Workforce Scheduling with Order-Picking Assignments in Distribution Facilities

Arpan Rijal
Rotterdam School of Management, Erasmus University, The Netherlands, rijal@rsm.nl

Marco Bijvank
Haskayne School of Business, University of Calgary, Canada, marco.bijvank@haskayne.ucalgary.ca

Asvin Goel
Kühne Logistics University, Hamburg, Germany, asvin.goel@the−klu.org

René de Koster
Rotterdam School of Management, Erasmus University, The Netherlands, rkoster@rsm.nl

Scheduling the availability of order pickers is crucial for effective operations in a distribution facility with manual order pickers. When order-picking activities can only be performed in specific time windows, it is essential to jointly solve the order picker shift scheduling problem and the order picker planning problem of assigning and sequencing individual orders to order pickers. This requires decisions regarding the number of order pickers to schedule, shift start and end times, break times as well as the assignment and timing of order-picking activities. We call this the *Order Picker Scheduling Problem* and present two different formulations. A branch-and-price algorithm and a metaheuristic are developed to solve the problem. Numerical experiments illustrate that the metaheuristic finds near-optimal solutions at 80% shorter computation times. A case study at the largest supermarket chain in The Netherlands shows the applicability of the solution approach in a real-life business application. In particular, different shift structures are analyzed, and it is concluded that the retailer can increase the minimum compensated duration for employed workers from 6 hours to 7 or 8 hours while reducing the average labor cost with up to 5% savings when a 15-minute flexibility is implemented in the scheduling of break times.

Key words: facility logistics; personnel scheduling; order picker planning; branch-and-price; metaheuristic

1. Introduction

Scheduling order pickers is one of the fundamental decision problems in manual picker-to-part warehouses, where order pickers walk (or drive) to the storage locations of items to retrieve all the items specified in a picking list. The order picking process is one of the most labour-, time-and capital-intensive activities in warehouses, responsible for more than 50% of the operating costs (Tompkins et al. 2010). Despite the rise of automated order picking, less than 3% of the warehouses are fully automated and less than 10% of the warehouses use automated parts-to-picker systems (Michel 2016). Specifically, Azadeh et al. (2019) estimate that only 40 out of thousands of warehouses in Western Europe are fully automated. Consequently, manual order picking has been studied extensively in the literature and most research focuses on the development of travel time or distance models for various storage assignment, picking routing and order batching policies (Van Gils et al. 2018b). In contrast, the *order picker planning* problem which assigns and sequences orders to order pickers has hardly been studied (Van Gils et al. 2018b). This is an important problem for warehouses where orders have temporal restrictions such as deadlines. The assignment and the sequence of execution of orders have a direct impact on the tardiness of orders and on the costs associated with the order picking operations. Furthermore, as order pickers are humans, the order picker planning problem is further constrained by *shift scheduling decisions*, which include decisions regarding the start and end times of shifts and breaks, as well as the workforce level requirements for different shifts. The literature on order picker planning ignores these shift scheduling decisions and only considers a single shift horizon (i.e., shifts with one start and end time for all order pickers) without the need for breaks (Matusiak et al. 2014, Henn 2015, Scholz et al. 2017, Matusiak et al. 2017). Consequently, the available solution approaches in the literature can only be applied in a straightforward manner to manual order picker planning problems in warehouses where orders do not have temporal restrictions.

Many distribution centers in Western Europe face two main restrictions in the order picking operations: *due time windows* of orders and *flexible order pickers*. On-time retrieval of customer orders has become more important nowadays with companies offering deliveries to customers within a small time interval (e.g., one or two business days). To ensure that customer orders are delivered on time, trucks have departure deadlines from the warehouse. In retail logistics, these departure deadlines can also be imposed by strict city access time window regulations (Quak and de Koster 2007) and contractual agreements with retail stores (Bodnar et al. 2017). Besides these temporal restrictions, there are also spatial restrictions due to limited capacity at the outbound staging areas of warehouses to consolidate orders that need to be delivered by the same truck. Consequently, every order has a *due time window* during which it needs to be picked and sent to the allocated staging lane. These due time windows present severe challenges to warehouse managers in maintaining the right order picking workforce at the appropriate times. To cover demand during peak periods, a large number of order pickers is required. These order pickers can become superfluous when the volume of order picking tasks decreases. To alleviate this problem, warehouses employ *flexible order pickers* who can be called upon to work on short notice. The shift start and end times vary for these employees, but they are guaranteed a minimum payment equal to the payment corresponding to the minimum compensated duration, which is defined as the duration of time an order picker is paid for even if the order pickers is asked to work in a shift with a shorter shift length. Labour laws in many countries specify a minimum compensation duration. For instance, employees in the United States, Canada and Australia must be paid for at least 3 hours each time they are required to report to work. Under these circumstances, the aim of the warehouse manager is to solve the order picker planning problem such that due time windows of orders are respected while minimizing the labor cost. This requires them to determine how many order pickers to schedule (including the start times, end times, and breaks for each order picker), assign the orders that need to be picked during each shift, as well as the sequence in which the orders are picked by the order pickers. We call this optimization problem the *order picker scheduling problem* (OPSP). Most warehouse managers rely on their experience and intuition to make these decisions. Even though our study is inspired by the largest grocery retail chain in The Netherlands, the use of flexible order pickers with minimum compensation and one or multiple break periods is common in many countries. Our definitions of flexible order pickers and break requirements are fully compliant with the current European Union (EU) Directive 91/533/EEC (European Parliament, Council of the European Union 1991) as well as the new Directive (EU) 2019/1152 (European Parliament, Council of the European Union 2019) that will replace the current Directive in 2022. An overview of other labor laws around the world are included in Appendix D. Consequently, our study is generally applicable and relevant to many manual order picking warehouses where orders have tight due times and the resources to prepare orders (such as the number of staging lanes) and order pickers are limited.

In this paper, we combine the order picker planning problem with the shift scheduling problem to jointly determine the scheduling of start, end and break times for the shifts of flexible order pickers as well as the assignment and sequencing of orders with due time windows to these order pickers. The shift decisions have direct implications for the order picking process that should not be ignored. In Appendix A, an illustrative example is given that highlights the importance of explicitly constructing shifts that take breaks into account when orders have due time windows and order pickers are flexible. The contributions of our work are four fold: (i) We introduce the OPSP to the order picking literature and formulate the OPSP as a mixed integer linear program (MILP); (ii) To solve the problem, we present an exact branch-and-price algorithm in combination with an efficient heuristic to generate tight upper bounds based on the savings algorithm; (iii) We propose a computationally efficient metaheuristic that is capable of producing near-optimal solutions for large instances; (iv) A case study is performed to investigate the practical impact of flexible shift structures and show the impact can be substantial.

The outline of this paper is as follows. Relevant literature is reviewed in Section 2. Section 3 presents the problem description and the model formulation of the problem. In Section 4, we present a branch-and-price algorithm to find optimal solutions for the problem. A metaheuristic to solve the problem is proposed in Section 5. Results from computational experiments and the case study follow in Section 6. Finally, Section 7 concludes the paper.

2. Literature review

As identified in the previous section, the OPSP operates at the intersection of shift (or personnel) scheduling and order picker planning. More details on both research streams in the literature are provided in this section.

Order picker planning problem

When orders have temporal restrictions (such as due time windows) or when they result in penalties when completed early or late, the assignment of orders to order pickers and the sequencing to execute these orders have a direct impact on the feasibility of workforce schedules and the associated costs. Elsayed et al. (1993) and Elsayed and Lee (1996) are the first authors to study the *joint order batching and sequencing problem* (JOBSP) for a single automated storage and retrieval system (AS/RS) where the objective is to minimize the earliness and tardiness of orders. The authors suggest simple heuristic methods to generate solutions for the problem. Henn and Schmid (2013) and Henn (2015) extend this work to multiple order pickers, which is considered the *joint order batching, assignment and sequencing problem* (JOBASP). The authors suggest iterated local search and attribute-based hill climber, variable neighborhood search and variable neighborhood depth algorithms to solve this problem. Tsai et al. (2008) introduce a *joint order batching, assignment, sequencing, and routing problem* (JOBASRP), which is an extension of the JOBASP with routing decisions for the order pickers within the warehouse. Chen et al. (2015) and Scholz et al. (2017) propose heuristic solution approaches for this problem. Matusiak et al. (2014) investigate a variation of JOBASRP where the sequencing of batches is not relevant but the routing is part of the optimization problem which aims at minimizing the overall travel distance. In most applications, the storage racks are stationary, however, Boysen et al. (2017) consider an interesting variation with mobile rack warehouses, where an entire storage aisle may need to be moved to access items in it. Here, the objective is to sequence orders to minimize the number of aisle relocations.

In a recent review on order picking problems, Van Gils et al. (2018b) note that there is hardly any literature on the integration of the order assignment and sequencing decisions for order pickers (i.e., the order picker planning problem) while determining the order picking workforce (i.e., the shift scheduling problem). All work in the literature on scheduling manual order pickers assumes a single shift start and end time without the need for a break, which can be traced back to Elsayed et al. (1993) and Elsayed and Lee (1996). This simplifying assumption is only valid for machine environments or for manual order picking environments where a fixed number of order pickers can start and end their shift at only one given time, no breaks are scheduled and orders do not have temporal restrictions (as discussed in Section 1). When order pickers have fixed employment contracts, the shift scheduling decisions are typically made at a tactical level or at least before any order assignment and sequencing decisions are made. However, when order pickers have flexible employment contracts, it is crucial to make shift scheduling decisions at the same time as the order assignment and sequencing decisions are made. Figure 1 illustrates the typical order in which decisions are made in the two types of employment contracts. These differences require us to review shift scheduling literature which is done in the following. Note that batching is decoupled in both of the contracts because integrating optimal order batching with other decisions is computationally prohibitive in realistic settings. Furthermore, an appropriate batching policy alone can explain much of the variance in travel times of order pickers compared to related decisions (storage, zoning, and routing) (Van Gils et al. 2018a).

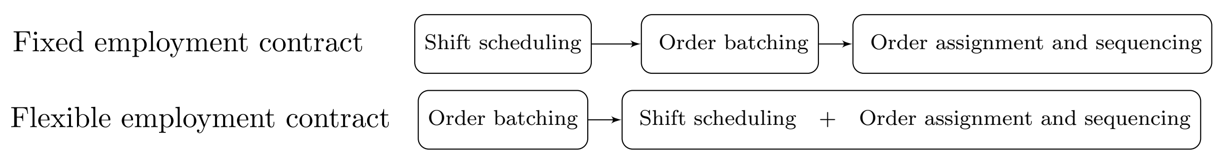


Figure 1 Sequence of decision problems with fixed and flexible employment contracts for order pickers

Shift (or personnel) scheduling problem

In contrast to the literature on order picking processes, the shift scheduling literature explicitly considers shift decisions as part of the planning problem. Shift scheduling is one of the oldest problems in the field of operations research. It dates back to Edie (1954) and Dantzig (1954) who scheduled toll booth operators and it has received a lot of attention in the literature since then (Ernst et al. 2004b,a, Van den Bergh et al. 2013). Many of the mathematical formulations are based on a generalized set covering model where each possible shift (i.e., a combination of start time, end time, and break placement) is represented by a decision variable. The goal is to determine the optimal complement of shifts such that operational constraints are satisfied while optimizing some objective function. It has applications in many industries including airlines, public transportation, hospitality, military, health care and call centers.

Shift scheduling problems can be divided into two broad categories based on the type of workload they consider: *workload-coverage problems* or *task-coverage problems*. The main distinction between these two categories relies on what is known prior to performing the personnel planning. In *workload-coverage problems*, the actual tasks that need to be to executed during the planning horizon are not known by the time personnel is scheduled. Consequently, the *demand* for employees is forecasted based on expected workloads and workers are scheduled to cover these predicted personnel demands. Employees are usually scheduled to perform one type of task that can be preempted between employees working in different shifts (e.g., manning a cash register in a shop).

