans as well as the smallest number of picking locations.

Let the set $\mathcal{S}_i$ contain all the stops for order i. The stops non-identical spans metric z_{ij} is then computed by

$$z_{ij} = \begin{cases} (|\mathcal{S}_i| + |\mathcal{S}_j| - |\mathcal{S}_i \cap \mathcal{S}_j|) + (|\widetilde{P}_i^{\min}| + |\widetilde{P}_j^{\min}| - |\widetilde{P}_i^{\min} \Cap \widetilde{P}_j^{\min}|), & i < j \\ - & i \geq j. \end{cases} \tag{5}$$

In the next metric, the number of non-identical picking locations for two orders is added to the biggest gap of the two orders. This approach is called the non-identical stops-spans metric. The total number of non-identical picking locations suggested by Hofmann & Visagie [19] is this metric's first part. The second part adapts to the specific layout according to equation (3). In this approach orders form a batch which has the shortest non-identical span and the smallest number of non-identical picking locations. The non-identical stops-spans metric is calculated in a matrix B with general element b_{ij}. The formula is given by

$$b_{ij} = \begin{cases} (|\mathcal{S}_i| + |\mathcal{S}_j| - 2\,|\mathcal{S}_i \cap \mathcal{S}_j|) + (|\widetilde{P}_i^{\min}| + |\widetilde{P}_j^{\min}| - |\widetilde{P}_i^{\min} \Cap \widetilde{P}_j^{\min}|), & i < j \\ - & i \geq j. \end{cases} \quad (6)$$

In the final addition metric, the stops-spans ratio with general element c_{ij} is calculated by using the stops ratio as developed by Hofmann & Visagie [19] in the first part and adding a minimum span ratio with equation (4) in the second part. The first part of this metric divides the length of the non-identical minimum spans by the length of the combined minimum spans. The second part divides the number of non-identical picking locations by the number of combined picking locations. The incompatibility between orders is lower the closer this metric gets to zero. This metric can be computed by

$$c_{ij} = \begin{cases} \frac{|\mathcal{S}_i|+|\mathcal{S}_j|-2\,|\mathcal{S}_i\cap\mathcal{S}_j|}{|\mathcal{S}_i|+|\mathcal{S}_j|-|\mathcal{S}_i\cap\mathcal{S}_j|} + \frac{d_{ij}}{s_{ij}}, & i < j \\ - & i \geq j. \end{cases} \quad (7)$$

The matrix resulting from computing the stops non-identical spans metric for the example picking line in Table 1 is given in Table a. The first batch would contain the first and third order and the second batch the second and fourth order. In Table b the non-identical stops-spans metrics are calculated. The first batch would be composed of the first and fourth order and the second batch would contain the second and third order. Table c displays the stops-spans ratios. In Table c the stops-spans ratios are described with first and fourth, and second and third order being batched.

$$Z = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 15 & \boxed{13} & 14 \\ - & 12 & \boxed{13} \\ - & - & 13 \end{bmatrix} \end{array}$$

(a)

$$B = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 13 & 10 & \boxed{12} \\ - & \boxed{9} & 11 \\ - & - & 10 \end{bmatrix} \end{array}$$

(b)

$$C = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1.51 & 1.13 & \boxed{1.35} \\ - & \boxed{0.95} & 1.21 \\ - & - & 1.13 \end{bmatrix} \end{array}$$

(c)

Table 4: *Metrics for the small example in Table 1. The best batching is indicated by the framed numbers.*

Evaluated against the best performing picking location metric suggested by Hofmann & Visagie [19], namely the R-GR, the three addition metrics improve, it is have a lower total number of cycles traversed (see Figure 5). Using c_{ij} gives the shortest distance, namely 24 222 cycles on average. This approach is followed by r_{ij}, b_{ij} and z_{ij}. All the descriptive statistics over the 50 waves for GR and in combination with every addition metric are displayed in Table 5.

Nevertheless, the lowest number of total cycles is obtained when using the GS heuristic combined with the route overlap metrics. The algorithm searches deterministically and thus only generates one solution. For the route overlap metrics together with GS result in 24 189, 24 166 and 24 149 for c_{ij}, b_{ij} and z_{ij}, respectively. This leads to the recommendation of a combination of route overlap metrics and GS.

