Operational Workload Balancing in Manual Order Picking

Peer-reviewed author version

VANHEUSDEN, Sarah; VAN GILS, Teun; CARIS, An; RAMAEKERS, Katrien &
BRAEKERS, Kris (2020) Operational Workload Balancing in Manual Order Picking.
In: Computers & Industrial Engineering, 141, (Art N° 106269).

DOI: 10.1016/j.cie.2020.106269
Handle: http://hdl.handle.net/1942/30364

Operational Workload Balancing in Manual Order Picking

Abstract

A growing e-commerce market and increasing customer requirements put extra pressure on order picking operations. Collecting large quantities of relative small orders within limited time windows makes workload balancing in order picking a challenging and complicated task. Therefore, warehouse managers experience difficulties in balancing the daily workload of order pickers in every pick zone. This paper introduces the operational workload balancing problem within the domain of order picking. A mathematical model is introduced to describe the new order picking planning problem. Furthermore, an iterated local search algorithm is developed to solve the operational workload balancing problem efficiently and effectively. The benefits of daily workload balancing in order picking operations are analysed by means of a real-life case.
Keywords: warehouse management, workload balancing, zone picking, meta-heuristic

---

1. Introduction

Receiving, storage, order picking and shipping are critical warehouse operations to each supply chain. Recent market developments, such as e-commerce, globalisation and increased customer expectations, increase the complexity of managing warehouse operations in an efficient and effective way. These market developments intensify competition among warehouses (Van Gils et al., 2018c). In order to differentiate from competitors, warehouses aim to increase the service level by promising a quick and accurate order delivery (Ho and Lin, 2017).

The main warehouse operating costs can be attributed to the order picking process (Van Gils et al., 2018c). Order picking is defined as the retrieval of stock keeping units (SKUs) from storage locations to fulfil incoming customer orders. Order picking plays a critical role, both in terms of costs and service level (De Koster et al., 2012; Henn and Schmid, 2013). Time windows for picking orders have been reduced over the years. Customer orders need to be retrieved and shipped in a timely and efficient way within these limited windows. Previous studies have focussed on layout, storage, routing and batching planning problems (Elbert et al., 2017; Van Gils et al., 2018b; Žulj et al., 2018). The literature review of Van Gils et al. (2018c) concludes that there remains a need to account for more practically relevant factors when optimising order picking: hourly fluctuating demand and planning uncertainties pose new challenges for planning order picking operations such as balancing workload peaks (Boysen et al., 2018; Wruck et al., 2017). Despite the importance of human operators in the order picking process, research on human factors and workload balancing in warehouses is limited compared to literature on warehouse planning (Boysen et al., 2018; Grosse et al., 2015).

Figure 1 a illustrates the daily resource capacity planning process of most warehouses. First, the workload of tasks is forecast and afterwards, based on those forecasts, the workforce level (i.e., required number of order pickers) is determined. The number of order pickers is set sufficiently high to cover the forecast workload (i.e., ability to cover the highest expected workload in time in order to guarantee customer service). However, due to an increased need for flexibility and hourly fluctuating workload in warehousing environments, it would be more efficient to derive the number of needed order pickers from a balanced workload schedule (i.e., hourly daily schedule for which the workload is very similar each hour). Therefore, this study adds a new step to the daily resource capacity planning process of warehouses in which the forecast tasks are scheduled in such a way that workload peaks within a working day are avoided and the number of pickers can be derived from this balanced schedule afterwards (Figure 1 b).

The aim of this study is to schedule and balance the workload in order to prevent workload peaks. Workload peaks (i.e., situations for which the required order throughput exceeds the available picker capacity at certain periods of the day) in warehouses occur due to restricted time windows for retrieving orders. These peaks result in a high risk of missed departure deadlines of shipping trucks and therefore may result in delayed deliveries and stress among pickers and supervisors. Warehouse supervisors currently try to cope with these peaks in daily workload by assigning workers of other departments in the warehouse to the pick zones in need of extra hands. This last minute reassignment of workers, based on individual experiences and judgement of supervisors, often results in inefficiencies in these corresponding activities. These activities often get delayed or even shut down.

To avoid workload peaks and the consequent negative effects, workload needs to be balanced over the planning period. The operational workload balancing problem (OWBP), observed by the authors in a large international B2B spare parts warehouse located in Belgium, determines which customer orders to retrieve during which period of the day in order to avoid workload peaks, while respecting time windows. Workload balancing has been intensively studied in a production context (e.g., assembly line (Becker and Scholl, 2006) balancing). However, the labour-intensive operations and the strongly hourly fluctuating demand (i.e., number of customer orders that should be retrieved), requiring flexibility in resource capacity, differentiate warehouses from production facilities (Van Gils et al., 2017). Therefore, existing workload balancing methods

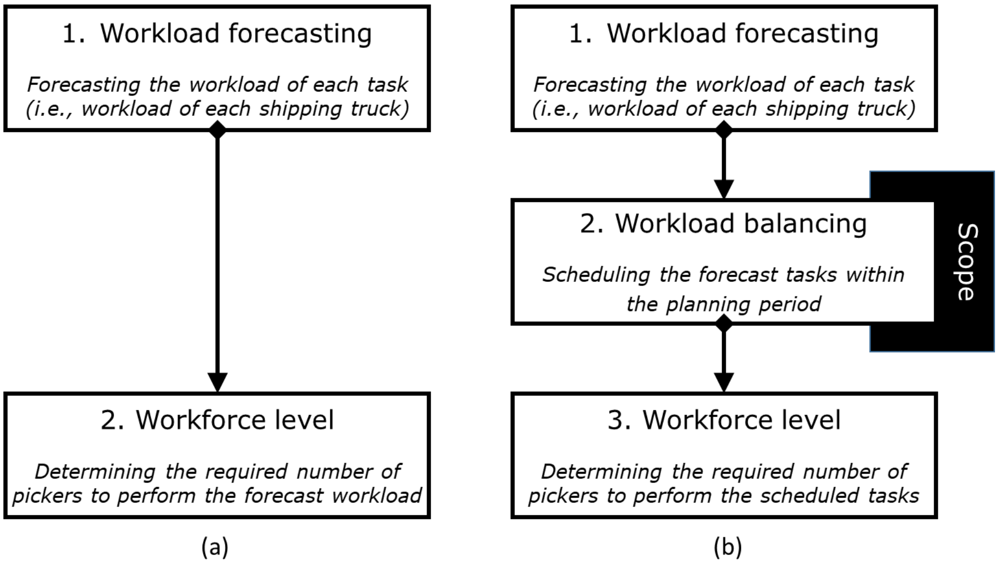


Figure 1: Current (a) and new (b) daily resource capacity planning process of warehouses.

do not cover all challenges of order picking environments.

The order picking system of the real-life case is manually operated, which is a system that is still popular in practice due to flexibility of human pickers, high investment costs of automated systems and the risk of interruptions (Calzavara et al., 2017; Marchet et al., 2015; Van Gils et al., 2019). Although the problem context is based on this real-life case, the operational workload balancing problem can be easily transferred to other manually operated warehouses or warehouses making use of pick robots, given the typical daily resource capacity planning process of warehouses.

The warehouse ground plan of the real-life case, illustrated in Figure 2 is used to clarify the terminology and specifications of the OWBP. Customer orders for the warehouse have to be shipped worldwide and therefore customer orders are grouped based on their destination. Orders for the same geographical location are assigned to the same shipping truck. In Figure 2 there are, by means of an example, three trucks shipping and multiple customer orders for three different geographical locations. Customers can order multiple spare parts (i.e., multiple order lines) in each customer order. In order to pick the requested order lines of a customer order, multiple zones are visited in the warehouse. One order line represents the SKU (or quantity of an SKU) that is retrieved at a single storage location. All order lines that should be shipped in a single pick truck and that are to be retrieved in a single pick zone are referred to as an *order set*. An example of an order set is visualised in Figure 2. By combining requested SKUs into order sets, we keep together SKUs with the same pick deadline, ensuring process control and preventing that these SKUs need to be consolidated after retrieval. Therefore, each order set can be assigned to only a single time slot in the daily balanced workload plan.

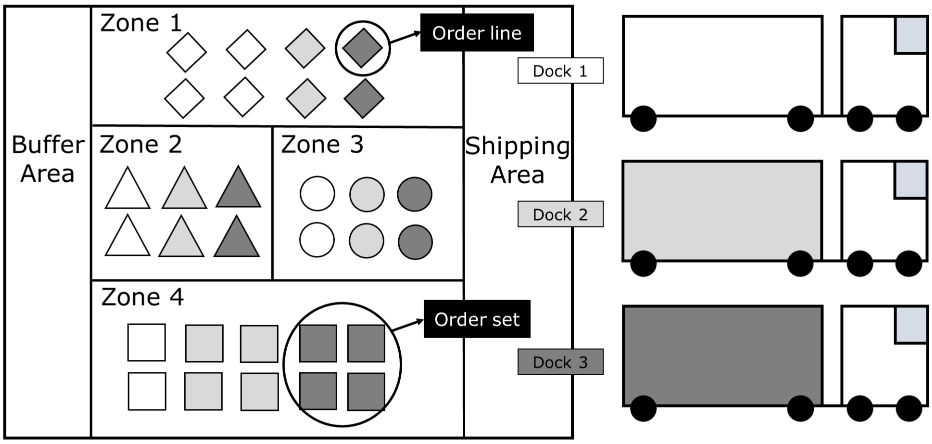
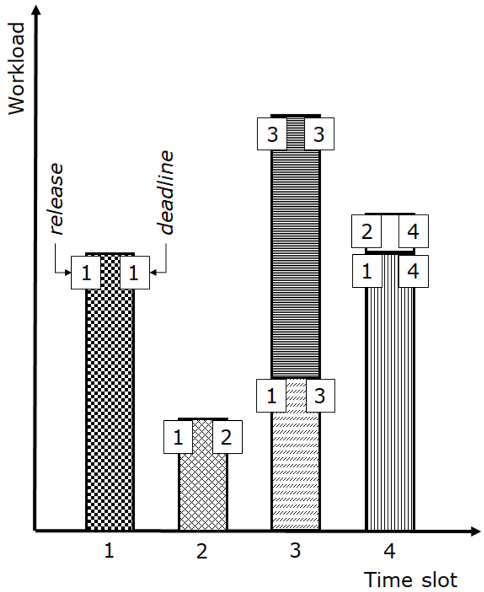


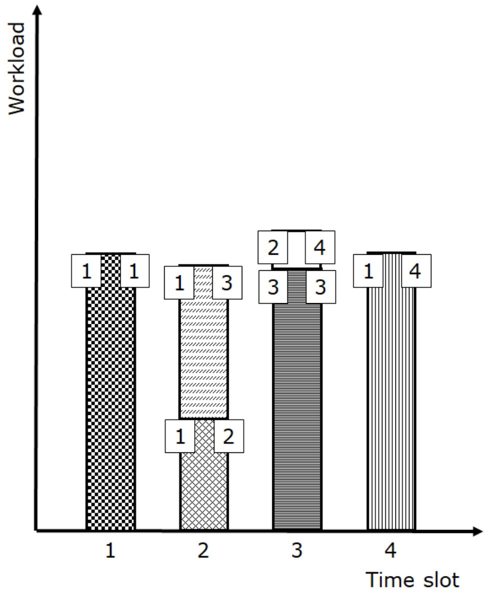
Figure 2: Simplified warehouse ground plan.