In contrast to workload-coverage problems, the actual *tasks* that need to be executed are known in *task-coverage problems*. Besides the creation of shifts, these problems also include assignment decisions of tasks to individual employees or shifts such that all, or as many as possible, tasks are completed. Consequently, task-coverage problems are generally more complicated to solve than workload-coverage problems. Task-coverage problems can be further divided into two subcategories: fixed task timing problems and flexible task timing problems. In *fixed task timing problems*, the timing when to execute each task is known *a priori* (therefore, sequencing decisions are not included). These problems aim to generate schedules that cover the fixed tasks with a minimum number of machines or shifts. Examples of these problems are fixed job scheduling problems (Fischetti et al. 1987, 1989), interval scheduling problems (Kroon et al. 1995, Kolen et al. 2007) and shift minimization personnel task scheduling problems (Krishnamoorthy et al. 2012). The navy personnel planning problem studied by Holder (2005) is a closely related problem. Another example of the fixed task timing problem is the *integrated task scheduling and personnel rostering problem*, which generates the roster of employees while explicitly considering the coverage of tasks (Smet et al. 2016). Beliën and Demeulemeester (2008) present a branch-and-price algorithm for the integrated rostering problem of nurses while incorporating the scheduling of tasks that arise from surgery schedules. When the planning horizon of the fixed task timing problem is divided in periods and the duration to execute each task is equivalent to the length of a period, this is known in the literature as the *multi-activity shift scheduling problem* (Côté et al. 2011, Elahipanah et al. 2013, Dahmen et al. 2018).

In *flexible task timing problems*, the timing when to execute tasks is a decision. Consequently, sequencing decisions have to be made besides the shift scheduling and task assignment decisions. For instance, home care workers are assigned to locations where tasks (such as cooking, cleaning and administering medicine) have to be performed within specific time windows (Rasmussen et al. 2012). Closely related problems include the *field workforce scheduling problem* where individual workers with appropriate skills are assigned to geographically distributed tasks (Alsheddy and Tsang 2011) and the *technician task scheduling problem* where individuals with the correct skill mix are assigned to tasks of different priorities (Cordeau et al. 2010, Fırat and Hurkens 2012).

Shift scheduling problem with breaks

The inclusion of breaks in the personnel scheduling literature is mainly limited to workload-coverage problems only. Thompson (1988) is one of the first authors to explicitly plan for breaks when shift schedules are generated. In its simplest form, the set covering formulation of Dantzig (1954) is extended with additional decision variables to represent breaks and reliefs. For problems involving a high degree of flexibility with respect to the timing of breaks, the number of enumerated shifts increases drastically and the resulting set covering problem can be very difficult to solve (if even possible). To overcome these challenges, Bechtold and Jacobs (1990) propose a compact formulation that implicitly considers breaks, but is tractable for realistic instances. This model is extended by Thompson (1995) to consider different types of breaks and even overtime. Aykin (1996) also presents a compact integer programming model that is capable of considering time windows for multiple breaks in one shift. Aykin (2000) shows that this model is computationally superior compared to the formulation in Bechtold and Jacobs (1990), who only consider one break in a shift. Sungur et al. (2017) present a goal programming approach for the same problem as studied by Aykin (1996).

The work of Bechtold and Jacobs (1990) is also extended by Brusco and Jacobs (2000) to introduce break and relief planning in *tour scheduling problems*. In this type of problem, the aim is to generate a schedule with multiple shifts for each employee as well as off days during which the employee is not working. Consequently, the planning horizon is longer compared to shift scheduling problems. Bard et al. (2007) also model a tour scheduling problem with break and labor rules but in a stochastic environment of a parcel sorting center. Gérard et al. (2016) present a heuristic that is based on column generation for a more extensive problem which simultaneously considers off days, shift scheduling, shift assignments and task assignments within shifts. A key difference of these problems compared to our OPSP is that the tasks have a fixed timing rather than a time window during which they need to be performed.

For flexible task timing problems, the scheduling of breaks is only included in the *truck driver scheduling problem*. In these problems, the sequence in which locations are visited by trucks has to be determined while satisfying appropriate time windows. The maximum amount of time a truck driver is allowed to be on the road is restricted such that breaks and rest periods have to be considered to satisfy the strict hours-of-service regulations (Goel 2010, Goel and Kok 2012). The truck driver scheduling problem is extended to vehicle routing decisions in Goel and Irnich (2017). In these studies, the objective of the problem is to minimize the travel distance. An alternative objective function for the problem is presented by Tilk and Goel (2020), where the problem aims to minimize the number of working days for a given route instead of the travel distance.

A comparison between OPSP and the available literature on shift scheduling problems can be found in Table 1. It becomes clear that the order picker planning problem does not consider shift scheduling decisions when orders have temporal restrictions. Furthermore, most of the flexible task timing problems in the shift scheduling literature do not consider the characteristics that are unique

Table 1 Comparison of OPSP to the shift scheduling and order picker planning literature

Type of problem (representative paper)	Coverage: Task	Coverage: Workload	Task timing: Fixed	Task timing: Flexible	Task timing: Window	Break timing	Minimum compensated duration
Parcel sorting center scheduling (Bard et al. 2007)		✓				✓	
Order assignment sequencing (Scholz et al. 2017)	✓			✓	✓		
Fixed job scheduling (Fischetti et al. 1989)	✓		✓				
Interval scheduling (Kroon et al. 1995)	✓		✓				
Shift minimization (Krishnamoorthy et al. 2012)	✓						
Nurse rostering and task scheduling (Beliën and Demeulemeester 2008)	✓		✓				
Home care scheduling (Rasmussen et al. 2012)	✓			✓	✓		
Field workforce scheduling (Alsheddy and Tsang 2011)	✓			✓	✓		
Technician task scheduling (Cordeau et al. 2010)	✓			✓			
Call center scheduling (Bhandari et al. 2008)		✓				✓	
Hotel staff scheduling (Thompson and Pullman 2007)		✓				✓	
Navy personnel planning (Holder 2005)	✓		✓			✓	
Tour scheduling (Brusco and Jacobs 2000)	✓		✓			✓	
Multi-activity shift scheduling (Dahmen et al. 2018)	✓	✓				✓	
Truck driver scheduling (Goel and Irnich 2017)	✓			✓	✓	✓	
Order picker scheduling problem	✓			✓	✓	✓	✓

to warehouse environments (i.e., tasks with due time windows in combination with flexible workers who require breaks and a minimum payment).

In the shift scheduling literature with task assignments that have to be performed in a certain time window, only the truck driver scheduling problem considers breaks. It is therefore the closest related to our problem formulation. The break requirements for truck drivers considered in Goel and Irnich (2017) are similar to the break requirements for order pickers considered in the OPSP. However, a major difference is that Goel and Irnich (2017) focus on minimizing travel distances, whereas schedule durations do not play a role. The objective in Tilk and Goel (2020) is to minimize the sum of labor costs and distance-related costs whereas labor costs are related to the number of working days required to complete the route. The number of hours worked within a working day does not play a role and most schedules generated actually include long periods of waiting. The OPSP studied in our work combines elements of minimizing schedule duration with minimum compensated duration which make the OPSP structurally different from the aforementioned problems and it necessitates new solution approaches. Our work addresses this gap in the literature and combines the order picker planning literature and shift scheduling literature. In Section 4, we further explain the differences between our solution approach to solve the OPSP compared to other approaches in the literature.

3. Problem description and model formulation

In this section, we explain the warehouse operations that define our order picker scheduling problem (OPSP) and we formulate the corresponding mixed integer linear program (MILP) model. Symmetry breaking constraints and additional constraints to tighten the model formulation are included in Appendix B. An alternative formulation of the problem as a network flow problem is presented in Appendix C. This formulation takes more computational effort to solve in our numerical experiments, it is therefore included as reference only. Extending the OPSP formulation with full-time order pickers and different types of break time constraints are discussed in Appendix D.

Production facilities or retail stores place orders to receive items from a distribution warehouse based on their needs. An *order* is composed of multiple order lines, where each order line consists of a particular item and the corresponding requested quantity. The order lines that should be processed together create a *pick list*. The list contains all items that need to be picked and it guides the order picker through the warehouse. Items that are collected are put in roll cages such that products in the same roll cage are sent to a single customer. However, a customer's order can result in multiple roll cages picked by one or more order pickers. An order picker's tour finishes when all roll cages from the pick list are delivered to the corresponding staging lanes at the outbound docks. The total number of order lines and roll cages can exceed hundreds, which prohibits a joint optimization of the personnel scheduling and order batching problems within a reasonable computational effort. Consequently, we assume that order batching (i.e., the construction of pick lists) is done *a priori*. In the remainder of the paper, we use the term *batch* to refer to a pick list that is to be completed by a single order picker in a single pick tour.

Let I be the set of batches that are generated *a priori*. The time required to pick batch $i \in I$ is denoted by t_i. It includes the time for an order picker to travel between product locations of items in the batch, search for the items, place them in roll cages, and transport the filled roll cages to the staging lanes. We assume that t_i is independent of the order picker and its value is deterministic since the picking route is determined by the storage locations of the items in the batch and the routing strategy of the warehouse. The company in our case study uses norm times that are set to pick a certain batch.

Each batch $i \in I$ has a corresponding delivery due time window $[r_i, d_i]$. All items in the batch have to be delivered to the designated staging area(s) within this time window. The values of r_i and d_i are determined based on the outbound truck departure schedule and the capacity of the staging lanes. The value of r_i usually corresponds to the departure time of the previous vehicle that departed from the same staging lane as where the vehicle for batch i is departing from, and d_i is the latest time batch i can be delivered at the staging lane for the vehicle to depart on time (i.e., without violating the delivery due time at the customer).

Let P represent the set of the flexible order pickers that can be employed by the warehouse, where $|P| = p_{max}$. Flexible workers are scheduled to work when needed, and as such, they are assigned one of a variety of possible shift lengths with different start times on any day. They are only compensated for the amount of time they spend at the warehouse. Although there is often no restriction on the minimum shift length for a worker, warehouses favor providing a minimum compensation if an employee is scheduled to work. This improves the working relation between the flexible order pickers and the warehouse to increase employee retention. The time corresponding to the minimum compensation duration is denoted by T_{min}. The maximum amount of time an employee can work per day is restricted by law and gives an upper limit on the shift length, which we denote by T_{max}.

There are also labor rules and union agreements on breaks for human order pickers. The amount of time an employee can work without a break is denoted by T_{break}. If an employee works for a duration that exceeds T_{break} time units she must be given an uninterrupted break of at least l_b time units. An employee can be entitled to more than one break in the same shift depending on the values of T_{break} and T_{max}. The length of the planning horizon is T_{day} time units. Formulations for alternative types of breaks are presented in Appendix D. We assume that order picking is scheduled non-preemptively and breaks cannot interrupt this. Interrupting a pick tour and leaving picking equipment in the storage area creates congestion as well as safety and security hazards. Limited parking space for order picking equipment in the break areas and issues of theft or responsibility of already picked items may also prevent preemptive batch scheduling. In case breaks can preempt the order picking of a batch, we propose an updated solution framework and perform a numerical comparison in Appendix K.

Even though flexible employees can potentially start and end their shifts at any time, many shift start and end times are an administrative and operational burden, and labor union agreements can prohibit this as well (Brusco and Jacobs 1998). Furthermore, employees are paid in integral multiples of a certain duration (even if they completed the last task of their shift before the end of a certain time period). Therefore, we divide the planning horizon into W time periods of equal length, where each period consists of l time units. The set of admissible time periods to start or end a shift is denoted by S and E, respectively. Note that the discretization of the time horizon is only used for the start and end times of shifts. The actual tasks that need to be executed can still start and end at any point in time during the shift (i.e., they do not have to coincide with the time periods) and the same holds for breaks.

We make the assumption that all order picking operations associated with the batches in the planning horizon are performed within the same planning horizon, and we assume that all shifts of the order pickers start and end in the same planning horizon that they are scheduled for (i.e., there is no overlap between either order picking tasks of a batch or shifts of order pickers in different planning horizons).

Furthermore, we define a *task* as an activity that needs to be scheduled; either picking orders of a batch or taking a break. Arranging tasks in a sequence creates a shift, and each task in the sequence has a position (first, second and so on). This is illustrated with a Gantt chart in Figure 2.