	R-GR	Z-GR	B-GR	C-GR
Mean	24 230	24 332	24 236	24 222
Median	24 229	24 336	24 236	24 223
Range	52	51	87	43
Std. Dev.	16.900	16.068	21.541	15.521
Variance	285.600	258.178	464.011	240.900

Table 5: *The statistics summarised (in terms of total number of cycles) to pick* 50 *waves by applying the GR heuristic.*

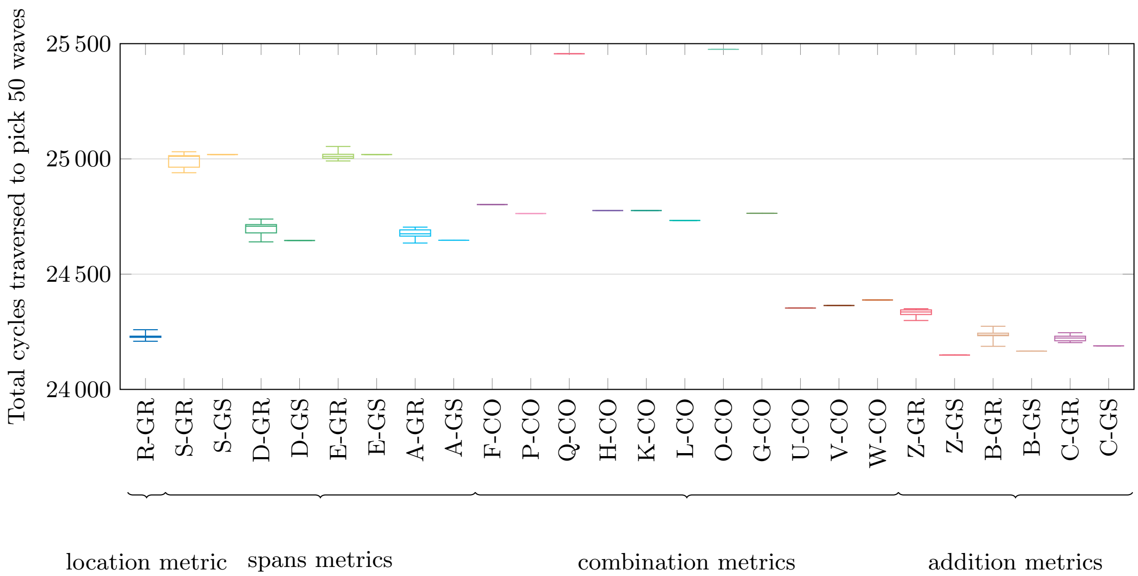


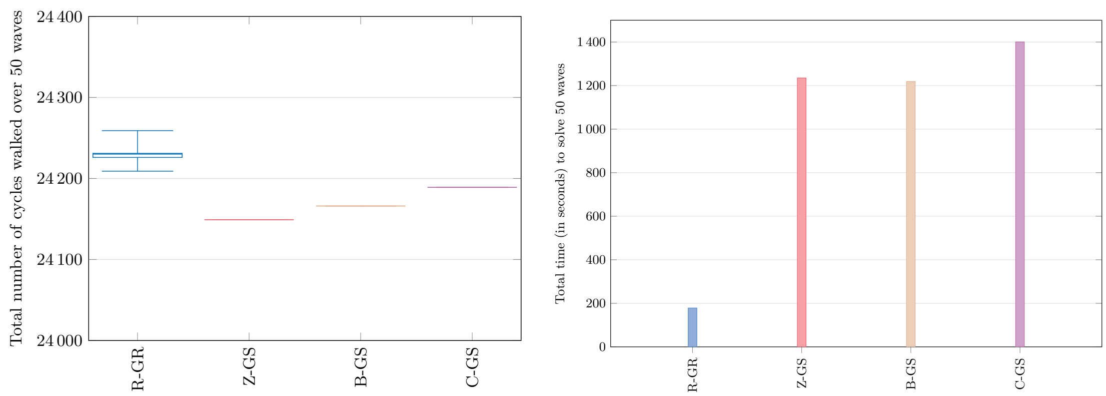
Figure 5: *Total number of cycles traversed per metric and solution algorithm displayed as box and whisker plots. The GR instances were solved* 10 *times.*

4 Discussion of experimental results

The route overlap addition metrics consistently provide the lowest number of total cycles to pick the 50 real-life waves. Thus, these metrics will be compared to the picking location metric stop ratio suggested by Hofmann & Visagie [19]. Additionally, the experimental results will be statistically analysed to determine if the solutions are indeed statistically significant.