Figure 3 visualises the aim of the OWBP. Each block on the graph represents the workload of an order set, including the release and deadline of the particular order set. The OWBP balances daily workload by evenly dividing the order sets over the different time slots of the day, thereby respecting the release and deadline of each order set. In Figure 3(a) an undesirable workload schedule is visualised. Figure 3(b) represents the desired outcome for which the workload is as similar as possible in each time slot (i.e., as a result of balancing the workload for each time slot). Therefore, the solution of the model depicts a situation in which warehouses may hire a smaller number of order pickers (compared to Figure 3(a)) to process the resulting workload during the planning period in each zone. Thus, defining the number of workers is a decision taken after balancing the workload. Furthermore, warehouse supervisors can now check for every hour if they are on schedule. In other words, checking if all order sets of the respective time slot have been fulfilled. Splitting order sets is undesirable due to a loss of control. All order sets scheduled in a time slot (i.e. the result of solving the OWBP) can be used to create batches. In this way, wave picking (i.e., each time slot corresponds to a wave) can be efficiently applied (Petersen, 2000).

Furthermore, there are several practical issues that are taken into account. First, spare part warehouses store products that have large variations in dimensions and weights. The assignment of products to pick zones is assumed to be based on these characteristics (Dekker et al., 2004; Van Gils et al., 2018a) and assumed to be fixed at an operational level. For example, heavy and large products that do not fit standard Euro pallet measurements are grouped in a separate pick zone. This division is required because different handling methods are used for products with different dimensions. The specific characteristics of the pick



(a) Unbalanced workload schedule.



(b) Balanced workload schedule.

Figure 3: Simplified example of the OWBP (each block represents the workload of a set of orders.)

zones result in different productivity levels among the zones: the average productivity is low in the pick zone storing heavy SKUs, while the zone storing smaller items is designed to maximise the productivity. To mimic a realistic scenario, it is important to take these productivity differences into account when balancing the workload. Second, the warehouse uses a parallel zone picking system and all completed order sets are temporarily stored in front of a shipping dock before being loaded into the shipping truck. Each shipping truck can consist of multiple order sets (i.e., a single order set for each pick zone). A shipping dock is assigned to a truck whenever the first order set of a shipping truck is picked. The shipping dock stays occupied until the shipping deadline of that truck. Whenever all shipping docks are occupied, no order sets of new shipping trucks can be picked. In order to prevent shipping docks from overcrowding, the shipping dock capacity should be accounted for in the mathematical model. In addition, order sets of a shipping truck should be retrieved before the order due time (i.e., shipping deadline). The challenge of balancing these order sets over time slots is coherent with their corresponding time window (i.e., the available time to pick a customer order). Order sets can only be scheduled for picking in time slots between their order entry into the system (i.e., release) and the order due time as demonstrated in Figure 4. The assignment of the order sets to a shipping truck as well as the shipping deadline of these shipping trucks are assumed to be fixed at an operational level. The time schedule of shipping trucks is based on the delivery preferences

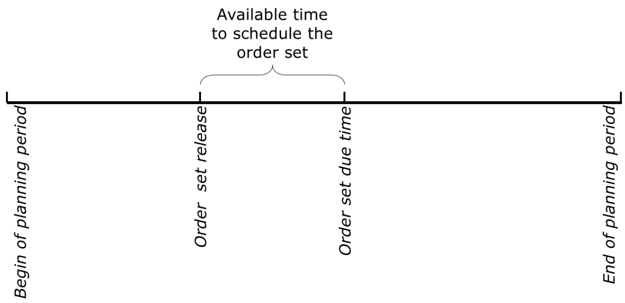


Figure 4: Time window of an order set

of customers. Unevenly divided daily shipping deadlines of shipping trucks on top of late customer order acceptance, results in tight time windows to fulfil orders and consequent workload peaks during the day.

This paper has three main contributions. The first contribution is the introduction of the OWBP, which is unexplored within the domain of order picking. A mathematical programming model is established to describe the OWBP. Due to the complex nature of the problem, Cplex is not able to solve the OWBP to optimality for all problem instances. Therefore, an iterated local search algorithm is developed to efficiently and effectively solve the OWBP. Second, the performance and practical applicability of the algorithm are shown by analysing and explaining the effect of a wide range of warehouse parameters. The proposed model provides schedules that show in which time slot (i.e., usually a single time slot for each hour) to pick orders of each shipping truck in order to balance the workload over all time slots. Third, a real-life case is used to show the benefits of balancing the workload over a short term planning period, thereby analysing the positive consequences of a balanced workload schedule.

The remainder of this paper is organised as follows. Section 2 highlights the differentiating elements of the OWBP compared to existing models in literature. The problem formulation is provided in Section 3. The developed iterated local search algorithm is discussed in Section 4, followed by the computational results in Section 5. Results of the case study and practical implications of this research for warehouse managers and supervisors are outlined in Section 6. Section 7 concludes the paper and elaborates on future research opportunities.

2. Literature Review

The literature section of this paper is organised as follows. Section 2.1 highlights the importance of workload balancing and discusses relevant differences from workload balancing papers in other research domains. Section 2.2 reviews related order picking literature with a focus on zoning and workforce scheduling in connection to workload balancing.

2.1. Workload Balancing in General

In general, workload balancing is defined as managing the variability of workloads over a time horizon (Irastorza and Deane, 1974). This concept of balancing workload is studied within various industries and research streams such as project scheduling (Rieck et al., 2012), vehicle routing (Matl et al., 2017), assembly line balancing (Becker and Scholl, 2006; Kingsman, 2000; Hillier and Brandeau, 2001) and facility balancing (Huang et al., 2006) Although these problems are related to the OWBP, the operational order picking context of our problem consists of clearly distinguishable elements, as explained below: flexible resource capacity, a fixed planning period, and tasks constrained by fixed time windows. As a result, existing solution approaches cannot be applied directly to the OWBP. An overview of the differences with respective problems are depicted in Table 1 and discussed below.

Table 1: Overview related problems.

	Resource capacity	Planning period	Tasks
OWBP	Flexible	Fixed	Constrained by time windows
Project scheduling	Fixed	Flexible	Constrained by precedences
Vehicle routing	Fixed	Flexible	Constrained by time windows
Assembly line balancing	Fixed	Flexible	Constrained by precedences
Facility balancing	Fixed	Flexible	Independent

Warehouses differ from production and manufacturing by delivering labour-intensive services to customers instead of goods assembled or produced by machines. The capacity of these production and manufacturing facilities is usually fixed in the short term (although overtime and subcontracting can be used in case of shortage) (Becker and Scholl, 2006), while warehouses can hire additional temporary order pickers in case of shortage (Van Gils et al., 2017). However, they are often more expensive due to their flexibility (Moons et al., 2018). Consequently, instead of including capacity as constraint before balancing, the workload is balanced without resource capacity constraints in the OWBP. Afterwards, the required number of pickers is derived form the balanced workload schedule.

In contrast to the project scheduling problem (Rieck et al., 2012) or balancing in the context of vehicle routing (Matl et al., 2017), the duration of the planning period is assumed to be fixed and the workload is balanced over this fixed planning period (i.e., usually a single working day). Instead of reducing the makespan to fulfil the tasks by increasing the number of resources or vehicles (Rieck et al., 2012; Matl et al., 2017), the OWBP assumes a fixed planning period and variable capacity to ensure that all order sets are picked before the deadline.

Existing methods often presume strict precedence constraints among tasks, such as assembly line balancing (Kingsman, 2000) and project scheduling (Rieck et al., 2012) problems, or completely independent tasks (e.g., workload balancing among facilities (Huang et al., 2006)), while scheduling order sets in the OWBP is not subject to precedence constraints, nor is scheduling completely independent. In the OWBP, order sets are picked independently from each other in each pick zone, while scheduling order sets from different pick zones is dependent as the shipping area is limited. Instead of precedence constraints, order sets are subject to fixed time windows, bounded by a release and deadline time slot in which they should be scheduled.

2.2. Workload Balancing in Order Picking

Zoning and workforce scheduling are well-known topics in warehousing literature. Both zoning and workforce scheduling can cause substantial workload peaks. Dividing the picking area into different zones can cause workload imbalances between pick zones (Jane and Laih, 2005). Poor workforce scheduling induces workload imbalance over time (Kim et al., 2018). Related literature, focussing on each of these planning problems, is briefly discussed below. Differences between the OWBP and respective zoning and workforce scheduling literature is depicted in Table 2.

Table 2: Literature on workload balancing in order picking.

		Decision level	Balancing unit
	OWBP	Operational	Over time (hourly)
Zoning	Zone assignment	Tactical	Among zones
	Zone size	Tactical	Among zones
	Bucket brigade	Operational	Among pickers
Workforce scheduling	Workforce capacity planning	Tactical	Over time (monthly)
	Workforce level	Operational	Over time (days)

A well-known opportunity to increase order picking performance is the division of the warehouse into different pick zones. Each order picker is assigned to a dedicated zone, and only picks the items of an order that are located in this pick zone (Yu and De Koster, 2009). Zone picking reduces travelling as pickers traverse only a small area of the warehouse. Furthermore, picker congestion is reduced, which results in substantial performance benefits compared to strict order picking (De Koster et al., 2012; Ho and Lin, 2017). Either parallel zoning (i.e., all zone pickers work on the same batch of orders) or sequential zoning (i.e., a batch of orders is sequentially passed from one zone to the other), cause workload imbalances among pick zones, as pick densities vary across these zones (Yu and De Koster, 2009). By varying the size of the pick zone (i.e., zone size) and varying the assignment of SKUs to pick zones (i.e., zone assignment), workload of the zones can be balanced in the long run (Jane, 2000; Jane and Laih, 2005). However, the proposed solution methods will be less suitable to balance the workload among pickers in the short term. Short term balancing can be achieved by considering dynamic zone picking systems, such as bucket brigades. Bucket brigades assume sequential zone picking with flexible zone borders, resulting in a self-balancing pick system with respect to the workload of individual order pickers (Hong et al., 2016).

Another way to assure customer service is not affected by peaks in workload is efficient scheduling and staffing of the order picking personnel. Efficient employability of human resources is necessary because of the labour intensive nature of warehouse operations. Warehouses are forced to deal with strong fluctuations in hourly demand and should simultaneously be able to meet fixed deadlines in short time intervals. To face these challenges, warehouses need to be highly flexible (Van Gils et al., 2017). Adaptations in the labour force can be used to cope with fluctuations in demand (Van den Bergh et al., 2013). Temporary workers are often hired in order to capture workload peaks between different days (Grosse et al., 2013). On the one hand, an insufficient number of workers causes a large picker workload and may reduce the service level because of missed deadlines. On the other hand, planning too many workers will cause unnecessarily high labour costs and a decrease in picking efficiency due to a small workload. To summarise, on a tactical level it is important to set up and manage a pool of fixed and temporary skilled order pickers (i.e., workforce capacity planning). Workers from this pool can be employed in order to balance the workload between days (i.e., workforce level) (Van Gils et al., 2017).