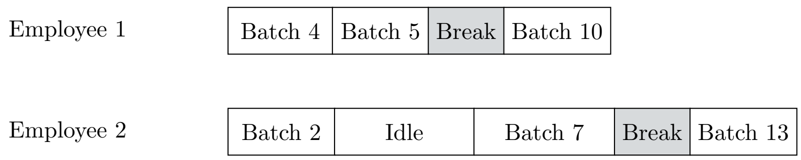


Figure 2 **A Gantt chart to illustrate the concepts of tasks, shifts and task positions in a sequence**

Table 2 **Overview of the parameters for the order picker scheduling problem (OPSP)**

notation	description
P	set of order pickers that can be scheduled, $\{1, \ldots, p_{max}\}$
I	set of batches that need to be picked
K	set of positions in which an order picker can perform a task, $\{1, \ldots, \bar{k}\}$
t_i	duration to pick and deliver the items of batch $i \in I$
r_i	earliest due time of batch $i \in I$
d_i	latest due time of batch $i \in I$, where $d_i \geq \max\{r_i, t_i\}$
T_{min}	minimum time an order picker needs to be compensated if scheduled
T_{max}	maximum shift length
T_{break}	maximum time duration an order picker can work consecutively without a break
T_{day}	length of the planning horizon
l	length of a time period
J	set of time periods, $\{1, \ldots, W\}$
S	set of time periods where a shift can start at the beginning of that period, $S \subseteq J$
E	set of time periods where a shift can end at the end of that period, $E \subseteq J$
l_b	duration of a break
M	a very large number

Employee 1 picks the items in batch 4 and 5 successively, then takes a break, and finally picks items in batch 10 before ending her shift. Note that the order picker completes four tasks but not necessarily consecutively (i.e., there can be an interruption or gap between two successive tasks), which is the case for Employee 2. Each order picker can perform at most $\bar{k}$ tasks in a shift. A summary of all parameters is provided in Table 2.

The following decision variables are used in our model formulation:

- x_{ikp} is 1 if batch $i \in I$ is scheduled to be picked at the k^{th} position in the shift for order picker $p \in P$, where $k \in K$, else 0
- y_{kp} is 1 if a break is scheduled at the k^{th} position in the shift for order picker $p \in P$, where $k \in K$, else 0
- s_{jp} is 1 if order picker $p \in P$ starts the shift at the beginning of period $j \in S$, else 0
- e_{jp} is 1 if order picker $p \in P$ ends the shift at the end of period $j \in E$, else 0
- c_{kp} completion time of the task scheduled at the k^{th} position in the shift for order picker $p \in P$, where $k \in K$
- m_p amount of time for which order picker $p \in P$ is compensated

The order picker scheduling problem (OPSP) is formulated as a MILP model as follows:
OPSP:

$$\min \sum_{p \in P} m_p \tag{1}$$

subject to

$$\sum_{i\in I} x_{ikp} + y_{kp} \le 1 \qquad \forall k \in K, p \in P \tag{2}$$

$$\sum_{k\in K}\sum_{p\in P} x_{ikp} = 1 \qquad \forall i \in I \tag{3}$$

$$\sum_{i\in I} x_{i1p} + y_{1p} \le \sum_{j\in S} s_{jp} \qquad \forall p \in P \tag{4}$$

$$\sum_{j\in S} s_{jp} = \sum_{j\in E} e_{jp} \qquad \forall p \in P \tag{5}$$

$$c_{1p} \ge \Big(\sum_{j\in S}(j-1)s_{jp}\Big)l + \sum_{i\in I} t_i x_{i1p} + l_b y_{1p} \qquad \forall p \in P \tag{6}$$

$$c_{kp} \ge c_{k-1,p} + \sum_{i\in I} t_i x_{ikp} + l_b y_{kp} \qquad \forall k \in K\setminus\{1\}, p \in P \tag{7}$$

$$\sum_{j\in E}(je_{jp})l \ge c_{\bar{k}p} \qquad \forall p \in P \tag{8}$$

$$c_{kp} + M(1 - x_{ikp}) \ge r_i \qquad \forall i \in I, k \in K, p \in P \tag{9}$$

$$c_{kp} - M(1 - x_{ikp}) \le d_i \qquad \forall i \in I, k \in K, p \in P \tag{10}$$

$$c_{kp} - \Big(c_{hp} - \sum_{i\in I} t_i x_{ihp}\Big) \le T_{break} + M\Big(\sum_{k'=h+1}^{k} y_{k'p}\Big) \qquad \forall h, k \in K, h < k, p \in P \tag{11}$$

$$\sum_{i\in I} x_{ik-1,p} + y_{k-1,p} \ge \sum_{i\in I} x_{ikp} + y_{kp} \qquad \forall k \in K\setminus\{1\}, p \in P \tag{12}$$

$$T_{min}\sum_{j\in S} s_{jp} \le m_p \qquad \forall p \in P \tag{13}$$

$$\Big(\sum_{j\in E} je_{jp} - \sum_{j\in S}(j-1)s_{jp}\Big)l \le m_p \qquad \forall p \in P \tag{14}$$

$$c_{kp} \ge 0 \qquad \forall p \in P, k \in K \tag{15}$$

$$x_{ikp} \in \{0,1\} \qquad \forall i \in I, k \in K, p \in P \tag{16}$$

$$y_{kp} \in \{0,1\} \qquad \forall k \in K, p \in P \tag{17}$$

$$s_{jp} \in \{0,1\} \qquad \forall j \in S, p \in P \tag{18}$$

$$e_{jp} \in \{0,1\} \qquad \forall j \in E, p \in P \tag{19}$$

$$0 \le m_p \le T_{max} \qquad \forall p \in P \tag{20}$$

The objective function (1) expresses the minimization of the total labor cost over all order pickers who are scheduled to pick the items that need to be delivered during the planning horizon. Constraints (2) ensure that an order picker can perform at most one task in the k-th position of her shift. Constraints (3) ensure that each batch is picked exactly once. An order picker can only perform the first task in a shift if she is scheduled to start a shift according to constraints (4). Constraints (5) ensure that every order picker who starts a shift also has to end a shift (and vice versa).

Constraints (6) and (7) determine that the task in the k-th position of the order picker's shift can only be labeled as completed after it is executed. Constraints (8) ensure that the order picker can only finish her shift after completing the last assigned task. Constraints (9) and (10) require that batches are completed within their due time windows. Note that an order picker can have fewer than $\bar{k}$ tasks assigned to her shift. In that case, for all positions in a shift without an actual task assigned (i.e., for all k where $\sum_i x_{ikp} + y_{kp} = 0$), the completion times c_{kp} are set equal to the completion time of the last assigned task (i.e., $c_{kp} = c_{k-1,p}$).

Constraints (11) require that an order picker cannot work successively for a duration more than T_{break} time units without a break. The constraint specifies that the time between the start of the task at position h of the shift and the end of the task at position k, where $k > h$, has to be less than or equal to T_{break} in case no break is scheduled between these two tasks. Constraints (12) specify that a task can only be assigned to a position if there is also a task assigned to the previous position.

Constraints (13) ensure that an order picker is compensated for at least T_{min} time units if she is scheduled to work. Constraints (14) ensure that an order picker is compensated for at least the amount of time the order picker is scheduled to work (i.e., from the start time of the shift to the end time of the shift). Constraints (15) to (20) define the domain and range of the decision variables.

PROPOSITION 1. *Generating a feasible solution for the OPSP is NP-hard in the strong sense.*

Proof. $P||C_{MAX}$ problem is a special case of the OPSP.

4. Branch-and-price algorithm for OPSP

This section outlines an exact procedure to solve the OPSP using a branch-and-price framework. In this solution approach, the linear relaxation in each node of a branch-and-bound tree is solved with column generation (Barnhart et al. 1998, Vanderbeck 2000). A branch-and-price solution approach remains a successful and popular solution strategy for generating optimal solutions for problems in a variety of fields ranging from transport planning (Bertsimas et al. 2019), routing (Dellaert et al. 2018) to personnel scheduling (Van den Bergh et al. 2013). We also develop a branch-and-price algorithm for the order picker scheduling problem. We first present the *reduced master problem* (RMP). The *pricing problem* to verify the optimality of an LP solution is presented in Section 4.2. The branching that occurs when the LP solution does not satisfy the integrality conditions is discussed in Section 4.3.

The proposed framework for the branch-and-price algorithm has similarities to the one used by Goel and Irnich (2017). However, because we use the schedule duration in the objective function (which includes employee waiting times between the performance of two tasks) and include the minimum compensated duration as constraints, the details of the building blocks for the branch-and-price algorithm are different from the algorithm in Goel and Irnich (2017). Specifically, the augmented graph for the pricing problem requires information on shift starting and ending times. The definitions of resources and resource extension functions that are used to solve the pricing problem also differ and are more comparable to those used for the minimum tour duration problem (MTDP) (Tilk and Irnich 2017) rather than the truck driver scheduling problem. Furthermore, because of the constraints regarding the minimum compensated duration and flexible breaks, the problem suffers from significant issues of symmetry. Therefore, we develop a tailored acceleration strategy to address these issues (see the end of Section 4.2).

4.1. Reduced master problem

To present the reduced master problem for the OPSP in a column generation format, we first introduce the concept of a *column* as a *feasible shift schedule* that is specified by the start and end time as well as the assignment and sequence of tasks (both order picking and breaks) to be performed by a single order picker while respecting the due time windows of order picking tasks, maximum shift length T_{max} and maximum time between breaks T_{break}. Let Ω denote a set of all feasible schedules, where Ω' is a subset of Ω (i.e., $\Omega' \subseteq \Omega$). The cost for an individual schedule $q \in \Omega'$ is given by m_q. The parameter α_{iq} is set to 1 if batch i is processed (or picked) in schedule q, and zero otherwise. The decision variable θ_q represents the number of schedules of type q to be selected in the solution. The reduced master problem (RMP) can be formulated as a set covering problem:

RMP:

$$\min \sum_{q \in \Omega'} m_q \theta_q \tag{21}$$

subject to

$$\sum_{q \in \Omega'} \alpha_{iq} \theta_q \geq 1 \qquad \forall i \in I \tag{22}$$

$$\sum_{q \in \Omega'} \theta_q \leq p_{max} \tag{23}$$

$$\theta_q \geq 0 \qquad \forall q \in \Omega' \tag{24}$$

The objective in the RMP is the same as in the OPSP. Constraints (22) ensure that all batches are processed (or covered) with the selected schedules. Constraints (23) do not select more than p_{max} schedules to be performed by order pickers. The constraints (2) and (4) to (20) of the OPSP are included in the pricing problem where columns are generated that result in feasible schedules.

4.2. Pricing problem

The pricing problem for the OPSP can be formulated as an Elementary Shortest Path Problem with Resource Constraints (ESPPRC) (Feillet et al. 2004). This is a variation of the Shortest Path Problem with Resource Constraints (SPPRC) where cycles are not allowed, i.e., a node cannot be visited more than once. The SPPRC can be solved with pseudo-polynomial algorithms (Irnich and Desaulniers 2005), whereas the ESPPRC is NP-hard in the strong sense (Dror 1994). Nevertheless, ESPPRC is known to generate a superior lower bound compared to SPPRC when used as pricing problem (Contardo et al. 2015). A technique to solve the ESPPRC is a labeling algorithm based on dynamic programming (Feillet et al. 2004). This approach uses the concepts of resources in a graph and resource extension functions. A *resource* is an arbitrary one-dimensional piece of information that can be determined or measured at the vertices of a directed walk in a graph (e.g., cost, time, load). In this paper, time is the main resource. *Labels* are used to store the information on the resource values for partial paths. Labels reside at vertices and they are propagated via *resource extension functions* when they are extended along an arc. To keep the number of labels as small as possible, we define *dominance rules* to identify labels that need not be extended. We first introduce the graph structure, labels, resource extension functions and dominance rules.