4.1 Route overlap versus picking location metric

The best stops based metric found by Hofmann & Visagie [19] was the stops ratio together with a greedy random approach or R-GR. For route overlap metrics, presented here, the combinations of stops non-identical spans (Z-GS), non-identical stops-spans (B-GS), and the stops-spans ratio (C-GS) and greedy smallest entry heuristic generates the best results. The results for all these approaches are summarised in Figure 5 when applied to the 50 real-life waves of picking. In Figure 6 both these metric approaches are evaluated with regards to solution quality and time.



(a) Solution quality in cycles per metric.

(b) Solution time per combination of metric/solution approach.

Figure 6: *A summary of the solution quality (in total cycles) and times (in seconds) for different batching metrics.*

An upper and a lower bound is introduced to contextualise the different metrics' performance. Order batching would not be applied in the worst case. Therefore, the *maximal SKU* per picking wave describes the SKU that is visited by the largest number of orders. This generates a minimum or lowest number of cycles to be covered to pick all orders in a wave [23]. The maximal SKU for the 50 waves is 40 361 – resulting in at least so many cycles to pick all orders. Walking distance could be reduced by 50% in an extreme case of a perfect batching of every two orders. This means that the lower bound for the 50 waves when orders are batched is 20 181 full cycles. This is not a good or tight lower bound because it uses two consecutive relaxations (maximal SKU and perfect batching). The distance needed to link up orders is completely ignored. Furthermore, the fact that spans from different orders may not overlap is not included. The nearest end heuristic could also provide an approximation for comparison, since it is known to generate good solutions. Even though it performs well it should be noted that it is still a heuristic and thus cannot be used as a lower bound [23]. Without any batching, the nearest end heuristic produces 46 711 cycles. If this total pick path is halved and the two halves picked in parallel, it would result in 23 356 cycles. Note that this approach might not necessarily be a feasible solution. The 23 356 cycles are thus an estimate to aim for since it is generally closer to the optimal solution than the strict lower bound. Literature provides another benchmark, the FIFO rule. It is described in Section 3, and provides an additional solution for comparison. It gives a solution to a random batching approach.

The FIFO benchmark used in literature is outperformed by all four metrics. The R-GR combination, from the picking location metrics, generates the best solution quality with on average 24 230 cycles. In terms of the route overlap metrics, the worst one is the C-GS combination, which results in 24 189 cycles. This is still performing better than the best picking location metric. Therefore, it can be concluded that all the route overlap metrics outperform the picking location metrics in terms of solution quality. The data is summarised in Figure 6(a).

The best solution overall, with 24 149 cycles, is generated by the stops non-identical spans metric z_{ij} as different spans result from the smallest number of combined stops. In this metric order specific locations are important.

If the entire wave of picking should be optimised, the additional optimisation tasks of assigning the best subset of available SKUs to a wave as well as placing those SKUs in the best locations on the picking line have to be solved. Therefore, because in real life all three problems must be solved, a computational time limit of 30 seconds was set to solve the batching and routing problem. This time was determined in consultation with the management of the Retailer. In real-life applications they want to solve all three optimisation problems within one minute.

The solution times of the combination R-GR is shortest, as displayed in Figure 6(b). The route overlap metrics together with GS take longer to solve than R-GR – the best picking location metric. The entries in C, generated by a ratio, are very close to each other, as opposed to the entries in B or Z which have a wider range. This causes C-GS to take longer to solve. Regardless of the longer solution times it could still be implemented in real life.

4.2 Statistical analysis

Picking location metrics can be used as approximations for walking distance as validated by Hofmann & Visagie [19]. Therefore, a regression analysis also validates the application of route overlap metrics.

Each metric is solved as an assignment problem. The question thus arises if there is a correlation between the objective function value of the assignment problem and the actual number of cycles needed to pick a wave. Performing a regression analysis yields R^2 values of well above 0.5 for all metrics indicating a strong correlation. The R^2 values are 0.782, 0.786 and 0.785 for stops non-identical spans, the non-identical stops-spans, and stops-spans ratio, respectively. However, care should be taken with this regression analysis, since each metric's objective values is expressed in different units and can thus not be compared directly to each other.

Metrics influence the number of cycles traversed in different ways. This is analysed by inferential statistical tests. A univariate model that focuses on one metric or a bivariate model focusing on metrics and algorithms is suggested by Chiarandini *et al.* [3]. The total distance (in cycles) walked to pick the 50 waves is the main focus of analysis as the time requirements have proven to be suitable. To achieve this, a one-way ANOVA with a Tuskey's HSD *post-hoc* test [33] was applied to investigate the relation of the picking location metric stop ratio introduced by Hofmann & Visagie [19] and the route overlap metrics developed in this study to the final solution.