While most papers that cover the issue of workload balancing start at a tactical level (Jane, 2000; Jane and Laih, 2005; Hong et al., 2016), the emphasis of our research is on the operational level, to avoid peaks in the number of orders to be picked in certain time slots during the day. Therefore, the operational workload balancing problem considered in this paper is new, and differs from existing literature in zoning and workforce scheduling. Previous literature on zoning aims to balance the workload over the different zones (Jane, 2000; Jane and Laih, 2005), the OWBP balances daily workload within each pick zone and provides a schedule on the workload that needs to be performed within each pick zone. Additionally, the OWBP goes beyond the current state-of-the-art workforce scheduling literature (Van den Bergh et al., 2013) by balancing the workload for every hour of the day instead of balancing over shifts or days. To the best of our knowledge, we are the first to focus on workload peaks during a single day in a warehouse.

3. Problem Formulation

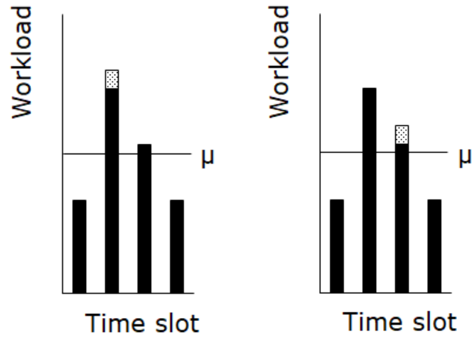
In order to balance the workload, one must be able to measure and correct existing imbalances in the order picking system. To measure and compare imbalances between solutions, the right equity metric and equity function have to be determined. The resource for which balancing is needed is defined as the equity metric (Matl et al., 2017). For example, for the OWBP the workload can be measured by the time that it takes to pick order sets. The term equity function is used interchangeably with the term equity measure. An equity measure is the index that is calculated for a workload allocation (Matl et al., 2017). Selecting the right equity measure does not only depend on the decision makers' interpretation and understanding of the concept of fairness, but is also subject to several underlying theoretical properties that differ between equity measures (Karsu and Morton, 2015).

In general, variance, standard deviation, mean absolute deviation (MAD), and range are considered as good metrics to measure balancing objectives (Nguyen and Wright, 2014). In order to solve the OWBP, the range is chosen as suitable objective function for several reasons. First, the range is a linear function in contrast to variance or standard deviation which penalise deviations from the average at a quadratic rate. Quadratic objective functions can be less intuitive measures for decision makers in practice and they also have a higher computational complexity (Matl et al., 2017). Second, range is able to minimise peaks and maximises the minimum value simultaneously in contrast to for example MAD, maximum or minimum, although these functions are linear as well. However, MAD not always able to minimise peaks in workload as shown in the examples of Figure 5. The MAD is equal in both examples, but the workload peak is substantially higher in Figure 5(a) than Figure 5(b) because transfers of order sets on the same side of the mean will not affect this measure. Minimising the maximum workload or maximising the minimum workload are not considered to be suitable objective functions for balancing the workload. For example, if the maximum is minimised, further balancing of the solution is not performed whenever peaks are at their minimal value. Although any changes between the extremes (i.e., minimum and maximum) have no effect on the range, it already provides more information than the simple min-max. Range is therefore the best linear metric for solving the OWBP. Furthermore, the range is easy to interpret and implement and is therefore frequently used in various applications (Matl et al., 2017).

A mixed integer programming (MIP) model is developed to formulate the OWBP. The notation outlined below is used throughout the paper:

Sets

κ	set of pick zones with $k = 1, 2, ..., K$
λ	set of shipping trucks with $l = 1, 2, ..., L$
τ	set of time slots with $t = 1, 2, ..., T$
$\tau_l \subset \tau$	subset of time slots containing all feasible time slots to schedule orders of



(a) (b)

Figure 5: Example of different solutions with equal MAD.

shipping truck l with $\tau_l = \{t_l^r, t_l^r + 1, ..., t_l^d - 1, t_l^d\}$

Parameters

o_{kl}	number of order lines for shipping truck l in zone k
p_k	mean productivity in minutes per order line in zone k
a_{kl}	workload of shipping truck l in zone k with $a_{kl} = o_{kl}p_k, \forall l \in \lambda, \forall k \in \kappa$
q	number of shipping docks
t_l^r	release time slot of shipping truck l
t_l^d	deadline time slot of shipping truck l

Decision variables

X_{klt}	binary variable which is equal to 1 if and only if order set $(k;l)$ is planned in time slot t
Y_{lt}	binary variable which is equal to 1 if and only if truck l reserves a shipping dock in time slot t
Z_l	first time slot that a shipping truck l occupies a dock
A_k^{max}	maximum planned workload over all time slots in zone k

A_k^{min} minimum planned workload over all time slots in zone k

Objective function

$$\min \mathcal{Z} = \sum_{k \in \kappa} \left(A_k^{max} - A_k^{min} \right) \tag{1}$$

Subject to

$$\sum_{l \in \lambda : t \in \tau_l} a_{kl} X_{klt} \leq A_k^{max} \qquad \forall t \in \tau \qquad \forall k \in \kappa \tag{2}$$

$$\sum_{l \in \lambda : t \in \tau_l} a_{kl} X_{klt} \geq A_k^{min} \qquad \forall t \in \tau \qquad \forall k \in \kappa \tag{3}$$

$$\sum_{t \in \tau_l} X_{klt} = 1 \qquad \forall l \in \lambda \qquad \forall k \in \kappa \tag{4}$$

$$\sum_{t \in \tau_l} t X_{klt} \geq Z_l \qquad \forall l \in \lambda \qquad \forall k \in \kappa \tag{5}$$

$$t - Z_l + 1 \leq T Y_{lt} \qquad \forall t \in \tau_l \qquad \forall l \in \lambda \tag{6}$$

$$\sum_{l \in \lambda} Y_{lt} \leq q \qquad \forall t \in \tau \tag{7}$$

$$X_{klt}, Y_{lt} \in \{0,1\} \qquad \forall t \in \tau \qquad \forall l \in \lambda \qquad \forall k \in \kappa \tag{8}$$

The objective function of the OWBP is defined by equation 1. The objective function minimises the range, which is the difference between the maximum (A_k^{max}) and minimum scheduled workload (A_k^{min}) over all time slots in each pick zone. Constraints 2 and 3 define the maximum and minimum workload over all time slots for each pick zone. Note that each combination of k and l represents an order set with workload a_{kl}. Constraints 4 assign each order set $(k; l)$ to a single time slot. Order sets are assigned to a single time slot in order to increase process control and in order to be able to apply wave picking on the order sets that are scheduled in the same time slot. The shipping dock capacity is included by Constraints 5 to 7, assuming that a shipping dock is reserved, or at least the buffer area in front of the shipping dock, from the first time slot that order sets of a shipping truck are scheduled until the shipping deadline of the truck. Constraints 5 define the first time slot that order sets of each shipping truck are planned among the pick zones. Constraints 6 define the time slots that a shipping truck reserves a shipping dock (i.e., from time slot Z_l until t_l^d). If t is larger than or equal to the first time slot that a shipping truck occupies a dock, the shipping dock is occupied in time slot t. Constraints 6 force Y_{lt} to be non-zero if $t \geq Z_l$. Constraints 7 limit the number of reserved docks to the shipping dock capacity. Note that only a single shipping truck is able to reserve a dock in each time slot. Finally, Constraints 8 define the domain of the decision variables. Note that the formulation forces the Z_l to be an integer.

4. Iterated local search algortihm for the OWBP

Due to the complex nature of the OWBP, solving all instances of realistic size to optimality in a reasonable amount of computation time does not seem feasible. Metaheuristic algorithms, such as Iterated local search (ILS), have shown promising results to solve complex warehouse management problems (Öncan, 2015; Scholz and Wäscher, 2017; Van Gils et al., 2018c). Therefore, an ILS algorithm dedicated to the OWBP is introduced in this section.

The reader is referred to Sörensen and Glover (2013) for the general principles of ILS (i.e., the creation of an initial solution, a local search procedure and perturbation strategy). The ILS principles are applied to the OWBP in the following way. In the initial solution, all order sets are assigned to the deadline time slot of their shipping truck. A feasible shipping schedule is assumed (i.e., the number of truck deadlines at each time slot is smaller than or equal to the number of available shipping docks). In this way, the shipping dock capacity constraint is satisfied. Next, a local search is performed on the initial solution until no further improvement in the objective function is possible. When the local optimum is reached, an iterative loop of diversification (i.e., perturbation) and intensification (i.e., local search) on a local optimum is performed. In addition to the general diversification and intensification principles of ILS, the diversification is enlarged by maintaining a pool of five solutions S rather than searching in a single solution (i.e., solutions with the smallest (s^1) and second smallest (s^2) objective function value as well as the three last found solutions(s^{r1}, s^{r2} and s^{r3})). Moreover, the local search is intensified by considering multiple neighbourhoods (i.e., operators) instead of a single neighbourhood. These deviations from the general ILS principles are commonly applied in other order picking planning problems, such as order batching and picker routing (Van Gils et al., 2018b). The iterative diversification and intensification are repeated until γ consecutive iterations without improved objective function value $f(s)$. This ILS framework, dedicated to the OWBP, is described in Algorithm 1.

The local search procedure, consisting of three deterministic operators (i.e., *order set shift*, *time slot swap*, and *order set swap*) and a stochastic operator (i.e., *ejection chain*), is described in Algorithm 2. The

Algorithm 1 Iterated local search algorithm for OWBP.

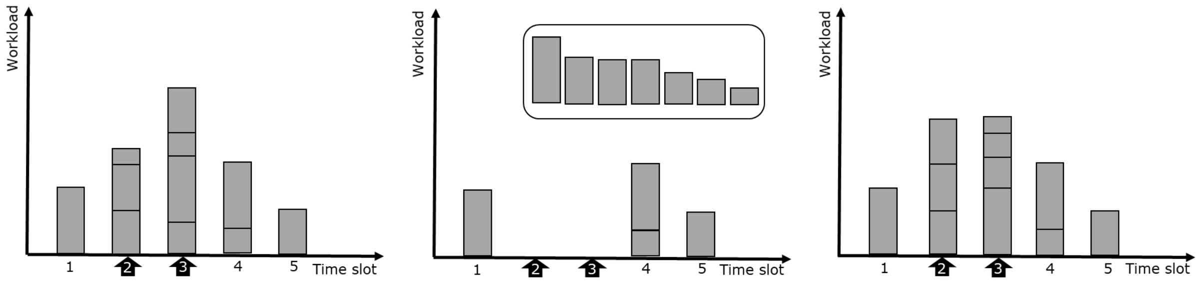
initialise the consecutive number of non-improving iterations: $\gamma' = 0$;
create initial solution s^0: assign each order set $(k;l)$ to t_l^d;
local search on s^0 (Algorithm 2);
initialise solution set $S(s^1; s^2; s^{r_1}; s^{r_2}; s^{r_3}) = S(s^0; s^0; s^0; s^0; s^0)$;
repeat
    count consecutive number of non-improving iterations: $\gamma' = \gamma' + 1$;
    randomly select s from S;
    perturbation on s (Algorithm 4);
    local search on s (Algorithm 2);
    if $f(s) \leq f(s^1)$ **then**
        new second best solution: $s^2 = s^1$;
        new best solution: $s^1 = s$;
        improved solution: $\gamma' = 0$;
    else if $f(s) \leq f(s^2)$ **then**
        new second best solution: $s^2 = s$;
    else
        new solution: $s^{r_3} = s^{r_2}$;
        new solution: $s^{r_2} = s^{r_1}$;
        new solution: $s^{r_1} = s$;
    end if
until $\gamma' = \gamma$;