Graph representation Consider a subgraph $G = (V, A)$, where V is the set of vertices indicating the set of batches $i \in I$ that have to be picked and the arcs A indicate the subsequent sequence in which the batches are completed. The nodes in the sets S and E indicate the start and end times of a shift, respectively. Furthermore, dummy origin and destination nodes are indicated by $\mathtt{o}$ and $\mathtt{d}$, respectively. The complete set of all vertices is $V' := \{\mathtt{o}\} \cup S \cup V \cup E \cup \{\mathtt{d}\}$.

We introduce arcs between the dummy origin node $\mathtt{o}$ and the shift start time nodes in S, between each vertex in S and V, between each vertex in V and E as well as between the shift end time nodes and the dummy destination node $\mathtt{d}$. See Figure 3 for an example. The travel time for each arc is set to zero. The service time t_i at each node $i \in V$ equals the processing time of batch i, whereas the service time at the remaining vertices $V' \setminus V$ is zero.

The time windows for the origin and destination nodes are $[r_i\,;\,d_i] = [0\,;\,T_{day}]$ for $i \in \{\mathtt{o}, \mathtt{d}\}$ such that these nodes can be visited at any time during the time horizon. For the shift start time nodes $i \in S$, the value of $r_i = d_i$ equals the possible shift start times such that these nodes are visited at these specific times. Similarly, for the shift end time nodes $i \in E$, the value of $r_i = d_i$ equals the possible shift end times. A feasible schedule for an order picker comprises of a tour from node $\mathtt{o}$ to node $\mathtt{d}$ respecting the due time windows $[r_i\,;\,d_i]$ for $i \in V'$, maximum shift length T_{max} and the time until breaks T_{break}. As an illustrative example, Figure 3 represents a graph where there are three possible shift start and end times. The dashed arrow indicates a feasible schedule that starts at $s1$, then executes batch $i3$ and ends at $e1$.

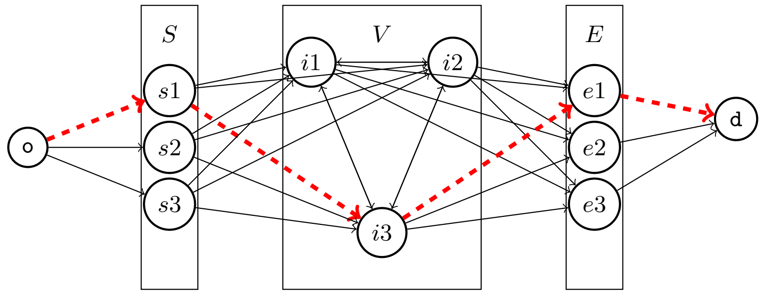


Figure 3 **A representation of a graph structure for the pricing problem of the OPSP with 3 shift start and end times and 3 batches**

Labels A *partial schedule* corresponds to a partial path in the graph G. A partial schedule h where vertex i is visited as last node is defined by label $L_h^i = (i, c_h^i, T_i, (V_h^1, \ldots, V_h^{|V|}))$, where

- i is the last vertex that has been visited in the partial schedule
- c_h^i is the reduced cost of the partial schedule (i.e., the actual cost minus the dual values of the nodes visited, see below for more details)
- $T_i = (T_i^{time}, T_i^{dur}, T_i^{start}, T_i^{work}, T_i^{brk})$ indicates the resource vector, where the resource variables are
  - T_i^{time} is the time when the batch at node i is completed
  - T_i^{dur} is the minimum duration required to service all the nodes in the partial schedule including the waiting times if necessary to respect the due time windows
  - T_i^{start} is the latest possible start time of the shift to feasibly visit all of the vertices in the partial schedule while respecting the due time windows
  - T_i^{work} is the amount of time since the end of the last break
  - T_i^{brk} is the latest time to start picking the first batch after the previous break to ensure feasibility of the schedule
- V_h^v is 1 if node $v \in V$ is visited in the partial schedule or if it is infeasible to visit (due to the due time windows or maximum shift length), 0 otherwise

To guarantee elementarity of a (partial) path, it is sufficient to add the extra resources V_h^v for each node $v \in V$ indicating whether or not the node has been visited on the path. When this resource has the value one, it prohibits the path to re-enter previously visited nodes. Feillet et al. (2004) enhance this idea by observing that some nodes are not reachable due to the resource constraints, which they indicate by setting the resources V_h^v to one for these nodes without the path having to visit them. They use this to speed up the dominance check, which is explained later in this section.

The resource windows of resource vector T_i are given by $T_i^{time} \in [r_i\,;\, d_i]$, $T_i^{dur} \in [0\,;\, T_{max}]$, $T_i^{start} \in (-\infty\,;\, T_{day}]$, $T_i^{work} \in [0\,;\, T_{break}]$, and $T_i^{brk} \in (-\infty\,;\, \infty)$. A path is called resource-feasible if there exist resource vectors for each node in the path that satisfy their resource windows. Therefore, a feasible schedule is a resource-feasible path that starts in $\mathtt{o}$ and ends in $\mathtt{d}$. Furthermore, let $\mathscr{L}_i$ denote the set of all labels corresponding to partial schedules where node $i \in V'$ is the last visited node.

The initialization of a label is done at shift start nodes $i \in S$ as $L_h^i = (i, T_{min} - \psi, T_i, (V_h^1, \ldots, V_h^{|V|}))$, where ψ is the dual variable associated with Constraint (23) of the RMP, $T_i^{time} = d_i$, $T_i^{dur} = 0$, $T_i^{start} = d_i$, $T_i^{work} = 0$ and $T_i^{brk} = \infty$, and $V_h^j = 0$ for all nodes $j \in V$.

Resource extension functions A resource extension function (REF) is used to extend a label (or partial schedule) with an additional vertex such that all constraints related to the scheduling problem are still satisfied. There are two options to extend label L_h^i at vertex i to vertex j when $V_h^j = 0$. The first extension executes the batch in node j directly after finishing the batch in node i without a break. The second extension starts with a break before the execution of the batch in node j. Consequently, we consider the two resource extension functions $f(\cdot)$ and $g(\cdot)$, respectively.

The resource extension functions $f(T_i, j)$ for the extension of label L_h^i to node j without a break define the new resource variables of resource vector T_j as follows

$$T_j^{time} = f^{time}(T_i, j) := \max\{T_i^{time} + t_j, r_j\} \tag{25}$$

$$T_j^{dur} = f^{dur}(T_i, j) := \max\{T_i^{dur} + t_j, r_j - T_i^{start}\} \tag{26}$$

$$T_j^{work} = f^{work}(T_i, j) := \max\{T_i^{work} + t_j, r_j - T_i^{brk}\} \tag{27}$$

$$T_j^{brk} = f^{brk}(T_i, j) := \min\{d_j - (T_i^{work} + t_j), T_i^{brk}\} \tag{28}$$

Similarly, when label L_h^i is extended with a break before the order picking task is completed as indicated by node j, the resource extension functions $g(T_i, j)$ define the resource vector T_j as

$$T_j^{time} = g^{time}(T_i, j) := \max\{T_i^{time} + t_j + l_b, r_j\} \tag{29}$$

$$T_j^{dur} = g^{dur}(T_i, j) := \max\{T_i^{dur} + t_j + l_b, r_j - T_i^{start}\} \tag{30}$$

$$T_j^{work} = g^{work}(T_i, j) := t_j \tag{31}$$

$$T_j^{brk} = g^{brk}(T_i, j) := \min\{d_j - t_j, \infty\} \tag{32}$$

Note that the resource variable T_i^{start} is never updated after it is set in the shift start node. The REF for T_j^{time} is a classic REF from the routing literature (Irnich 2008). The REFs for T_j^{dur} and T_i^{start} bear resemblance to REFs from the MTDP (Tilk and Irnich 2017). The REFs for T_j^{work} and T_j^{brk} are new and specifically designed to determine the amount of time elapsed since the last break. Desaulniers and Villeneuve (2000) use similar extension functions to estimate the cost of waiting at nodes for the shortest path problem with time windows and linear waiting costs.

The reduced cost for the partial schedule when label L_h^i is extended to node j, is given by $c_h^j := \max\{T_{min}, T_j^{dur}\} - \pi_j - \psi - \sum_{\hat{j} \in \hat{B}_h} \pi_{\hat{j}}$, where π_j is the dual value of constraints (22) for vertex j, ψ is the dual value associated with constraint (23), and $\sum_{\hat{j} \in \hat{B}_h} \pi_{\hat{j}}$ indicates the accumulated dual values associated with constraints (22) for the set of batches previously added to the partial schedule represented by the set $\hat{B}_h$. Note that the payment to pickers for the entire period (even if they work only for a fraction of the period) is accounted for by the use of shift end nodes which restrict the visit to the shift end nodes at the end of a period.

The resource V_h^j is set to one to prevent that vertex j is visited again. Furthermore, $V_h^{j'}$ is also set to one for any node $j' \in V'$ that cannot be visited anymore when node j is added to the partial path because of the resource constraints. The new label is then given by $L_h^j := (j, c_h^j, T_j, V_h)$, which is only feasible if the resource variables of the resource vector T_j fall within the associated resource windows.

Dominance A dominance principle can be used to accelerate the solution technique by eliminating unnecessary labels. To define dominance in our pricing problem, we note that the REFs are either non-decreasing or non-increasing, such that an element-wise comparison can be made to determine dominance (Irnich and Desaulniers 2005). A label L_h^i dominates a label $L_{h'}^i$ if both labels reside at the same vertex $i \in V'$ and if, for each feasible extension of $L_{h'}^i$ to $L_{h'}^j$, there exists a feasible extension of L_h^i to L_h^j where the value of each resource with a non-decreasing (or non-increasing) REF is less than (or larger than) or equal to the value of the resource in the extension of $L_{h'}^i$, i.e., $c_h^i \leq c_{h'}^i$; $T_{i,h}^{time} \leq T_{i,h'}^{time}$; $T_{i,h}^{dur} \leq T_{i,h'}^{dur}$; $T_{i,h}^{start} \geq T_{i,h'}^{start}$; $T_{i,h}^{work} \leq T_{i,h'}^{work}$; $T_{i,h}^{break} \geq T_{i,h'}^{break}$; $V_h^v \leq V_{h'}^v \, \forall v \in V$. Consequently, the partial schedule corresponding to label $L_{h'}^i$ cannot be part of the optimal solution. Note that the differentiation of time resources T_i for label h and h' is done for comparison required by dominance. For ease of notation, we do not use the differentiation of time resources for specific labels in the remainder of the paper.

Labeling algorithm The pricing problem is solved by embedding the resource definitions, resource extension functions and dominance rules in the label correction algorithm by Feillet et al. (2004). The pseudocode for the labeling algorithm is presented in Appendix E .

Acceleration strategies Acceleration strategies are commonly used to speed up branch-and-price algorithms and are key to successfully solving sizable problems (Kallehauge et al. 2005). We propose three acceleration strategies for the pricing problem.

Initial columns: It is known that column generation with good initial upper bounds accelerates the convergence of the linear relaxation at the root node (Desaulniers et al. 2002). Therefore, we first generate initial primal solutions with the savings algorithm outlined in Section 5.1. This algorithm aims to rapidly find a feasible solution. If the savings algorithm is not able to generate a feasible solution with at most p_{max} order pickers, the initial columns for column generation are initialized with an additional artificial column that covers all batches of the problem and has an arbitrarily high cost to ensure that this artificial column will not be part of the optimal solution.

Limited extension: To exploit the time windows and processing time information between order picking tasks to reduce the use of REFs, we first present the following proposition. The proof of this proposition is presented in Appendix F.

PROPOSITION 2. *If there is an optimal schedule with two batches i and j such that $r_i \geq r_j$, $d_i \geq d_j$, $t_i < t_j$ and i precedes j in the same order picker's schedule without a break in between the execution of the two batches, the execution order can be reversed with the same objective function value.*

For any partial schedule h ending with node i (i.e., presented by label L_h^i), if there is a node j for which $V_h^j = 0$ and the conditions in Proposition 2 satisfy, we only have to consider the extension with a break between the execution of the batches from node i and j. Consequently, we limit the extension of resources in the arc (i, j) with the resource extension function $g(\cdot)$ only. This particular strategy allows us to have fewer extensions and maintain a smaller set of labels while solving the pricing problem.