In Table 6 the statistical significance from the one-way ANOVA is ($F(3, 16) = 71.63, p = 1.72 \times 10^{-09}$). This emphasises the influence of the metric on the final solution (total

One-way ANOVA	Sum of squares	*df*	Mean square	*F*	*p*
Between metrics	18 632	3	6 211	71.63	1.72×10^{-09}**
Within metrics	1 387	16	87		

*** significance at less than or equal to a 5% level.*

Table 6: *Metrics analysed in one-way ANOVA.*

cycles). The Tuskey's HSD *post-hoc* test [34] shows a statistical significant difference between the picking location metrics and the route overlap metrics.

5 Conclusion

This study introduced the idea of order batching to a very specific type of picking system. Different layout specific order-to-route closeness metrics all of which are approximating the distance to pick orders before picker routes are known are introduced and investigated. There are three families of approaches, namely (a) stops non-identical spans metrics, (b) non-identical stops-spans metrics, and (c) stops-spans ratio metrics. The best solutions when applying these methods to 50 real-life picking waves (of varying sizes) is obtained when the metrics are combined with the greedy smallest entry algorithm.

In an attempt to incorporate more of the problem specific information into the solution approaches route overlap metrics using a minimum span measurement were introduced. This extends on the metrics suggested in an earlier study by Hofmann & Visagie [19].

Further improvement was found by combining these two metrics – it is route based and stop based metrics. The combination of Z-GS results in 81 cycles less than the R-GR combination. Thereby, on average almost 2 cycles per picking wave can be saved in combining the metrics. This saving is significant when taking into account how many waves of picking are performed in real life.

The combination Z-GS produces a solution that picks all 50 waves in 24 149 cycles. It is the lowest number for the 50 sample picking waves and is thus recommended. Random batching, which is often used as a benchmark in literature, uses 25 451 cycles – an increase of 5.4%. Batching alone has the biggest impact, reducing the distance by 48.3%.

Saving walking distance does not necessarily implies a saving in operational cost, which is what businesses would like to achieve. The decrease in cost thus needs to be quantified for these distance savings. Simulating the picking system could shed light on the cost savings. Picking more orders simultaneously lengthens the time pickers spend per location. Even though the picking system allows pickers to pass each other this may lead to a reduction in picking speed. Therefore future work could simulate the proposed solutions to estimate the direct operational time and cost savings they achieve.

References

[1] Bartholdi JJ & Hackman ST, 2016, *Warehouse & Distribution Science: Release 0.97*, Supply Chain and Logistics Institute, Available from `https://www.warehouse-science.com/book/index.html`.

[2] Bozer YA & Kile JW, 2008, *Order batching in walk-and-pick order picking systems*, International Journal of Production Research, **46(7)**, pp. 1887–1909.

[3] Chiarandini M, Paquete L, Preuss M & Ridge E, 2007, *Experiments on metaheuristics: Methodological overview and open issues*, Technical Report DMF-2007-03-003, Available from `https://pdfs.semanticscholar.org/9fcd/f48d4ae1770a6c653e33caee25aeb109b44d.pdf`.

[4] Clarke G & Wright JW, 1964, *Scheduling of vehicles from a central depot to a number of delivery points*, Operations research, **12(4)**, pp. 568–581.

[5] Dallari F, Marchet G & Melacini M, 2009, *Design of order picking system*, The International Journal of Advanced Manufacturing Technology, **42(1)**, pp. 1–12.

[6] De Koster M, Van der Poort ES & Wolters M, 1999, *Efficient orderbatching methods in warehouses*, International Journal of Production Research, **37(7)**, pp. 1479–1504.

[7] De Koster R, Le-Duc T & Roodbergen KJ, 2007, *Design and control of warehouse order picking: A literature review*, European Journal of Operational Research, **182(2)**, pp. 481–501.

[8] De Villiers AP, 2012, *Minimising the total travel distance to pick orders on a unidirectional picking line*, Masters Thesis, Stellenbosch University.

[9] Elsayed EA & Unal O, 1989, *Order batching algorithms and travel-time estimation for automated storage/retrieval systems*, The International Journal of Production Research, **27(7)**, pp. 1097–1114.

[10] Frazelle E, 2002, *World-class warehousing and material handling*, volume 1, McGraw-Hill New York.