order of the local search operators is fixed. More information on the order of the local search operators is available in Appendix A. A best improving move strategy is used to reach new solutions and only feasible solutions with respect to release and deadline constraints and shipping dock constraints are accepted as summarized in the performance evaluation of Algorithm 3. If a move results in an equal objective function value compared to $f(s)$, the new solution is accepted with a probability of θ percent. This happens frequently because the objective function value (i.e., range) is defined by only two time slots. The size of parameter θ is clarified in Appendix A. The shipping dock constraint is tested by comparing $Q_t^{s'}$ with q as follows:

$$Q_t^{s'} = \sum_{l \in \lambda} Y_{lt} \leq q \qquad \forall t \in \tau$$

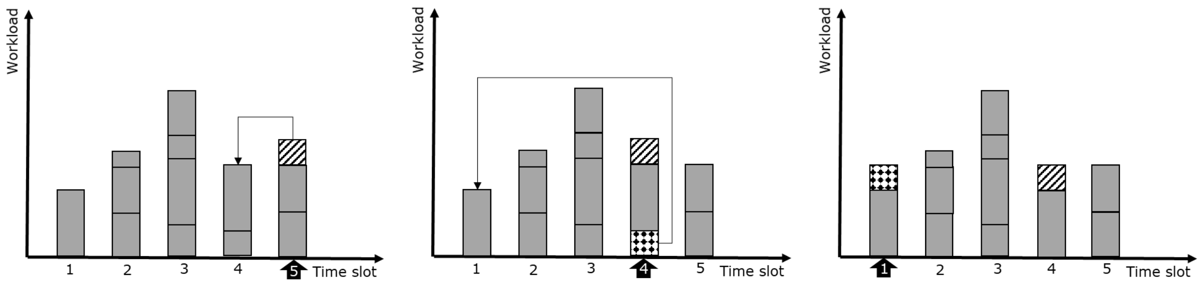
with $Q_t^{s'}$ the currently used number of shipping docks in time slot t of solution s'. If $Q_t^{s'} \leq q$, the number of used shipping docks does not exceed the capacity. Consequently, the move is feasible. All deterministic operators are repeated until no further improvement is possible, while the stochastic ejection chain is repeated for ι iterations without improved workload balance.

The *order set shift* operator aims to shift an order set in the solution from one time slot to another one. The *time slot swap* aims to optimise the workload between two consecutive time slots by removing all order sets that can be shifted between the two time slots (Figure 6(a)). The removed order sets are sorted in descending order with respect to the workload (Figure 6(b)) and are one by one reassigned to the time slot with the smallest workload (Figure 6(c)). The stochastic *ejection chain* aims to shift an order set to a new time slot (Figure 7(a)). If the objective function value is not improved, a random order set scheduled in the new time slot is shifted to another time slot (Figure 7(b)). These steps are repeated until an improved objective function value or a maximum of ϵ ejections within a chain is reached (Figure 7(c)). The *Order set swap* operator aims to reduce the workload imbalance by switching two order sets that are assigned to different time slots.



(a) (b) (c)

Figure 6: Time slot swap.



(a) (b) (c)

Figure 7: Ejection Chain.

While the local search algorithm results in a local optimum, the perturbation aims to escape from this local optimum by performing a large and random change to a randomly selected solution of the solution pool S. The dedicated OWBP perturbation procedure is described in Algorithm 4. The perturbation consists of a random removal of order sets from the solution and a random insertion. A random perturbation time slot t^p and pick zone k^p are selected. Order sets scheduled with a maximum of α time slots around time slot t^p are considered for removal: with a probability depending on γ' all order sets of a shipping truck are removed from the solution. Consequently, the chance of removing order sets of a shipping truck is higher when the number of consecutive iterations without improved objective function value increases. After removal, order

Algorithm 2 Local search.

```
repeat
  ORDER SET SHIFT
  repeat
    for all zones k ∈ κ do
      for all shipping trucks l ∈ λ do
        for all time slots t ∈ τ_l do
          create temporary solution: s' = s;
          shift order set (k; l) to t in s';
          performance evaluation of s' (Algorithm 3);
        end for
      end for
    end for
  until no further improvement is possible;
  TIME SLOT SWAP
  repeat
    for all zones k ∈ κ do
      for all time slots t ∈ τ\{T} do
        create subset of shipping trucks λ' ⊂ λ
        with λ' = all trucks l for which the order set (k; l) is currently assigned to t or t + 1, and t_l^r ≤ t ≤ t_l^d ;
        create temporary solution: s' = s;
        remove order sets (k; l') ∀l' ∈ λ' from s';
        sort all order sets (k; l') in descending order of a_{kl'};
        for all trucks l' ∈ λ' do
          if A_{kt} < A_{kt+1} then
            assign order set (k; l') to t in s';
          else
            assign order set (k; l') to t + 1 in s';
          end if
        end for
        performance evaluation of s' (Algorithm 3);
      end for
    end for
  until no further improvement is possible;
  EJECTION CHAIN
  repeat
    for all zones k ∈ κ do
      for all shipping trucks l ∈ λ do
        initialise number of ejections e = 0
        create temporary solution: s' = s;
        remove order set (k; l) from s';
        assign order set (k; l) to best time slot t, ∀t ∈ τ_l in s' and save best t* = t if f(s*) ≤ f(s);
        while f(s') ≤ f(s) and e ≤ ϵ do
          remove a random order set (k; l) in t* from s';
          assign order set (k; l) to best time slot t, ∀t ∈ τ_l in s'and save best t* = t;
          e = e + 1;
        end while
        performance evaluation of s' (Algorithm 3);
      end for
    end for
  until ι iterations without improvement;
  ORDER SET SWAP
  repeat
    for all zones k ∈ κ do
      for all shipping trucks l ∈ λ do
        for all shipping trucks l' ∈ λ if (k; l) and (k; l') assigned to different time slots do
          create temporary solution: s' = s;
          swap order set (k; l) and (k; l') in s';
          performance evaluation of s' (Algorithm 3);
        end for
      end for
    end for
  until no further improvement is possible;
until no further improvement is possible;
```

Algorithm 3 Performance evaluation.

```
calculate $Q_t^{s'}$;
if $Q_t^{s'} \leq q, \forall t \in \tau$ and $f(s') < f(s)$ then
    accept solution: $s = s'$;
else if $Q_t^{s'} \leq q, \forall t \in \tau$ and $f(s') = f(s)$ then
    accept solution with probability $\theta : s = s'$
end if
```

sets of a shipping truck are assigned to a random new time slot between the release and deadline. All order sets of the same truck are assigned to the same random time slot to increase the chance of a feasible solution with respect to the shipping dock capacity. If the complete perturbation results in a feasible solution, a local search is performed on the perturbed solution. Otherwise, a new perturbation is performed on the original selected solution from the solution pool.

Algorithm 4 Perturbation.

```
select a random pick zone $k^p \in \kappa$ and random time slot $t^r \in \tau$;
create temporary solution: $s' = s$;
create subset of shipping trucks $\lambda' \subset \lambda$
for all time slots $t \in [t^r - \alpha; t^r + \alpha]$ do
    for all shipping trucks $l$ planned in $t$ do
        remove order set $(k; l)$ from $s'$ in all pick zones $k \in \kappa$ with probability of $\min\{10\lceil 0.1\gamma' \rceil, 100\}$ percent;
        insert shipping truck $l$ in $\lambda'$;
    end for
end for
for all shipping trucks $l' \in \lambda'$ do
    repeat
        select randomly a time slot $t \in \tau_{l'}$
        for all zones $k \in \kappa$ do
            assign order set $(k; l')$ to $t$ in $s'$;
        end for
        calculate $Q_t^{s'}$;
    until $Q_t^{s'} \leq q, \forall t \in \tau$ or all $t \in \tau_l$ tested
end for
if $Q_t^{s'} \leq q, \forall t \in \tau$ then
    accept solution: $s = s'$;
else
    perturbation on $s$ (Algorithm 4);
end if
```

The parameters of the ILS algorithm (i.e., θ, γ, α, ϵ, and ι) are thoroughly tested to select appropriate parameter values. This procedure as well as the procedure to fix the order of local search operators is discussed in Appendix A.

5. Computational results of a generalised warehouse setting

In this section, experiments are performed to assess the performance of the ILS algorithm. The MIP model is implemented in `C++`. To solve the MIP formulation, `ILOG Cplex 12.7` is used with a runtime limit of 4 h. Cplex has been running on an Intel Xeon Processor E5-2680 at 2.8 gigahertz, using a single thread, provided by the Flemish Supercomputer Center. Section 5.1 gives an overview of the experimental design and the respective warehouse parameters that are used to show the practical applicability. Section 5.2 elaborates and discusses the results of the performed experiments by comparing the exact algorithm and the ILS algorithm. Furthermore, Section 5.3 provides an analysis of the effect of the different warehouse parameters on balancing possibilities.

5.1. Generalised experimental design

To assess the performance of the ILS algorithm and to generalise the conclusions of this study to other warehouses than the real-life case study discussed in Section 6, a comprehensive set of experiments is performed. The experimental factor setting is outlined in Table 3. Preliminary tests showed that the number of shipping trucks L, pick zones K, and shipping docks q are the most influential warehouse parameters. Additionally, two factors related to order characteristics are investigated: the variation in number of order lines between trucks (σ_a) and the mean number of available time slots to pick an order set (μ_{t_p}): the number of time slots between release and deadline time slots of a shipping truck.

Factor L contains three factor levels that vary the number of shipping trucks. The number of shipping trucks gives an indication on the number of order sets in every zone. For example, in case of the departure of 100 shipping trucks, 100 order sets need to be planned in every pick zone, assuming that each shipping truck contains order lines in every pick zone. Factor K defines the number of pick zones, which varies from one to three zones. The third factor in the experiment is the shipping dock factor q, which is tested at three levels. The size of q is of practical relevance as this factor determines the number of shipping docks as percentage of L (i.e., buffer space before the docks) and thus limits the number of order sets of different shipping trucks that can be picked simultaneously. The fourth factor σ_a defines the variation in number of order lines for every shipping truck. The parameter value (i.e., standard deviation) can be 50, 75 or 100 order lines, assuming a normal distribution with a mean of 175 order lines for every shipping truck. The fifth factor μ_{t_p} shows the possibilities for the mean number of time slots between the release and deadline time slots of a shipping truck.

From Table 3, it can be observed that the factorial setting of this paper results in a $3 \times 3 \times 3 \times 3 \times 3$ full factorial design. To reduce the stochastic effect of data generation, each factor level combination is replicated 10 times, resulting in $2,430$ problem instances. The ILS algorithm performs 10 runs for each of the $2,430$ problem instances. The wide range of factors allows us to provide general conclusions on the effect of different warehouse parameters on workload balancing. All problem instances are available from `XXX` *(instances will be made available after paper acceptance).*

Table 3: Experimental design.

factor	factor levels
L	(1) 100 trucks; (2) 150 trucks; (3) 200 trucks
K	(1) 1 zone; (2) 2 zones; (3) 3 zones
q	(1) $0.10L$ docks; (2) $0.15L$ docks; (3) $0.20L$ docks
σ_a	(1) 50 order lines; (2) 75 order lines; (3) 100 order lines
μ_{t_p}	(1) 1 time slot; (2) 2 time slots; (3) 3 time slots

Table 4 outlines other warehouse parameters that are assumed to be fixed in the experiments. According to these warehouse parameters and the factor levels of the experimental design, a Monte Carlo simulation is performed to create the problem instances.