Limited discrepancy search: Desaulniers et al. (2008) and Spliet et al. (2018) show that the branch-and-price algorithm can be solved more efficiently when the pricing problem is solved with heuristics until no negative reduced costs are found (such that no new columns are added to the RMP). Along the same principle, as proposed by Feillet et al. (2007) and Goel and Irnich (2017), we use limited discrepancy search (LDS) to heuristically accelerate the generation of columns with a negative reduced cost.

LDS speeds up the pricing problem by maintaining a limited set of labels and heuristically removing so-called unpromising labels from the problem. In our pricing problem, labels with batches that require large waiting times and numerous breaks are considered unpromising labels. The waiting time between two nodes i and j is measured as the time window distance $TW_{distance}(i, j) := \max\{0, r_j - d_i\}$. The outgoing arcs from each node i are partitioned into two sets called *good arcs* and *bad arcs* based on $TW_{distance}$. An additional resource (denoted as l^{bad}) is included to the label that is increased by one if a label traverses through an arc from the set of bad arcs or if it is extended with a break. Only labels that have $l^{bad} \leq \Lambda$ are extended, where the threshold Λ is called the *discrepancy limit*. If the LDS is unable to find any columns with a negative reduced cost, the value of Λ increased by one and the LDS is repeated. When the discrepancy limit reaches an upper bound, the LDS terminates and the ESPPRC is solved with the labeling algorithm. Additionally, an iteration of LDS is terminated if 100 columns with negative reduced cost are generated. Note that the use of LDS does not impact the optimality of the branch-and-price technique since the last pricing problem at every node of the branch-and-bound tree is solved exactly with ESPPRC.

4.3. Branching

If the pricing problem cannot find columns with a negative reduced cost and the LP solution to the RMP is not integral, a node of the branch-and-bound tree is selected for branching. Branching is done on flow variables using the best-lower-bound-first strategy (Desaulniers 2010).

A good upper-bound solution improves the efficiency of the branch-and-price technique by reducing the number of branch nodes in the search tree (Danna and Le Pape 2005). In our solution procedure, before branching from the root node, we solve the MIP of the RMP where we only consider the columns that are generated at the root node. The solution to the MIP provides the upper bound before branching. If the root node is not solved within the time limit, the MIP of the RMP is solved with the available columns to derive the best known upper-bound solution for benchmarking purposes.

5. Metaheuristic for OPSP

Given the size of real-world instances of the OPSP and the computational complexity of the problem, even the branch-and-price technique developed in the previous section is not likely to be a viable solution approach in real-life applications. In this section, we present an efficient metaheuristic that adapts the classic savings principle by Clarke and Wright (1964) to generate an initial feasible solution and that solution is improved by a large neighborhood search algorithm (LNS) with simulated annealing (Pisinger and Ropke 2010).

5.1. Savings algorithm

The savings algorithm iteratively combines two schedules into one schedule based on the savings principle (Clarke and Wright 1964). The procedure begins by relaxing the maximum number of order pickers constraint and creating schedules that each consist of one batch to be picked. Then, it iteratively determines the saving in terms of the labor cost that is generated when two schedules are combined into one schedule (if possible). This saving is easy to calculate. Consider that schedule h' and h'' are combined in a feasible schedule h with the corresponding compensation $m_{h'}$, $m_{h''}$ and m_h, respectively, then the savings is $(m_{h'} + m_{h''}) - m_h$. Combining batches in two schedules into one schedule has the potential to overcome inefficiencies of individual schedules when these schedules have waiting times (or breaks) between tasks or the shift length is shorter than T_{min} time units.

To verify whether two (randomly) selected schedules can be combined into one schedule, we try to solve a simplified (or reduced) version of our OPSP, which is formulated as a MILP model in

Appendix G. Since the reduced problem finds the optimal schedule for only one order picker (or one shift) with a small number of order picking tasks, the MILP can be solved exactly in a reasonable amount of computation time. Even though the computation time of the MILP for the reduced problem is short, a set of infeasibility checks can be performed first as pre-processing step to easily verify whether the order picking tasks cannot be combined in a feasible schedule. See Appendix G for the infeasibility checks. If these checks do not rule out that a feasible schedule can be found, the reduced OPSP with one order picker is solved. If no feasible solution is found, it is concluded that the two schedules cannot be combined. Otherwise, the solution of the MILP model provides the combined schedule with the largest savings (i.e., it finds the optimal sequencing of the order picking batches).

In the classical savings algorithm by Clarke and Wright (1964), the savings of combining any given two schedules are calculated first before combining solutions in a given iteration of the algorithm. However, in this paper, if any two randomly selected schedules can be combined in a feasible schedule and result in a savings of at least T_{min} time units, the combined schedule is accepted immediately and the two individual schedules will not be considered for other savings in the same iteration of the savings algorithm. If no two schedules exist that can be combined in a feasible schedule that also results in sufficient savings of at least T_{min}, all possible combinations are first calculated and then the schedules are combined such that the maximum savings is achieved. The procedure continues until no savings can be realized while combining schedules. When no further savings can be realized and the number of schedules in the solution is less than p_{max}, a feasible solution is found that satisfies all constraints of the OPSP and the algorithm terminates. If no feasible solution is found, the savings algorithm enters the second phase, in which the batches of any pair of schedules are chosen to be combined in a new schedule that results in the largest savings (which can be the least negative savings or additional cost) until the number of schedules equals p_{max}.

5.2. Large neighborhood search for improved solutions

After a feasible solution for the OPSP is generated by the savings algorithm, this solution is improved with a large neighborhood search (LNS) procedure. Let us denote the feasible solution at the beginning of an iteration by π, where the corresponding cost (or objective function value) is $z(\pi) := \sum_{p \in P} m_p$. This solution is destroyed and then repaired in every iteration, which results in a new feasible solution π' with cost $z(\pi')$. Furthermore, let the best found solution so far be denoted by π^*. The decision whether π' becomes the starting solution in the next iteration is based on a simulated annealing principle: if $z(\pi') < z(\pi^*)$ then $\pi^* := \pi'$ and $\pi := \pi'$, otherwise π' is accepted as new solution π with probability $e^{-(z(\pi')-z(\pi^*))/\mathscr{T}}$, where $\mathscr{T}$ is the temperature that is initialized as $\mathscr{T} := -w \cdot z(\pi^*)/\ln(0.5)$ (Ropke and Pisinger 2006). The value is updated at the end of every iteration: $\mathscr{T} := \rho\mathscr{T}$, where $0 < \rho < 1$ is the cooling parameter. Consequently, it becomes less likely for worse solutions to be accepted as the starting solution in the next iteration when the number of iterations increases. If the best solution is not improved in n_T iterations, the temperature is reset to the inital value $(-w \cdot z(\pi^*)/ln(0.5))$, such that it is more likely to explore new areas in the feasible solution space.

Destruction and repair The LNS destroys and repairs the solution π in two stages. In the first stage, two order pickers are selected. The first order picker is selected probabilistically with a roulette wheel principle based on a *wastage ratio*. The wastage ratio of an order picker is the fraction of the amount of unproductive duration spent by the order picker compared to the total unproductive hours spent by all of the order pickers in the solution. The wastage ratio for order picker $p \in P$, who is assigned to complete the batches B_p with the cost m_p in solution π, is given by

$$w_p := \frac{m_p - \sum_{i \in B_p} t_i}{\sum_{p' \in P}(m_{p'} - \sum_{i \in B_{p'}} t_i)} \quad \forall p \in P. \tag{33}$$

If an order picker has a higher wastage ratio, she is likely to be chosen as the first picker. The second order picker is randomly selected among the remaining order pickers.

In the second stage, the batches previously assigned to the two selected order pickers are reassigned to generate a new (feasible) solution π'. For this purpose, we use one of two operators with equal probability. The *swap operator* exchanges a random subset of batches between the two order pickers. The *insert operator* randomly selects a subset of batches from the first order picker and assigns them to the second order picker. In the literature, swap and insert operators are typically designed to exchange or insert one job, task or trip at a time. The swap and insert operator in this work swaps and inserts multiple batches at a time. This allows us to generate new solutions that would otherwise require multiple operations with the traditional operators. The number of batches to swap or insert from each order picker is uniformly sampled between one and σ (which is a user-set parameter). If the best solution is not improved by n_σ iterations, the value of σ is reduced by 1.

After the batches are reassigned to these two order pickers, the sequencing of the batches and scheduling of shifts for the order pickers is determined by solving the same MILP of the reduced problem as in the savings algorithm (see Appendix G). Note that we also verify whether any of the infeasibility conditions is satisfied before solving the reduced problem. Rather than directly solving a MILP, other solution techniques can be proposed to solve the reduced problem. For instance, the pricing problem in Section 4.2 can be adapted to develop a dynamic programming (DP) algorithm by creating a new graph for each reduced problem, in which only the relevant batches assigned to a picker are included and a path starting at the dummy source (i.e., node o) has to visit all batch nodes in the graph before returning to the dummy sink (i.e., node d). At the dummy sink, the solution with the cheapest cost is selected and returned to the metaheuristic for evaluation. In limited numerical experiments, this DP approach to solve the reduced problem produced the same results as the MILP approach but with shorter computation times. However, the development of a DP algorithm requires labels, resource extension functions, dominance rules and acceleration techniques that need to be tailored to solve a specific reduced problem. If there would be additional restrictions in the original problem formulation (such as an upper bound on the ratio between flexible to full time order pickers, a maximum number of order pickers at any time or specific time windows for different types of breaks), these components of the DP algorithm need to be redefined. In contrast, the MILP approach is able to address different variations of the original problem without the need to change the code (see Appendix D). To accommodate flexibility in our solution approach and easily adjust to different warehouse environments, we present the LNS that uses the MILP approach to solve the reduced problem.

Figure 4 illustrates how the two destroy operators work based on a simple example. The initial schedules of the two selected order pickers are represented by X^1 and X^2. With the swap operator, batch 2 and batch 6 are interchanged. The new batches assigned to the order pickers are indicated by $B^{1'}_{Swap}$ and $B^{2'}_{Swap}$, respectively. With the insert operator, batch 2 is unassigned from the first order picker and assigned to the second order picker. The new batches assigned to the order pickers are then indicated by $B^{1'}_{Insert}$ and $B^{2'}_{Insert}$, respectively. After solving the MILP as formulated in Appendix G for each of the two order pickers individually, we obtain the new schedules $X^{1'}$ and $X^{2'}$, respectively.

The LNS terminates if $z(\pi^*)$ does not exceed the lower bound formulated in Equation (43) (see Appendix B), if the number of iterations exceeds a maximum threshold or if the run time exceeds a maximum threshold. Once the LNS terminates and time is available, we pass the LNS solution to the branch-and-price algorithm to improve the solution further by solving the pricing problem for one iteration without solving the ESPPRC exactly.

6. Results

This section presents a numerical comparison of the branch-and-price algorithm (Section 4), savings algorithm (Section 5.1) and metaheuristic (Section 5.2) to solve the OPSP. Since state-of-the-art commercial solvers such as *Gurobi 9.0.1* (Gurobi Optimization 2020) are not able to generate an optimal solution for even the smallest instances and the branch-and-price algorithm outperforms *Gurobi* without exception, we do not report the performance of such commercial solvers here. See

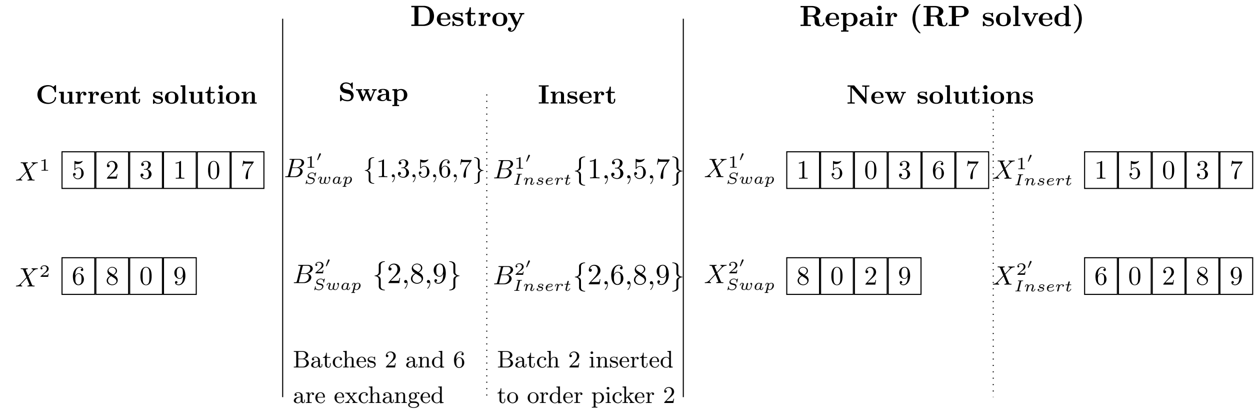


Figure 4 **Illustration of the destroy operations in an iteration of the LNS algorithm, where 0 in a schedule represents a break**

Appendix H for a comparison of the performance of commercial solver *Gurobi* and the branch-and-price algorithm. Furthermore, no existing solution procedures from the literature are included as benchmark since the authors are not aware of any other work that makes the same (or even similar) decisions and the objective function (see also Section 2).