[11] Gademann N & Velde S, 2005, *Order batching to minimize total travel time in a parallel-aisle warehouse*, Institute of Industrial Engineers Transactions, **37(1)**, pp. 63–75.

[12] Gibson DR & Sharp GP, 1992, *Order batching procedures*, European Journal of Operational Research, **58(1)**, pp. 57–67.

[13] Gu J, Goetschalckx M & McGinnis LF, 2007, *Research on warehouse operation: A comprehensive review*, European Journal of Operational Research, **177(1)**, pp. 1–21.

[14] Henn S, Koch S, Doerner KF, Strauss C & Wäscher G, 2010, *Metaheuristics for the order batching problem in manual order picking systems*, Business Research, **3(1)**, pp. 82–105.

[15] Henn S, Koch S & Wäscher G, 2012, *Order batching in order picking warehouses: a survey of solution approaches*, pp. 105–137 in *Warehousing in the Global Supply Chain*, pp. 105–137. Springer, London.

[16] Henn S & Wäscher G, 2012, *Tabu search heuristics for the order batching problem in manual order picking systems*, European Journal of Operational Research, **222(3)**, pp. 484–494.

[17] Ho YC, Su TS & Shi ZB, 2008, *Order-batching methods for an order-picking warehouse with two cross aisles*, Computers & Industrial Engineering, **55(2)**, pp. 321–347.

[18] Ho YC & Tseng YY, 2006, *A study on order-batching methods of order-picking in a distribution centre with two cross-aisles*, International Journal of Production Research, **44(17)**, pp. 3391–3417.

[19] Hofmann F & Visagie S, 2019, *Picking location metrics for order batching on a unidirectional cyclical picking line*, ORiON, **35(2)**, pp. 161–186.

[20] Litvak N & Vlasiou M, 2010, *A survey on performance analysis of warehouse carousel systems*, Statistica Neerlandica, **64(4)**, pp. 401–447.

[21] Matthews J, 2012, *Order sequencing and sku arrangement on a unidirectional picking line*, Masters Thesis, Stellenbosch: Stellenbosch University.

[22] Matthews J, 2015, *Sku assignment in a multiple picking line order picking system.*, Doctoral Dissertation, Stellenbosch: Stellenbosch University.

[23] Matthews J & Visagie SE, 2013, *Order sequencing on a unidirectional cyclical picking line*, European Journal of Operational Research, **231(1)**, pp. 79–87.

[24] Matthews J & Visagie SE, 2014, *Picking line data set a*, [Online], [Cited October 5th, 2018], Available from `http://hdl.handle.net/10019.1/86143`.

[25] Microsoft, 2018, https://www.microsoft.com/.

[26] Muter I & Öncan T, 2015, *An exact solution approach for the order batching problem*, IIE Transactions, **47(7)**, pp. 728–738.

[27] Nicolas L, Yannick F & Ramzi H, 2017, *Optimization of order batching in a picking system with carousels*, IFAC-PapersOnLine, **50(1)**, pp. 1106–1113.

[28] Pan CH & Liu S, 1995, *A comparative study of order batching algorithms*, Omega, **23(6)**, pp. 691–700.

[29] Python Software Foundation, 2018, *Python 3.6*, https://www.python.org/.

[30] R Core, 2019, *The R project for statistical computing*, https://www.r-project.org/.

[31] Rosenwein M, 1996, *A comparison of heuristics for the problem of batching orders for warehouse selection*, International Journal of Production Research, **34(3)**, pp. 657–664.

[32] Rouwenhorst B, Reuter B, Stockrahm V, van Houtum GJ, Mantel R & Zijm W, 2000, *Warehouse design and control: Framework and literature review*, European Journal of Operational Research, **122(3)**, pp. 515–533.

[33] Sheskin DJ, 2000, *Handbook of parametric and nonparametric statistical procedures*, CRC Press, Florida, United States.

[34] Statistical tools for high-throughput data analysis, 2020, *One way anova test in r*, [Online], [Cited May 20th, 2020], Available from `http://www.sthda.com/english/wiki/one-way-anova-test-in-r`.

[35] Van den Berg JP & Zijm WH, 1999, *Models for warehouse management: Classification and examples*, International Journal of Production Economics, **59(1-3)**, pp. 519–528.

[36] Wäscher G, 2004, *Order picking: a survey of planning problems and methods*, pp. 323–347 in *Supply chain management and reverse logistics*, pp. 323–347. Springer, Heidelberg.