The OWBP balances the workload over a day that is divided into time slots. Parameter T represents the number of time slots and corresponding wave length for batching orders. A single time slot for every hour of the day is assumed, which results in a total of 24 time slots. This wave length results in a sufficient mean number of orders per time slot, facilitating the creation of efficient batches. Moreover, 24 time slots allow the workload to be planned and controlled during the day.

Parameter o_l denotes the number of order lines in a shipping truck l. The number of order lines in a shipping truck is normally distributed with an average of 175 order lines and a standard deviation that varies by the experimental factor σ_a as discussed in Table 3.

Order lines of a shipping truck are picked in multiple pick zones. As the assignment of SKUs to pick zones is based on product properties, we assume that the number of order lines from a shipping truck (o_{kl}) is more or less proportionally divided over the zones (K). However, a uniform distribution $U(u_1; u_2)$ induces a slight variation in the number of order lines of each shipping truck among zones. The uniform distribution is based on the split shipping truck factor ι. This split shipping truck factor defines the distribution of order lines across pick zones. For example, if a shipping truck contains 175 order lines, originating from two zones, the number of order lines in the first zone is $175 \times U(0.45; 0.55)$ and the remaining order lines should be picked from the second zone.

Productivity values are different for each pick zone due to the specific characteristics of the products that are stored in each zone. For example, a zone may be designed to maximise throughput, storing only light weighted SKUs. Heavy products, often not fitting on standard Euro pallets, require different handling methods than the light weighted SKUs, and are therefore often stored in a separate zone. The different characteristics of the products stored in the zones result in productivity differences among these zones. For example, productivity will be lower in zones storing heavy SKUs as retrieval of products takes longer. Using the mean productivity for a pick zone (i.e., the mean time to retrieve an order line) is justifiable as the probability of picking large order lines (i.e., which take a longer time than the average retrieval time) is similar to the probability that smaller order lines need to be picked. Additionally, these productivity values indirectly reflect the efficiency of the different operational order picking policies (e.g., batching, storage assignment, routing). The more efficient the operational policies, the higher the productivity will be and the lower the average workload (expressed in time) will be in each time slot. The productivity varies over the zones without influencing the total workload. Values for p_k are denoted in Table 4.

Each shipping truck is characterised by a release time slot (t_l^r) and a deadline time slot (t_l^d). The release time slot and deadline time slot of a shipping truck are assigned to all order sets of this shipping truck. The release time slot of shipping truck l is uniformly distributed between 1 and 23. This simulates a real-life situation for international warehouses, as customers enter orders in the system from all over the world. The deadline time slot t_l^d is equal to time slot t_l^r raised with the number of available time slots to pick order lines from shipping truck l (t_l^p). Whenever this time slot exceeds the last time slot, t_l^d is set to time slot 24. Parameter t_l^p, which provides the available number of time slots to pick order lines from shipping truck l, is uniformly distributed within the range $[\max\{0; \mu_{t^p} - 2\}; \mu_{t^p} + 2]$, with experimental factor μ_{t^p} (Table 3).

In order to increase the practical relevance, we consider breaks that employees are allowed to take during the day. Because order pickers are unavailable during their break, the available time to pick orders is smaller in time slots containing a break compared to regular time slots. In the time slots containing a break, the available time to pick order lines is reduced by including artificial order lines in proportion to the unavailable time (i.e., $t_l^r = t_l^d =$ time slot containing a break). A small break of 10 minutes is included in time slots 8, 12 and 23. A large break of 20 minutes is added to time slots 5, 15 and 21.

Table 4: Warehouse parameter values.

Warehouse parameter		**Parameter value**
T	number of time slots	24 time slots
o_l	number of order lines of shipping truck l	$N(175; \sigma_a)$
o_{kl}	number of order lines of shipping truck l in zone k	$o_l\left(\frac{1}{K} + u\right)$ with $u \sim U(-\iota; \iota)$
ι	split shipping truck factor	0.05
p_k	mean productivity in zone k (in min. per order line)	$K = 1 : p_1 = 1$ $K = 2 : p_1 = 0.5, p_2 = 1.5$ $K = 3 : p_1 = 0.5, p_2 = 1, p_3 = 1.5$
t_l^r	release time slot of orders in shipping truck l	$U(1; I)$
t_l^p	available number of time slots to pick shipping truck l	$U(\max\{0; \mu_{t^p} - 2\}; \mu_{t^p} + 2)$
t_l^d	deadline time slot of orders in shipping truck l	$\min\left(T; t_l^r + t_l^p\right)$
$U(u_1; u_2)$	uniform distribution with u_1 (u_2) as lower (upper) bound	

5.2. Comparison between exact solution and ILS algorithm

The experimental points for this analysis are calculated by solving the OWBP using Cplex with a time limit of four hours for each of the 2,430 problem instances. Due to the complex nature of the OWBP, Cplex is not able to solve all instances to optimality. Table 5 shows the results of the MIP model by showing the number of non-optimal instances and their respective optimality gap. The number of observations for each factor level that did not solve to optimality within the run time limit is stated on the left-hand side of the table. For example, when the number of trucks L is 100, 73 out of 810 instances are not solved to optimality. In total, 27.37 % of the instances have not been solved to optimality (665 out of 2,430). The mean, minimum and maximum optimality gap of the non-optimal instances (excluding the 73 instances without feasible integer solution) are displayed on the right-hand side of the table. The number of non-optimal instances increases whenever there are more trucks L to be scheduled. More shipping trucks yield more order sets to assign to time slots, resulting in a substantially increasing solution space. The same goes for the number of zones K. More zones result in the creation of more order sets, increasing complexity and computation times. However, for the shipping dock capacity q, the opposite is true. Lower shipping dock capacities result in a higher number of non-optimal instances as it results in a more complex problem structure. Furthermore, the larger μ_{t_p}, the more time there is between the release and deadline time slot and the more planning possibilities exist for choosing a time slot to pick an order set, resulting in more non-optimal instances. For factor σ_a no clear relation seems to exist between the factor levels and the number of instances solved to optimality.

Table 5: Results of solving the MIP model with Cplex.

		Instances		**Optimality gap (in %)**		
		#	%	Mean	Min.	Max.
L	100	73	9.01	5.78	0.15	38.45
	150	236	29.14	5.20	0.11	99.84
	200	356	43.95	5.08	0.07	100.00
K	1	167	20.62	4.95	0.07	87.72
	2	229	28.27	5.59	0.11	100.00
	3	269	33.21	5.02	0.10	98.94
q	10	264	32.59	4.90	0.11	86.57
	15	216	26.67	4.55	0.07	98.94
	20	185	22.84	6.39	0.09	100.00
σ_a	50	234	28.89	5.67	0.10	100.00
	75	223	27.53	6.07	0.07	99.84
	100	208	25.68	3.73	0.09	57.79
μ_{t_p}	1	23	2.84	1.40	0.11	6.16
	2	219	27.04	3.64	0.07	61.78
	3	423	52.22	6.22	0.10	100.00
Total		**665**	**27.37**	**5.20**	**0.07**	**100.00**

To test the performance of the ILS algorithm, the optimal objective function values obtained by Cplex are compared with the results of solving the problem by the ILS heuristic. Cplex solved 1,765 instances out of 2,430 to optimality. The ILS algorithm performed 10 runs on the problem instances and finds in

80.06% the optimal solution in at least 1 of the 10 runs. Out of the 24,300 runs (i.e., 10 runs multiplied by 2,430 problem instances), only 1% of the runs have a gap that is larger than 5%. The mean percentage gap between the objective function values of the ILS and the optimal workload balance per factor level is illustrated in Figure 8. The various factor levels are shown on the horizontal axis. The mean optimality gap is less than 1.1% for all factor levels.

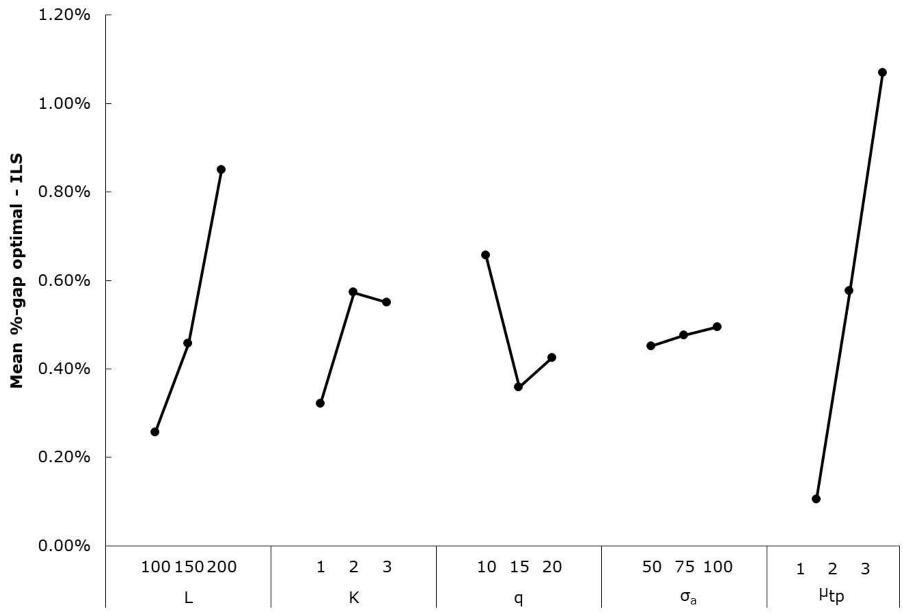


Figure 8: Percentage gap between ILS and the optimal solution provided by Cplex.

In order to determine the efficiency of the ILS algorithm, it is necessary to compare the CPU times of the ILS algorithm with the CPU times of solving the MIP by Cplex as illustrated in Figure 9. The ILS algorithm is faster than Cplex by a factor of 10^3. As can be observed from the graph, both Cplex and ILS computation times increase with the number of shipping trucks L, zones K and the available time to pick μ_{t_p}. The complexity of the problem is less affected by varying levels of the factors q and σ_a.

Computation times strongly increase with higher levels of factor L and K. Order lines of a shipping truck are divided over the pick zones in similar proportions. This is a reasonable assumption as zones are created based on product properties. More shipping trucks yields more order sets to be picked. Increasing the number of zones also leads to the creation of more order sets, as more zones need to be visited to satisfy customer demand. Because both factors increase the number of order sets, more order sets need to be assigned to time slots. This increases the complexity of the problem, resulting in higher CPU times. Factor μ_{t_p} reflects the available time between the release of the orders into the system and the latest pick deadline.