All solution procedures are implemented in *C++* and run on an i7 3.60GHz machine with 16GB of RAM. For the parameters of the branch-and-price algorithm, the maximum number of *good* arcs from any node is set to 2 and the number of increments for the discrepancy limit in LDS (i.e., Λ) is set to 10. The parameters values for the metaheuristic are guided by the literature, where $\rho := 0.95$ and $n_T := 200$ (Bodnar et al. 2017, Stenger et al. 2013). The initial value of σ is set to 4 and n_σ to 1,250. Furthermore, $w := 0.1$ produced the best results in our numerical experiments, but we cannot guarantee optimality of this parameter value. The stopping criterion for the branch-and-price method is set to 1,800 seconds. For the metaheuristic, it is set to 360 seconds or 5,000 iterations (whichever comes first) to ensure that the method is suitable for practical applications.

6.1. Instances in numerical test bed

The instances are generated to mimic the operations of a retail grocery warehouse for which we had detailed data where the target departure times of the outbound trucks determine the staging lane operations as well as their earliest and latest due times. For all instances, we consider a 24-hour time horizon and we use minutes as our time unit. The warehouse operates 24 hours a day, and 7 days a week. However, all order picking tasks and shifts of order pickers are disjoint between different planning horizons as all shifts in a day start and end between 11pm and 11pm the next day.

Due time windows Two different patterns of due time windows are considered in our instances: *waved* and *waveless*. In waved instances, trucks arrive at the staging lanes at the same time and depart from the staging lanes at the same time (i.e., batches to be picked in the same wave have the same due time windows). Alternatively, in the waveless operations, the arrival and departure times of trucks at different staging lanes are not related. The deadline for each truck departure from a staging lane is taken from a uniform distribution in the range of [120,1425] minutes.

To make sure that there is sufficient time for the staging and loading operations of a truck, we push back deadlines (if needed) to guarantee at least 30 minutes between two consecutive departure due times of batches destined for the same dock door (or staging lane). The earliest due time of a batch is set to the latest due time of the previous batches at the same staging lane plus 15 minutes to ensure that loads of different trucks are not mixed up, and previous trucks have finished loading. The earliest due time of the first batch that is due at a staging lane is set to 0. The number of staging lanes in the instances varies from 1 to 8 (see below).

Processing time distributions The processing times of batches are taken either from one of the following uniform distributions: **U**[30,60], **U**[60,90] or **U**[90,120], or from an exponential distribution with the same corresponding average (i.e., 45, 75 or 105 minutes, respectively). The maximum processing time of a batch is restricted to 330 minutes to ensure that employees do not violate the break constraint ($T_{break} = 330$ minutes, see below).

Shift types In accordance with Dutch and European working hours laws, the maximum shift length to employ an order picker (i.e., T_{max}) is 540 minutes (or 9 hours), and the maximum time duration that an employee can work without a break (i.e., T_{break}) is 330 minutes (or 5.5 hours). The length of the break (i.e., l_b) has to be at least 45 consecutive minutes (European Parliament, Council of the European Union 2003).

We consider six shift structures. In the shift structures SStr1, SStr2 and SStr3, shifts can start every 8 hours and T_{min} equals 8, 6 and 4 hours, respectively. In the shift structures SStr4, SStr5 and SStr6, shifts can start every 4 hours and T_{min} equals 8, 6 and 4 hours, respectively. In all shift structures, a shift can end at the end of any hour after T_{min}. Consequently, shift structure SStr1 is the most restrictive and SStr6 is the most flexible. Table 3 summarizes the six shift structures we consider.

Table 3 Shift structures considered in our numerical experiments

Shift structure	Starting hours(S)	T_{min} (hours)
SStr 1	0, 8, 16	8
SStr 2	0, 8, 16	6
SStr 3	0, 8, 16	4
SStr 4	0, 4, 8, 12, 16, 20	8
SStr 5	0, 4, 8, 12, 16, 20	6
SStr 6	0, 4, 8, 12, 16, 20	4

Number of batches and staging lanes Each outbound truck requires exactly four order batches to be picked and the number of trucks departing the warehouse in the planning horizon equals either 10, 20 and 40 trucks. This results in instances with 40, 80 or 160 batches to be picked, respectively. The number of staging lanes in an instance is chosen such that the number of departures per staging lane is fixed at either 5, 10 or 20 trucks. As a result, the number of staging lanes ranges between 1 and 8 lanes. Note that instances with 10 trucks can only have 5 or 10 departures per staging lane. Furthermore, for instances were the number of trucks equals the number of truck departures in a staging lane, there is only one staging lane (grouped under *waved* in Table 4). We assume that sufficient order pickers are available to schedule with $p_{max} = 100$.

6.2. Algorithmic performance

Table 4 presents a summary of the results over all 504 instances in the test bed, whereas the results for the individual instances are presented in Appendix H. For the branch-and-price algorithm, *Root solved* indicates the number of instances for which the column generation was able to solve the linear relaxation within the run time limit of 1,800 seconds. *Optimal solution* indicates the number of instances for which the optimal solution was found within this time limit. For those instances where the branch-and-price algorithm was not able to find the optimal solution, *Optimality gap %* presents the average relative percentage cost difference between the best lower bound found after branching and the best integer solution found after branching. The average time required to solve the root node and the overall branch-and-price algorithm is indicated by CPU^{LP} and CPU^{BP}, respectively. Note that CPU^{BP} also includes the time to generate an initial solution. The average relative performance gap between the solution generated by the savings algorithm and metaheuristic compared to the best branch-and-price solution is indicated by $\%\Delta^S$ and $\%\Delta^{MH}$, respectively[1], where a positive number indicates that the branch-and-price algorithm found a better solution. The average computation time of the savings algorithm and metaheuristic is indicated by CPU^S and CPU^{MH}, respectively.

Table 4 shows that the branch-and-price algorithm is capable of solving reasonable size instances. However, the size of the instances adversely affects the performance of the exact approach. For the instances with 40, 80 and 160 batches, the root node can be solved in 100%, 58.9% and 26.3% of the instances, respectively, and the algorithm converges to an optimal solution within the run time for 60.2%, 36.1% and 19.4% of the instances, respectively. For the instances where the branch-and-price algorithm is not able to find an optimal solution, the average optimality gap is only 3.5%. Figure 5a shows the average optimality gap of the branch-and-price solutions for instances where we were able to solve the root node but the optimal solutions were not obtained.

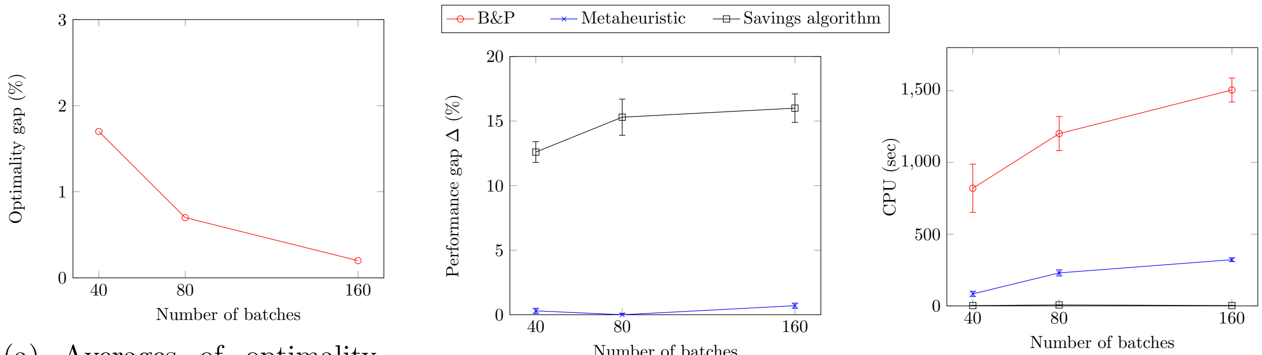
[1] $\%\Delta^S = (z(S) - z(BP))/z(BP) \times 100$ and $\%\Delta^{MH} = (z(MH) - z(BP))/z(BP) \times 100$, where $z(BP)$, $z(S)$ and $z(MH)$ denote the objective function value of the best integer solution found by the branch-and-price algorithm, savings algorithm and metaheuristic, respectively.

Table 4 Summary of results

			Branch-and-Price Algorithm					Savings Algorithm		Meta-heuristic	
			Number of instances			Average run time					
Dep. per lane	Instance type	Batches	Root solved	Optimal solution	Optimality gap %	CPU^{LP} (sec.)	CPU^{BP} (sec.)	$\%\Delta^{S}$	CPU^{S} (sec.)	$\%\Delta^{MH}$	CPU^{MH} (sec.)
5	Unif-Waved	40	18/18	11/18	4.00	10.5	809.7	11.6	2.5	0.2	48.2
		80	13/18	9/18	1.7	671.7	1,016.1	12.5	8.5	0.0	234.3
		160	3/18	1/18	0.6	1,601.5	1,758.4	12.1	33.8	0.7	361.4
	Unif-Waveless	40	18/18	10/18	5.0	54.9	839.2	17.9	2.3	0.5	110.3
		80	6/18	3/18	1.1	1,489.8	1,571.5	20.3	7.6	0.2	322.0
		160	1/18	0/18	0.8	1,956.9	1,800.0	17.4	26.2	1.9	362.2
	Exp-Waved	40	18/18	8/18	3.7	73.7	1,125.7	8.3	2.7	0.0	101.0
		80	4/18	2/18	1.5	1,496.3	1,603.9	10.5	9.0	0.0	302.8
		160	0/18	0/18	-	1,788.9	1,800.0	15.0	37.3	0.6	361.5
	Exp-Waveless	40	18/18	13/18	3.3	76.4	843.1	14.0	2.2	0.7	147.1
		80	4/18	0/18	1.4	1,637.2	1,800.0	14.7	7.9	-0.3	348.6
		160	0/18	0/18	-	1,889.1	1,800.0	12.9	28.3	1.1	363.3
10	Unif-Waved	40	18/18	16/18	2.1	1.7	204.5	12.2	1.8	0.6	30.7
		80	18/18	14/18	1.9	142.2	492.3	14.2	6.7	0.1	103.8
		160	14/18	11/18	0.4	507.2	785.1	11.3	26.1	0.2	210.0
	Unif-Waveless	80	12/18	6/18	2.5	759.8	1,278.9	18.8	6.2	-0.1	239.4
		160	6/18	5/18	0.3	1,402.5	1,447.3	17.8	21.9	0.3	336.3
	Exp-Waved	40	18/18	7/18	5.2	22.3	1,102.7	11.5	1.8	0.1	64.9
		80	12/18	8/18	3.2	758.2	1,030.2	15.0	6.5	0.4	205.0
		160	2/18	1/18	0.7	1,654.7	1,702.8	16.4	25.9	0.8	333.4
	Exp-Waveless	80	8/18	4/18	1.3	1,134.5	1,497.7	17.4	6.0	0.3	272.0
		160	1/18	0/18	3.6	1,727.7	1,800.0	20.0	21.3	1.2	364.2
20	Unif-Waved	80	17/18	14/18	1.2	108.1	410.0	13.0	4.9	0.0	66.4
		160	14/18	11/18	0.7	486.1	782.6	12.8	18.9	0.0	175.1
	Unif-Waveless	160	5/18	5/18	-	1,467.0	1,447.9	23.8	17.9	0.6	324.1
	Exp-Waved	80	12/18	5/18	1.7	671.1	1,316.7	16.5	5.0	0.0	215.0
		160	5/18	3/18	0.3	1,534.9	1,527.3	15.8	19.8	0.6	343.1
	Exp-Waveless	160	6/18	5/18	2.8	1,476.7	1,417.4	16.4	18.0	0.4	335.8

Note: Optimality gap % with "-" indicates that lower bound is not available for instances with non-optimal solutions for these set of instances.