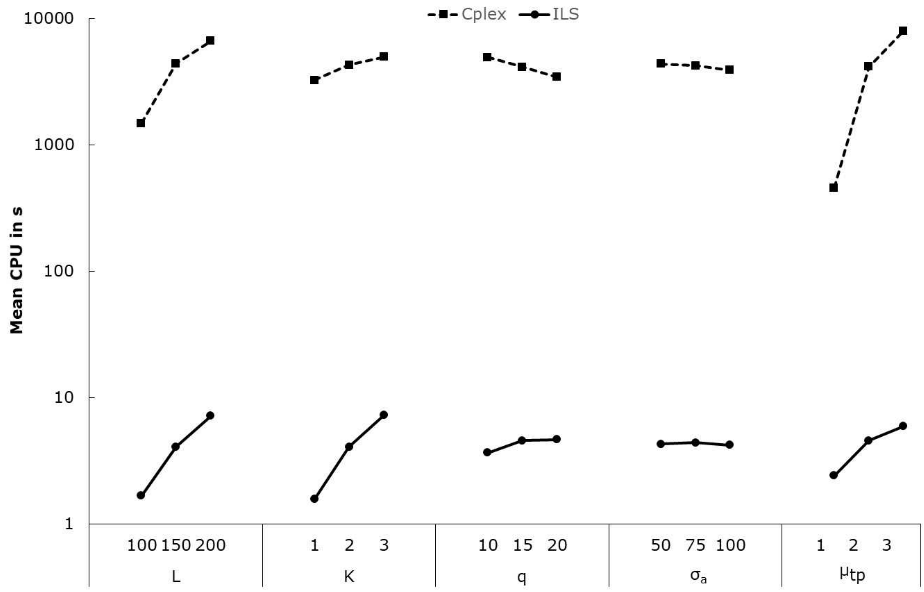


Figure 9: Mean CPU times for Cplex and ILS.

The higher this mean time to pick μ_{t_p}, the more possible pick time slots exist, increasing complexity and therefore also CPU times.

5.3. Analysis of the OWBP

In this section, each of the five experimental factors and their effect on the performance of the ILS algorithm are discussed based on Figure 10 and Table 6. Figure 10 visualises the mean range for each factor level of the experimental design. Furthermore, to make a more detailed analysis of the performance of the ILS algorithm, a balanced $3 \times 3 \times 3 \times 3 \times 3$ full factorial Analysis Of Variance (ANOVA) is presented (Table 6). ANOVA is frequently used to evaluate the effect of different warehouse parameters on the performance of order picking planning problems (Petersen, 2000; Quader and Castillo-Villar, 2018; Van Gils et al., 2018b). The first three columns of Table 6 show the sum of squares (SS), the degrees of freedom (df) and the resulting mean squares (MS) for each factor and each two way interaction among the factors, as well as for the residuals. The last two columns are devoted to the F statistic and the p-value for testing the statistical significance of the five factors and ten interaction effects.

Table 6 indicates that the main effects of the factors L, q, σ_a, and μ_{t_p} statistically significantly impact the range and resulting workload balance. Additionally, the two-way interactions among factors are tested, but only the relation between L and μ_{t_p} is found to be statistically significant at a significance level of 1 %. The effect of each factor is analysed and explained below.

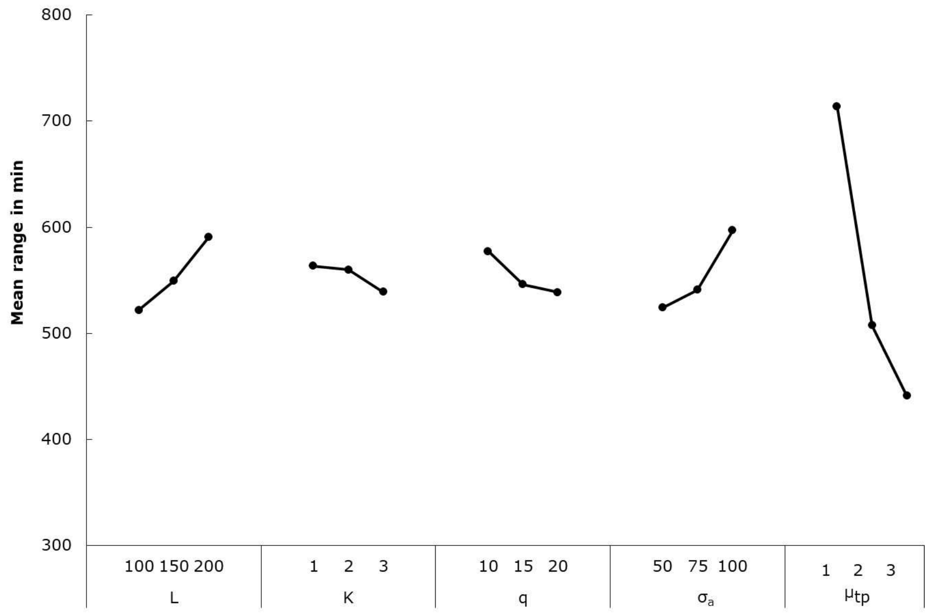


Figure 10: Mean range in minutes for ILS algorithm.

The workload imbalance statistically significantly grows by including more shipping trucks L (Table 6). Figure 10 shows an increasing range in case of more shipping trucks that are scheduled. More shipping trucks increase the probability of peaks that cannot be balanced due to restricted release and deadline time slots, as confirmed by the two-way interaction between L and μ_{t_p}.

Increasing the number of pick zones K may result in significant performance benefits in terms of order picking efficiency (De Koster et al., 2012). Results of this study show that varying the number of zones does not statistically significantly impact the mean range as stated in Table 6. Within zone picking, total workload (previously picked in 1 zone) is disaggregated into multiple zones. The total workload is preserved in the experiments resulting in no balancing differences between picking in a single zone or multiple zones.

The shipping dock capacity q varies from 10 to 20 shipping docks and statistically significantly impacts the mean range (Table 6). A shipping dock is reserved whenever the first order set of a shipping truck is picked. The shipping dock stays occupied until the shipping deadline of that truck. Whenever all shipping docks are occupied, no order sets of new shipping trucks can be picked. The smaller the number of shipping docks, the more difficulties the planning department will experience in balancing the workload, as less planning possibilities between release and deadline time slots are feasible. Therefore, more shipping docks result in a lower mean range as can be observed from Figure 10 .

Table 6 states that the effect factor σ_a on the mean range is statistically significant. The larger σ_a, the

Table 6: $3 \times 3 \times 3 \times 3 \times 3$ full factorial ANOVA on range.

	SS	df	MS	F	p
Main effects					
L	1,961,511	2	980,756	13.21	0.000
K	286,729	2	143,364	1.93	0.145
q	683,327	2	341,663	4.60	0.010
σ_a	2,361,677	2	1,180,838	15.90	0.000
μ_{t_p}	32,744,276	2	16,372,138	220.48	0.000
Two-way interaction					
$L \times K$	519,125	4	129,781	1.75	0.137
$L \times q$	102,918	4	25,729	0.35	0.847
$L \times \sigma_a$	342,366	4	85,591	1.15	0.330
$L \times \mu_{t_p}$	1,069,847	4	267,462	3.60	0.006
$K \times q$	20,815	4	5,204	0.07	0.991
$K \times \sigma_a$	22,451	4	5,613	0.08	0.990
$K \times \mu_{t_p}$	527,046	4	131,762	1.77	0.131
$q \times \sigma_a$	158,833	4	39,708	0.54	0.710
$q \times \mu_{t_p}$	821,077	4	205,269	2.76	0.026
$\sigma_a \times \mu_{t_p}$	267,518	4	66,879	0.90	0.463
Residuals					
Betw. subj.	176,656,184	2,379	74,256		
Total	218,545,700	2,429			

harder it becomes to balance the workload during the day, as can be observed from Figure 10. A large value for σ_a means that shipping trucks are very dissimilar regarding the number of order lines, resulting in a stronger imbalance in comparison to planning more similar shipping trucks. A larger variation in the number of order lines among shipping trucks results in the existence of large order sets. This increases the probability of peaks that can not be balanced due to restricted release and deadline time slots. Consequently, the workload imbalance statistically significantly grows in case of strong varying shipping trucks. Therefore, in addition to other objectives (e.g., minimising transportation cost) the distribution planning department should aim to balance the workload among shipping trucks when creating a shipping schedule; a smaller variation in number of order lines among shipping trucks yields a more balanced workload during the day.

Factor μ_{t_p} determines the available number of time slots that the order pickers have to pick order sets for a shipping truck. Varying the factor levels of μ_{t_p} does statistically significantly influence the range (see Table 6). Figure 10 shows the strong effect of the number of available time slots on the mean range. The larger μ_{t_p}, the more time there is between the release and deadline time slot and the more planning possibilities exist for choosing a time slot to pick an order set. This effect results in a steep downward trend in the range when the value of factor μ_{t_p} increases. The earlier customers order their products, the easier a warehouse is able to balance the workload during the working day. Moreover, the balanced schedule provided by the OWBP can reveal for which customers the available number of pick time slots is too small. The distribution planning department may reconsider current shipping schedules for certain clients. Negotiations may be started on earlier order entry by the customer or delayed shipping truck departures.

6. Real-life case study

This section discusses the applications, implications and benefits of using the OWBP model in practice. Real-life data are used to validate the OWBP model. First, the current operations of the real-life case are described. Second, results of balancing the workload of the real-life case study are discussed and visualised. Furthermore, insights are provided into the managerial decisions that the model can support. Finally, this section elaborates on the indirect implications and corresponding benefits of a balanced workload schedule in comparison to the current operation of the real-life warehouse.

In the current situation of the case study warehouse, order pickers gradually pick orders that enter the system, with a priority given to orders with pressing deadlines (i.e., earliest-due-time). Supervisors forecast the total workload of a working day (step 1 of the daily resource capacity process) and derive the resource capacity from this forecast (step 3 of the daily resource capacity process). This earliest-due-time scheduling of orders, and consequently skipping step 2 of the daily resource capacity process (Figure 1 b), means that the supervisors have no prior insights into the distribution of the workload during the day. Consequently, workload peaks occur in certain time slots, and the required order throughput may exceed the available capacity of order pickers. To solve this capacity problem, warehouse supervisors carry out last minute reassignments of employees from other warehouse activities to the different pick zones in need. The number of reassigned workers and the assignment of these workers to zones is based on individual experience and personal judgement. Thus, workload peaks negatively impact order picking operations and the last minute reassignments strongly disturb other warehouse activities.

To solve the above inefficiencies, the workload of the real-life case is balanced using the ILS algorithm. Historical data on customer order information of two consecutive years are provided by the case company. The case warehouse shows a weekly recurring cycle in number of order lines and shipping schedule (i.e., release and deadline time slot). To approximate the forecasting approach to define the workload of order sets, the average size of an order set for each working day is used as input data for the OWBP. The release time slot for an order set is the average time slot for which 95% of the order was entered into the system. The deadline time slot for each truck is determined by the shipping schedule. Table 7 shows the average CPU time and objective function value (i.e., range) for the five working days within the warehouse based on the 10 runs of the ILS algorithm.

To provide a more detailed insight into the decisions that could be supported by the OWBP model, Figure 11 depicts the workload schedules of an average Thursday in each pick zone of the case company (other days are similar). The left hand side (Figures 11(a), 11(c), 11(e) and 11(g)) illustrates the most

Table 7: Comparison of the working days on mean CPU and range.