When we compare the number of instances for which the root node (i.e., the linear relaxation) is solved and the number of instances for which an optimal solution is found within the run time limit of 1,800 seconds, we make the following observations: First, waveless instances are more difficult to solve than waved instances. A reason why the branch-and-price algorithm can solve waved instances easier is because the *limited extension* property (see Proposition 2) exploits the fact that the batches have non-overlapping due time windows when solving the pricing problem. As a result, the labeling algorithm does not have to explore as many extensions between nodes, and it is capable of solving the pricing problem more efficiently for waved instances. Second, instances with



(a) Averages of optimality gap

(b) 95% CI of performance deviation compared to B&P

(c) 95% CI of run times

Figure 5 **Performance comparision between solutions found with the branch-and-price algorithm (B&P), savings algorithm and metaheursitic**

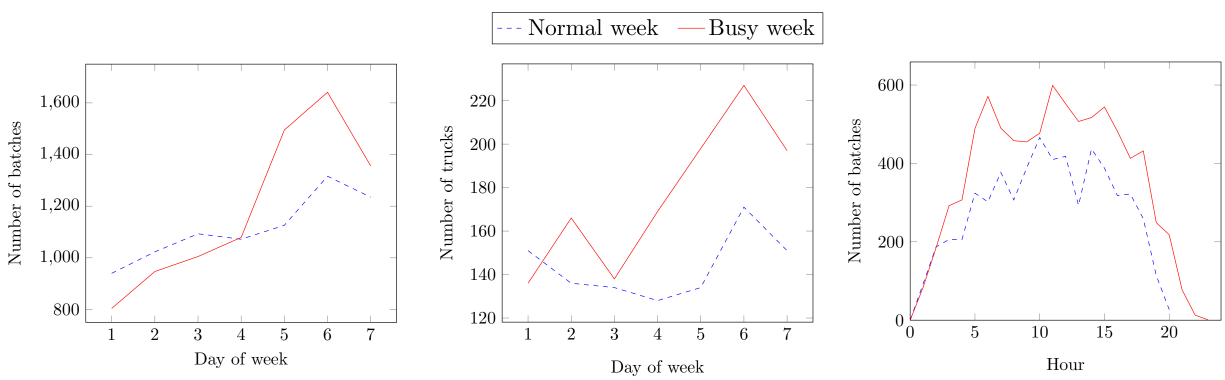
exponentially distributed processing times are more difficult to solve than instances with uniformly distributed processing times. Instances with exponentially distributed processing times have many batches with short processing times. On average, the number of tasks that can be assigned to an order picker is higher with the exponentially distributed processing times. As a result, the labeling algorithm has to consider more potential solutions and labels when solving the pricing problem. Third, instances with more truck departures per staging lane are easier to solve than instances with fewer truck departures. When there are more trucks departing from the same staging lane, the average length of the due time windows is smaller (see Figure 11 in Appendix H). As a result, the pricing problem needs to consider fewer extensions from any node as many potential solutions are not feasible. See Appendix H for a more detailed discussion on these observations.

The savings algorithm is able to quickly generate a feasible solution (on average within 4.3 seconds) for either the branch-and-price algorithm or the metaheuristic. However, the quality of these solutions is poor, with an average cost deviation of 15.0% compared to the best solutions found with the branch-and-price algorithm. In contrast, the solutions with the metaheuristic have an average performance gap of less than 0.4% which is found within less than one-fifth of the computational time required for the branch-and-price algorithm. Figure 5b and 5c present the performance gap of the heuristic procedures and the computational time for each of the three solution approaches, respectively.

6.3. Flexible shift structures: A case study

In this subsection, we apply the metaheuristic to the order picker scheduling problem at a warehouse with perishable products of a Dutch grocery retailer. The case study serves two purposes. First, it evaluates the usability of the metaheuristic that we propose to solve industrial instances. Second, the case illustrates some of the ways in which the methodology in this paper can be used to evaluate warehouse operating policies of interest to managers. In particular, we study the impact of the shift structures on the number of order pickers scheduled to perform the order-picking activities.

Description instances The retailer provided operational data regarding the processing times and due time windows of batches for two weeks of their operations. The first week represents a typical week in terms of the number of batches to be picked and shipped from the warehouse. The second week represents the busiest week of the year, which occurs during the Christmas season. There are 6.9% more batches to be picked in the busier week compared to the typical week (see Figure 6a, where day 1 is a Sunday). The warehouse has 53 staging lanes, and the number of trucks departing from the warehouse ranges between 128 and 227 trucks per day (see Figure 6b). When we consider the number of batches with a due deadline in a particular hour in Figure 6c, we identify two peak periods of operations: between hour 5 and hour 7, and between hour 10 and hour 12. In this figure, hour 0 corresponds to 11:00 pm since the warehouse starts its order-picking activities at that hour. The average processing time of a batch is around 41 minutes for both the busy and normal week, and the distribution of these processing times are similar in both weeks (see Figure 7a). The distribution of the duration of the due time windows is illustrated in Figure 7b. The larger due time windows in the right tail in this figure occur on days with fewer trucks departing from the warehouse (i.e., on day 1).



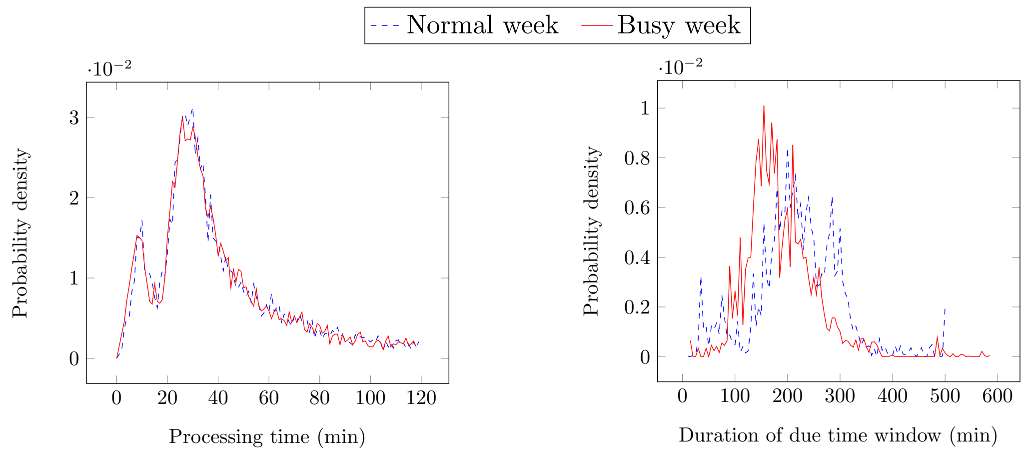
(a) Number of batches per day, where day 1 corresponds to Sunday

(b) Number of truck departures per day, where day 1 corresponds to Sunday

(c) Number of batches due per hour, where hour 0 corresponds to 11:00 pm

Figure 6 **The workload in the case study**

Current shift structure The employees are hired to work at the warehouse through third party agencies. Their shifts can start at hour 0, 8 and 9 (i.e., at 11:00 pm, 7:00 am and 8:00 am). The flexible workers are allowed to work for at most 9 hours (i.e., $T_{max} = 9$ hours) and are compensated for at least 6 hours (i.e., $T_{min} = 6$ hours). In contrast to our MILP formulation in Section 3, the order pickers receive three breaks at fixed times after they start their shift: a 15-minute break after



(a) Distribution of processing times for picking batches

(b) Distribution of the due time window durations to pick and deliver batches

Figure 7 Variability in the processing times and durations of due time windows for picking batches

2 hours, a 30-minute break after 3.5 hours and another 15-minute break after 6 hours. This shift structure is compliant with the EU and Dutch labor laws.

The warehouse manager has to determine the number of order pickers to schedule for each of the three shift start times, the shift duration of each order picker as well as the batches to be picked by each order picker. Currently, these decisions are made based on experience and intuition of warehouse managers. Due to data privacy concerns, the retailer was not willing to share the actual order picker schedules.

There are four interesting research questions in our case study with the retailer: (i) Can the metaheuristic that is developed in Section 5 be used in practice as a decision support tool? (ii) What is the value of flexible break times rather than fixed break times (that are currently used by the retailer)? (iii) What is the value of an additional shift start time? (iv) Can the retailer leverage flexible break times and an additional shift start time to offer a larger minimum compensation T_{min} without incurring higher labor costs? Especially the last question is of particular interest to the retailer since they believe that a larger minimum compensation helps to foster better working relationships with order pickers and to improve the retention rates of employees.

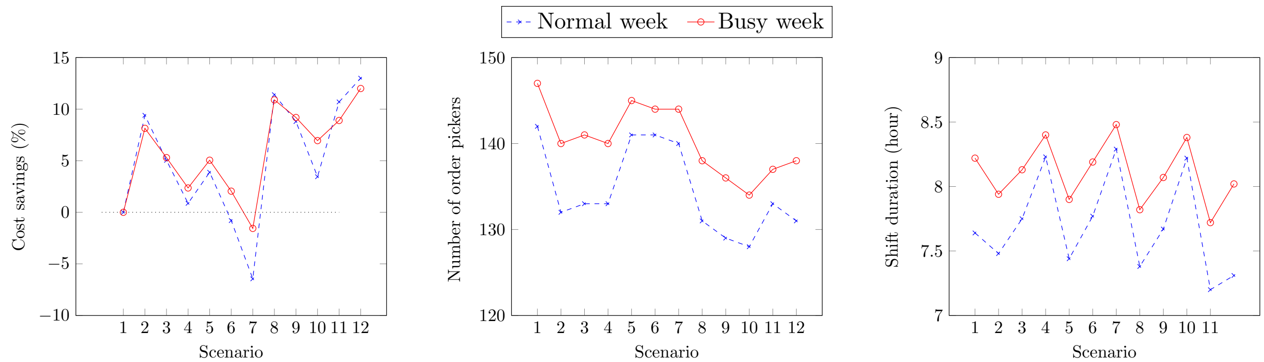
To answer these questions, we first conducted multiple rounds of consultation with the planners and managers to develop plausible and actionable scenarios. The scenarios can be distinguished along three dimensions. First is *flexible break times*. This means that the breaks for order pickers are scheduled at the current break start times ± 15 minutes. Second, an *additional shift start time* is introduced at hour 4 to account for the workload peak as illustrated in Figure 6c. We have also tried an additional start time at hour 3 and hour 5, but an additional shift start time at hour 4 resulted in the lowest objective function values. Third, the *minimum compensation time* can be increased to 7 hours or even 8 hours instead of 6 hours. Additionally, we introduce two shift

Table 5 Shift structure scenarios to analyse

Shift Structure	Description	Flexible Break Times	Additional Shift Start	T_{min} (hours)
Scenario 1	Current scenario (base case)			6 hours
Scenario 2	Flexible breaks and $T_{min} = 6$ hours	✓		6
Scenario 3	Flexible breaks and $T_{min} = 7$ hours	✓		7
Scenario 4	Flexible breaks and $T_{min} = 8$ hours	✓		8
Scenario 5	Extra shift and $T_{min} = 6$ hours		✓	6
Scenario 6	Extra shift and $T_{min} = 7$ hours		✓	7
Scenario 7	Extra shift and $T_{min} = 8$ hours		✓	8
Scenario 8	Flexible breaks, extra shift and $T_{min} = 6$ hours	✓	✓	6
Scenario 9	Flexible breaks, extra shift and $T_{min} = 7$ hours	✓	✓	7
Scenario 10	Flexible breaks, extra shift and $T_{min} = 8$ hours	✓	✓	8
Scenario 11	Theoretical breaks and $T_{min} = 6$ hours	$T_{break} = 2$ hours, $l_b = 20$ minutes		6
Scenario 12	Theoretical breaks, extra shift and $T_{min} = 6$ hours	$T_{break} = 2$ hours, $l_b = 20$ minutes	✓	6

structures that are in line with Section 3: a break of 20 minutes needs to be scheduled after at most 2 hours of work (i.e., $T_{break} = 2$ hours and $l_b = 20$ minutes). The minimum compensation time (i.e., T_{min}) is still 6 hours. This shift structure is comparable to the current shift structure in the sense that an employee is compensated for either two or three breaks in any shift, and the values of T_{min} and T_{max} are the same. An overview of the 12 different shift structure scenarios is provided in Table 5. Scenario 1 corresponds to the current shift structure, which serves as benchmark. Since the shift structure at the retailer is different than discussed in Section 3, we adapted the reduced problem of the metaheuristic to consider flexible break times (see Appendix I for details).