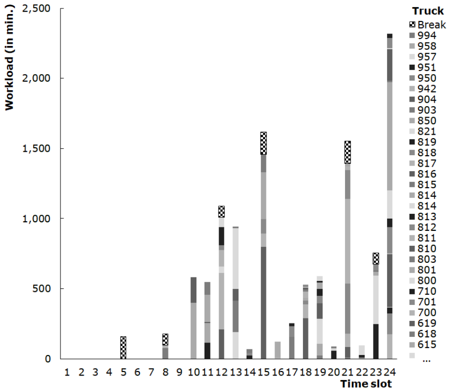
Day	Mean CPU	Mean range
Monday	37.22	624
Tuesday	33.26	594
Wednesday	24.69	939
Thursday	33.39	746
Friday	27.08	739

extreme case of imbalance by planning all order sets at the deadline of their respective shipping truck (i.e., the case company had no workload schedules). As becomes clear from the initial workload schedules, shipping deadlines for order sets are not equally divided over all time slots, resulting in workload peaks. Although these peaks are more moderate in practice due to the gradual picking of orders with pressing deadlines, the size of the workload peak is often unexpected. Figures 11(b), 11(d), 11(f) and 11(h) show the balanced workload schedules created by the OWBP: except from the small increase in the last time slots, the workload is properly balanced for each hour of the day in each pick zone.

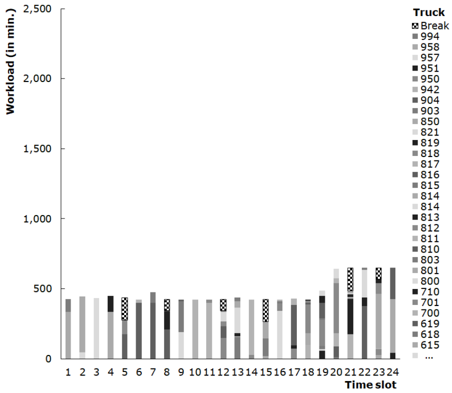
In addition to the main application (i.e., step 2 of the daily resource capacity planning process) and supporting the subsequent step of determining the required number of pickers to perform the scheduled workload in each time slot, the balanced workload schedules can be used by warehouse supervisors to support a wide range of other decisions, both operational and tactical.

First, at an operational level, the balanced workload schedules set goals for picking order sets in certain time slots. All order lines of the order sets scheduled in a time slot (or wave) can be used to create efficient batches. In this way, warehouse supervisors are better prepared and can check at each time slot end if all scheduled order sets are retrieved and if the right order sets are prepared to be loaded into the shipping truck. Supervisors are able to intervene timely if the picking process in not on schedule, without disturbing other warehouse processes. This results in an order picking process that is less depending on the individual experiences of warehouse supervisors.

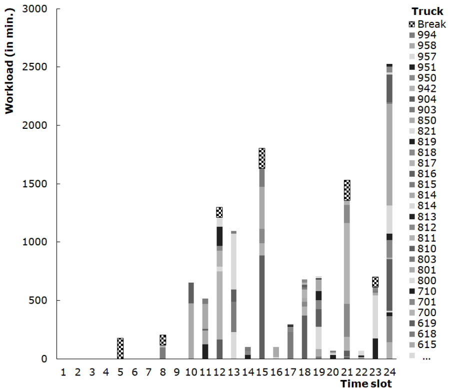
Second, in some situations however, further balancing by the OWBP could be impossible due to specific release and deadline time slots of shipping trucks (e.g., the last 3 to 5 time slots of the day in Figures 11(b), 11(d), 11(f) and 11(h)). Whenever many customers are allowed to enter orders at late time slots, the available time to pick these order sets is small and balancing the workload for these time slots becomes difficult. This corresponds to the findings of Section 5.3 which shows that the factor μ_{t_p} significantly impacts balancing possibilities. Consequently, at a tactical decision level, the OWBP can serve as an advisory tool for managers to start negotiations on changes in cut-off times for customer order entry and shipping schedules to further reduce these workload imbalances. Managers can negotiate with certain clients to put



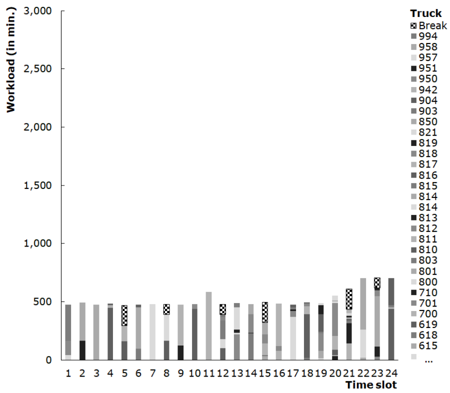
(a) Initial workload schedule ($k = 1$).



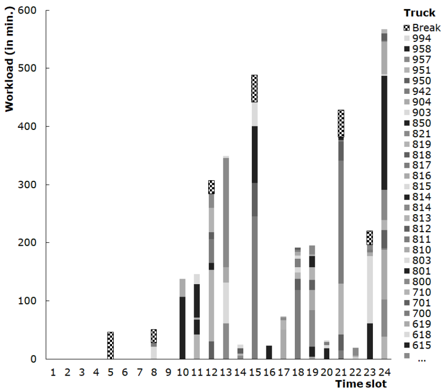
(b) Balanced workload schedule ($k = 1$).



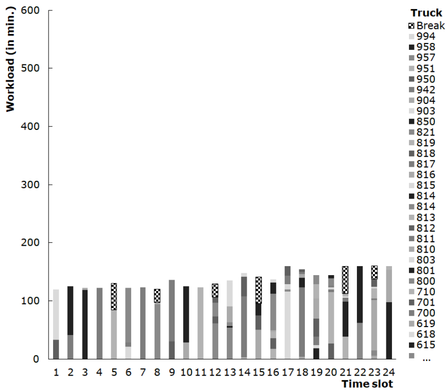
(c) Initial workload schedule ($k = 2$).



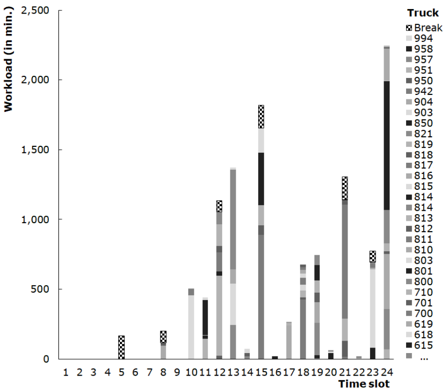
(d) Balanced workload schedule ($k = 2$).



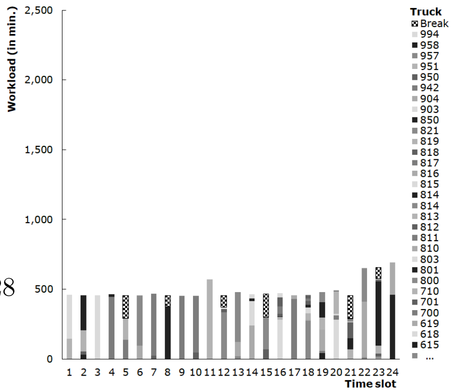
(e) Initial workload schedule ($k = 3$).



(f) Balanced workload schedule ($k = 3$).



(g) Initial workload schedule ($k = 4$).



(h) Balanced workload schedule ($k = 4$).

Figure 11: Example of the created workload schedules.

their orders some hours earlier in the system, resulting in more available time to pick orders. If a consensus can be achieved on changing the cut-off time for order entry, the warehouse can offer price reductions to that respective customer in return. These discounts can compensate for loss in customer service (i.e., earlier order entry). Changing the cut-off times could enable the OWBP model to balance the final existing imbalances.

Given the wide range of decisions that could be supported by the OWBP model, balancing the workload results in some additional indirect benefits. By anticipating on the unevenly divided departure deadlines of shipping trucks, the OWBP creates a balanced workload schedule in which order sets are picked further away from their deadline, resulting in lower probabilities of missing shipping deadlines compared to the current operation of the case warehouse. Moreover, the balanced workload schedules can be used as input to apply wave picking. This eventually increases order picking efficiency, especially compared to the current operation of the warehouse where smaller (and consequent less efficient) batches are created due to deadlines that pile up during peak periods. Finally, the OWBP model provides a balanced workload schedule, thereby preventing unexpected workload peaks. Order pickers do not have to cover unexpected peaks in workload, and supervisors can check at any time if the picking process is on schedule, which eventually could result in a lower work pressure and less stress.

7. Conclusions

Due to several upcoming trends in warehousing, time windows for order picking are shrinking, and the process of order picking is expected to keep improving in terms of flexibility. Late customer order acceptance in these limited time windows cause peaks in workload during the day, resulting in extra work pressure for both warehouse supervisors and order pickers. Until now, only solutions for long-term balancing have been introduced in literature. Practitioners are searching for a solution to balance the workload for every hour of the working day, to take their operational activities to a higher level.

This paper provides an effective linear mathematical formulation to describe the operational workload balancing problem in an order picking context. An efficient iterated local search algorithm is able to balance the workload during the planning period for a wide range of warehouse parameters. Although the paper focusses on a B2B context, the conclusions can be easily generalised to a B2C context as long as orders can be grouped based on a shipping truck and the workload can be accurately forecast, which are reasonable assumptions. The resulting balanced schedule is an excellent decision support tool for managers and supervisors to determine the required number of pickers each day, to allocate pickers to pick zones and to provide insights into the bottlenecks of the order process which can be used for negotiations with customers.

If the workload is balanced, warehouse supervisors are better prepared and other warehouse operations are less disturbed. In other words, the order picking process is less depending on individual experiences of warehouse supervisors. Furthermore, the manpower allocation process is supported due to a more accurate planning of the number of order pickers in each zone for every day of the week. By planning an evenly divided workload during the day, the probability of missing shipping deadlines will be smaller due to an intensified control. Additionally, the OWBP can be used as a supporting tool for managers to start negotiations in changes in cut-off times for customer order entry and shipping schedules to further reduce workload imbalances.

Several future research opportunities exist to extend this research work. The current manuscript introduces workload balancing as second step in the resource capacity planning process. Further investigating the complete resource capacity planning process, including the effect of forecast errors (step 1) and varying capacity levels (step 3) on order picking performance will be very interesting. Furthermore, this paper only assumes differences in productivity levels between zones due to the different characteristics of stored products (e.g., small items, heavy items). Adding varying levels of productivity among order pickers can therefore be an interesting addition to this research. Different levels of productivity exist due to different learning curves of order pickers (i.e., experienced order pickers and temporary workers). Searching for the optimal ratio between experienced order pickers and temporary workers can be an additional interesting research question. Interesting extensions to the current research could be the integration of individual pickers skills and characteristics within zoning in order to assign the right order picker to the right zone. This can further improve the process of manual order picking on top of the progress resulting from a balanced workload.

References

Becker C, Scholl A. A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research 2006;168(3):694–715. doi:10.1016/j.ejor.2004.07.023.

Boysen N, de Koster RBM, Weidinger F. Warehousing in the e-commerce era: A survey. European Journal of Operational Research 2018;doi:10.1016/j.ejor.2018.08.023.

Calzavara M, Glock CH, Grosse EH, Persona A, Sgarbossa F. Analysis of economic and ergonomic performance measures of different rack layouts in an order picking warehouse. Computers & Industrial Engineering 2017;111:527–36. doi:10.1016/j.cie.2016.07.001.

De Koster RBM, Le-Duc T, Zaerpour N. Determining the number of zones in a pick-and-sort order picking system. International Journal of Production Research 2012;50(3):757–71. doi:10.1080/00207543.2010.543941.

Dekker R, De Koster RBM, Roodbergen KJ, Van Kalleveen H. Improving Order-Picking Response Time at Ankor's Warehouse. Interfaces 2004;34(4):303–13. doi:10.1287/inte.1040.0083.

Elbert RM, Franzke T, Glock CH, Grosse EH. The effects of human behavior on the efficiency of routing policies in order picking: The case of route deviations. Computers & Industrial Engineering 2017;111:537–51. doi:10.1016/j.cie.2016.11.033.

Grosse EH, Glock CH, Jaber MY. The effect of worker learning and forgetting on storage reassignment decisions in order picking systems. Computers & Industrial Engineering 2013;66(4):653–62. doi:10.1016/j.cie.2013.09.013.

Grosse EH, Glock CH, Jaber MY, Neumann WP. Incorporating human factors in order picking planning models: framework and research opportunities. International Journal of Production Research 2015;53(3):695–717. doi:10.1080/00207543.2014.919424.

Henn S, Schmid V. Metaheuristics for order batching and sequencing in manual order picking systems. Computers & Industrial Engineering 2013;66(2):338–51. doi:10.1016/j.cie.2013.07.003.

Hillier MS, Brandeau ML. Cost minimization and workload balancing in printed circuit board assembly. IIE Transactions 2001;33(7):547–57. doi:10.1023/A:1010840432611.

Ho YC, Lin JW. Improving Order-Picking Performance by Converting a Sequential Zone-Picking Line into a Zone-Picking Network. Computers & Industrial Engineering 2017;113:241–55. doi:10.1016/j.cie.2017.09.014.

Hong S, Johnson AL, Peters BA. Order batching in a bucket brigade order picking system considering picker blocking. Flexible Services and Manufacturing Journal 2016;28(3):425–41. doi:10.1007/s10696-015-9223-5.

Huang HC, Lee C, Xu Z. The workload balancing problem at aircargo terminals. OR Spectrum 2006;28(4):705–27. doi:10.1007/s00291-006-0035-6.

Irastorza JC, Deane RH. A Loading and Balancing Methodology for Job Shop Control. A I I E Transactions 1974;6(4):302–7. doi:10.1080/05695557408974968.

Jane CC. Storage location assignment in a distribution center. International Journal of Physical Distribution & Logistics Management 2000;30(1):55–71. doi:10.1108/09600030010307984.

Jane CC, Laih YW. A clustering algorithm for item assignment in a synchronized zone order picking system. European Journal of Operational Research 2005;166(2):489–96. doi:10.1016/j.ejor.2004.01.042.

Karsu Ö, Morton A. Inequity averse optimization in operational research. European Journal of Operational Research 2015;245(2):343–59. doi:10.1016/j.ejor.2015.02.035.

Kim TY, Dekker R, Heij C. Improving warehouse labour efficiency by intentional forecast bias. International Journal of Physical Distribution & Logistics Management 2018;doi:10.1108/IJPDLM-10-2017-0313.

Kingsman BG. Modelling input–output workload control for dynamic capacity planning in production planning systems. International Journal of Production Economics 2000;68(1):73–93. doi:10.1016/S0925-5273(00)00037-2.

Marchet G, Melacini M, Perotti S. Investigating order picking system adoption: a case-study-based approach. International Journal of Logistics Research and Applications 2015;18(1):82–98. doi:10.1080/13675567.2014.945400.

Matl P, Hartl RF, Vidal T. Workload Equity in Vehicle Routing Problems: A Survey and Analysis. Transportation Science 2017;52(2):239–60. doi:10.1287/trsc.2017.0744.

Moons S, Ramaekers K, Caris A, Arda Y. Integration of order picking and vehicle routing in a B2c e-commerce context. Flexible Services and Manufacturing Journal 2018;:1–31doi:10.1007/s10696-017-9287-5.

Nguyen TH, Wright M. Variable neighborhood search for the workload balancing problem in service enterprises. Computers & Operations Research 2014;52:282–90. doi:10.1016/j.cor.2013.07.027.

Öncan T. MILP formulations and an Iterated Local Search Algorithm with Tabu Thresholding for the Order Batching Problem. European Journal of Operational Research 2015;243(1):142–55. doi:10.1016/j.ejor.2014.11.025.

Petersen CG. An evaluation of order picking policies for mail order companies. Production and Operations Management 2000;9(4):319–35. doi:10.1111/j.1937-5956.2000.tb00461.x.

Quader S, Castillo-Villar KK. Design of an enhanced multi-aisle order-picking system considering storage assignments and routing heuristics. Robotics and Computer-Integrated Manufacturing 2018;50:13–29. doi:10.1016/j.rcim.2015.12.009.

Rieck J, Zimmermann J, Gather T. Mixed-integer linear programming for resource leveling problems. European Journal of Operational Research 2012;221(1):27–37. doi:10.1016/j.ejor.2012.03.003.

Scholz A, Wäscher G. Order Batching and Picker Routing in manual order picking systems: the benefits of integrated routing. Central European Journal of Operations Research 2017;:1–30doi:10.1007/s10100-017-0467-x.

Sörensen K, Glover FW. Metaheuristics. In: Encyclopedia of operations research and management science. Springer; 2013. p. 960–70.

Van den Bergh J, Beliën J, De Bruecker P, Demeulemeester E, De Boeck L. Personnel scheduling: A literature review. European Journal of Operational Research 2013;226(3):367–85. doi:10.1016/j.ejor.2012.11.029.

Van Gils T, Caris A, Ramaekers K. Reducing picker blocking in a high-level narrow-aisle order picking system: insights from a real-life spare parts warehouse. In: 2018 Winter Simulation Conference (WSC). Gothenborg; 2018a. p. 2953–65. doi:10.1109/WSC.2018.8632435.

Van Gils T, Caris A, Ramaekers K, Braekers K, De Koster RBM. Designing efficient order picking systems: the effect of safety constraints, picker blocking, and high-level storage on the relation among planning problems. Transportation Research Part E: Logistics and Transportation Review 2019;125:47–73. doi:10.1016/j.tre.2019.02.010.

Van Gils T, Ramaekers K, Braekers K, Depaire B, Caris A. Increasing Order Picking Efficiency by Integrating Storage, Batching, Zone Picking, and Routing Policy Decisions. International Journal of Production Economics 2018b;197(Part C):243–61. doi:10.1016/j.ijpe.2017.11.021.

Van Gils T, Ramaekers K, Caris A, Cools M. The use of time series forecasting in zone order picking systems to predict order pickers' workload. International Journal of Production Research 2017;55(21):6380–93. doi:10.1080/00207543.2016.1216659.

Van Gils T, Ramaekers K, Caris A, De Koster RBM. Designing Efficient Order Picking Systems by Combining Planning Problems: State-of-the-art Classification and Review. European Journal of Operational Research 2018c;267(1):1–15. doi:10.1016/j.ejor.2017.09.002.

Wruck S, Vis IFA, Boter J. Risk control for staff planning in e-commerce warehouses. International Journal of Production Research 2017;55(21):6453–69. doi:10.1080/00207543.2016.1207816.

Yu M, De Koster RBM. The impact of order batching and picking area zoning on order picking system performance. European Journal of Operational Research 2009;198(2):480–90. doi:10.1016/j.ejor.2008.09.011.

Žulj I, Glock CH, Grosse EH, Schneider M. Picker routing and storage-assignment strategies for precedence-constrained order picking. Computers & Industrial Engineering 2018;123:338–47. doi:10.1016/j.cie.2018.06.015.

Appendices

Appendix A. Parameter setting

Thirty test instances are randomly selected from the experimental design of Table 3 to decide on the best order of the local search operators and are at the same time used to set the parameters of the ILS algorithm. On average, the smallest objective function values and CPU times are reached whenever local search operators are set in the following order: *order set shift*, *time slot swap*, *ejection chain* and finally *order set swap*. Furthermore, the following parameters are tested on the test instances: θ, γ, α, ϵ, and ι. Different levels of θ have been analysed and it can be concluded that the value of θ has a very small influence on the solution quality and is therefore not included in the experimental design for the parameter setting outlined in Table A.8. Parameter θ is set to 50%. The remaining set of parameter combinations is evaluated with respect to the resulting workload balance and CPU time. Results are shown in Fig. A.12. The factor levels with $\gamma = 0$ show strong increased objective function values. For visibility reasons, these results are not shown in Figure A.12.

With respect to all other factor level combinations, the objective function value is rather insensitive to the parameter values. Largest fluctuations in objective function value are visible whenever $\gamma = 125$. Therefore γ is set to 250. Larger values for γ do not show enough improvement in objective function value to justify the increase in computation times. The selected combination is marked in Figure A.12. On average, the chosen combination of $\gamma = 250$, $\alpha = 3$, $\epsilon = 6$, and $\iota = 3$ results in a relatively small workload imbalance accompanied by rather small computation times. Therefore, this combination is used throughout all further experiments.

Table A.8: Experimental design for parameter setting.

factor	factor levels
γ	125; **250**; 375; 500
α	1; 2; **3**
ϵ	4; **6**; 8
ι	1; 2; **3**

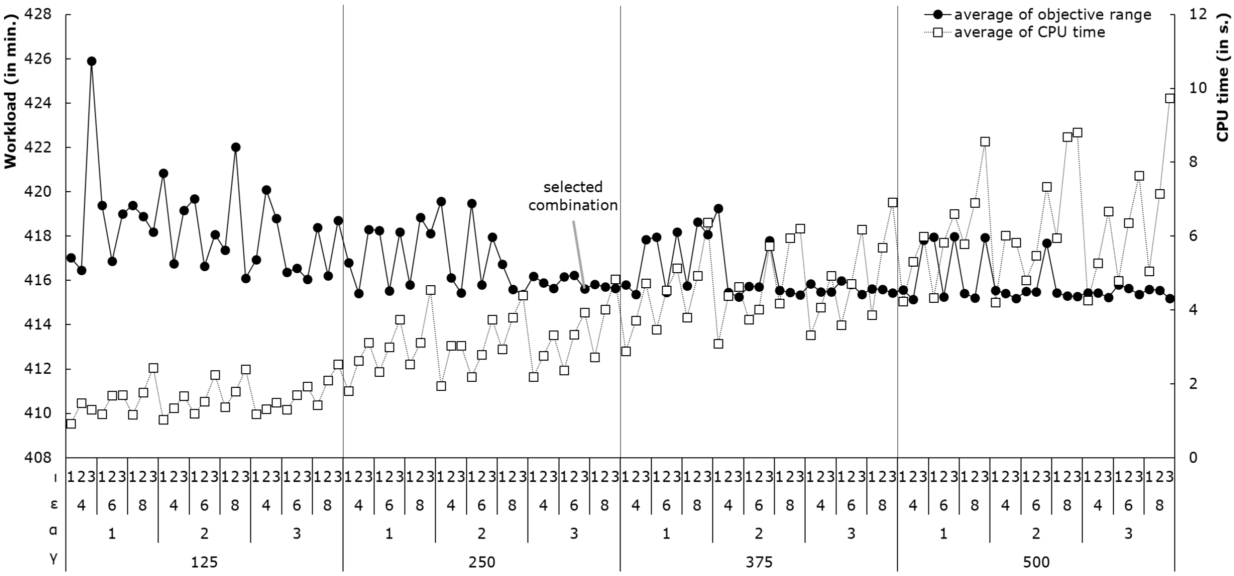


Figure A.12: Parameter tuning.