The overall cost savings as well as the impact on the average number of scheduled order pickers and on the average shift length are presented in Figure 8a, 8b and 8c, respectively, whereas the detailed results are presented in Appendix J. By allowing 15 minutes of flexibility in the break times, the labor cost savings for the retailer are on average 8.8% (comparing scenario 2 to the base case of scenario 1). In particular, fewer employees have to be scheduled and the average shift length decreases as well. Example schedules under scenario 1, 2, 5 and 11 are presented in Appendix J. When the minimum compensation time is increased from 6 hours to 7 or 8 hours (i.e., scenario 3 and 4), the retailer can still expect to have an average cost saving of 5.2% and 0.7%, respectively, by adopting flexible break times. The number of employees to schedule remains similar in the scenarios 2, 3 and 4, however, the average shift length increases. Interestingly, the average shift length in scenario 3 is comparable to scenario 1, i.e., the cost savings of 5.2% in scenario 3 are mainly due to the scheduling of fewer order pickers. Increasing the minimum compensation time to 8 hours (in scenario 4) still results in cost savings. This is good news for the retailer, since the additional cost of an increased value of T_{min} is offset against 15 minutes of flexibility in the break start times. Allowing even more flexibility in scheduling breaks (in scenario 11), the average labor cost can decrease an additional 1% (the cost savings in scenario 11 is 9.8%, whereas this is 8.8% in scenario 2). However, this is considered not favorable by the retailer since there is less overlap between the breaks of employees in scenario 11 (see also Appendix J), which is of social importance for the employees. Since the majority of the cost savings in scenario 11 are also captured by allowing 15 minutes of flexibility in the break times (as in scenario 2), these results provided sufficient motivation to initiate implementing this 15 minutes of flexibility.



(a) Average cost savings relative to scenario 1

(b) Average number of order pickers scheduled

(c) Average shift length for an order picker

Figure 8 **Performance of the different shift structure scenarios in the case study**

The average cost savings of an additional shift start time at hour 4 is less substantial compared to flexible break times: 4.5% when T_{min} equals 6 hours, only 0.6% when T_{min} equals 7 hours and an average cost increase of 4% when T_{min} equals 8 hours (for scenario 5, 6 and 7, respectively). This is mainly because the number of order pickers that need to be scheduled decreases significantly less compared to flexible break times, whereas the average shift lengths are comparable.

When combining flexible break times and adding a shift start time at hour 4, the average labor cost can (obviously) decrease even further. What is interesting to observe is that the number of order pickers that is scheduled is actually decreasing as the minimum compensation time T_{min} increases from 6 to 7 hours and from 7 to 8 hours (comparing scenario 8, 9 and 10). Since the increase in average shift length is similar as before, the marginal decrease in average cost savings is less when T_{min} increases. The corresponding average cost savings in these scenarios are 11.1%, 9.0% and 5.2%, respectively. Finally, we observe that most of the cost savings of the flexible break times are captured by the 15-minute flexibility of the break times, since the cost savings in scenario 8 and 12 correspond to 11.1% and 12.5%, respectively (similar when comparing the cost savings between scenario 2 and 11). This reinforces our previous conclusion that it is sufficient to include only 15 minutes of flexibility when scheduling the break times.

7. Conclusion

In this paper, we study the order picker scheduling problem where order-picking tasks can be done flexibly but are constrained with due time windows. The problem intersects with the personnel scheduling literature. However, unlike the available literature, our problem minimizes the labor cost while considering the minimum promised pay to order pickers, the shift start and end times of employees as well as break times. Therefore, break times are explicitly included as scheduling variables as well as shift start and end times (with a minimum compensation time for each order picker). This is a common problem at warehouses with manual order pickers where batches of items need to be picked and delivered to outbound dock doors (or staging lanes) within the time windows that the trucks are scheduled to load the items. Since it combines the order picker planning problem and the shift scheduling problem, we call this the order picker scheduling problem. We present several formulations of the problem with a range of operational restrictions that are important to consider. Two methods are presented to solve the problem. First, an exact branch-and-price algorithm is developed. Since this algorithm can be prohibitive for practical applications, we also present an efficient metaheuristic that combines a savings algorithm and large neighborhood search. The results indicate that the heuristic has a stable performance and is capable of producing near-optimal solutions in a reasonable time for real-life instances.

In a case study, we show how the problem and solution approaches can be used to study different shift structures. In particular, the results show that the retailer can readily increase the minimum compensation duration for workers from 6 to 7 or 8 hours and still realize average labor cost savings of 5.2% or 0.7%, respectively, when a 15 minute flexibility in the scheduling of break times is implemented. By increasing the minimum compensation duration, order pickers might experience an improved job satisfaction to promote job retention. More cost savings of around 4-4.5% can be achieved when an additional shift start time is introduced. Inspired by the result, the retailer under study has decided to implement additional shifts and flexible breaks. Moreover, the findings are applicable beyond this grocery retailer as most retailers in Western Europe operate their warehouses constrained by staging time windows with flexible order pickers in a similar manner.

For the sake of brevity, we only consider identical order pickers in our study. Since the evaluation of a schedule in both our solution approaches to the problem is on the individual employee, order picker specific characteristics such as age and seniority-based breaks as well as restricted and preferred shift starting times can be added to the pricing problem for the branch-and-price algorithm and the reduced problem for the metaheuristic.

Furthermore, we assumed that shifts and order picking tasks are non-overlapping between different planning horizons (i.e., the shift start and end times of every shift are in the same planning horizon when batch orders can be picked). If this assumption were to be relaxed, we propose to use our solution methodology with a rolling horizon. In particular, we suggest to extend the planning horizon with an additional time period during which no new shifts are allowed to start but employees who started their shift in the original planning horizon can finish their shift in the extended time period and perform order picking tasks during that time period. Consequently, order pickers can be scheduled more efficiently at the end of the planning horizon and order picking tasks in the next planning horizon can be performed already. Such a rolling horizon approach results in feasible solutions that may not be optimal as there may not be sufficient order picking tasks available for the order pickers that start their shift in the next planning horizon. A shift scheduling problem that can dynamically include arrivals of new orders in an online environment could be of significant value for e-commerce companies.

Future research can take two additional trajectories within the offline retail environment. First, given the size of the instances in real-life business applications, order batching decisions are made *a priori* (similar to our approach). It can be worthwhile to jointly consider the order batching and shift scheduling problem. Second, we assume norm times to perform the order-picking activities in a deterministic manner. A compelling research direction would be to consider robust order picker scheduling problems with stochastic processing times of batches. These two research directions can be of significant value to both academia and practice.

References

Alsheddy, A. and Tsang, E. P. Empowerment scheduling for a field workforce. *Journal of Scheduling*, 14(6): 639–654, 2011.

Aykin, T. Optimal shift scheduling with multiple break windows. *Management Science*, 42(4):591–602, 1996.

Aykin, T. A comparative evaluation of modeling approaches to the labor shift scheduling problem. *European Journal of Operational Research*, 125(2):381–397, 2000.

Azadeh, K., De Koster, R., and Roy, D. Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4):917–945, 2019.

Bard, J. F., Morton, D. P., and Wang, Y. M. Workforce planning at usps mail processing and distribution centers using stochastic optimization. *Annals of Operations Research*, 155(1):51, 2007.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

Bechtold, S. E. and Jacobs, L. W. Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science*, 36(11):1339–1351, 1990.

Beliën, J. and Demeulemeester, E. A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal of Operational Research*, 189(3):652–668, 2008.

Bertsimas, D., Chang, A., Mišić, V. V., and Mundru, N. The airlift planning problem. *Transportation Science*, 53(3):773–795, 2019.

Bhandari, A., Scheller-Wolf, A., and Harchol-Balter, M. An exact and efficient algorithm for the constrained dynamic operator staffing problem for call centers. *Management Science*, 54(2):339–353, 2008.

Bodnar, P., de Koster, R., and Azadeh, K. Scheduling trucks in a cross-dock with mixed service mode dock doors. *Transportation Science*, 51(1):112–131, 2017.

Boysen, N., Briskorn, D., and Emde, S. Sequencing of picking orders in mobile rack warehouses. *European Journal of Operational Research*, 259(1):293–307, 2017.

Brusco, M. J. and Jacobs, L. W. Personnel tour scheduling when starting-time restrictions are present. *Management Science*, 44(4):534–547, 1998.

Brusco, M. J. and Jacobs, L. W. Optimal models for meal-break and start-time flexibility in continuous tour scheduling. *Management Science*, 46(12):1630–1641, 2000.

Chen, T.-L., Cheng, C.-Y., Chen, Y.-Y., and Chan, L.-K. An efficient hybrid algorithm for integrated order batching, sequencing and routing problem. *International Journal of Production Economics*, 159: 158–167, 2015.

Clarke, G. and Wright, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

Contardo, C., Desaulniers, G., and Lessard, F. Reaching the elementary lower bound in the vehicle routing problem with time windows. *Networks*, 65(1):88–99, 2015.

Cordeau, J.-F., Laporte, G., Pasin, F., and Ropke, S. Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13(4):393–409, 2010.

Côté, M.-C., Gendron, B., and Rousseau, L.-M. Grammar-based integer programming models for multiactivity shift scheduling. *Management Science*, 57(1):151–163, 2011.

Dahmen, S., Rekik, R., and Soumis, F. An implicit model for multi-activity shift scheduling problems. *Journal of Scheduling*, 21(3):285–304, 2018.

Danna, E. and Le Pape, C. Branch-and-price heuristics: A case study on the vehicle routing problem with time windows. In *Column generation*, pages 99–129. Springer, 2005.

Dantzig, G. B. Letter to the editor—a comment on Edie's "Traffic Delays at Toll Booths". *Journal of the Operations Research Society of America*, 2(3):339–341, 1954.

Dellaert, N., Dashty Saridarq, F., Van Woensel, T., and Crainic, T. G. Branch-and-price–based algorithms for the two-echelon vehicle routing problem with time windows. *Transportation Science*, 53(2):463–479, 2018.

Desaulniers, G. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research*, 58(1):179–192, 2010.

Desaulniers, G. and Villeneuve, D. The shortest path problem with time windows and linear waiting costs. *Transportation Science*, 34(3):312–319, 2000.

Desaulniers, G., Desrosiers, J., and Solomon, M. M. Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems. In *Essays and surveys in metaheuristics*, pages 309–324. Springer, 2002.

Desaulniers, G., Lessard, F., and Hadjar, A. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404, 2008.

Dror, M. Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5):977–978, 1994.

Edie, L. C. Traffic delays at toll booths. *Journal of the Operations Research Society of America*, 2(2): 107–138, 1954.

Elahipanah, M., Desaulniers, G., and Lacasse-Guay, E. A two-phase mathematical-programming heuristic for flexible assignment of activities and tasks to work shifts. *Journal of Scheduling*, 16(5):443–460, 2013.

Elsayed, E. and Lee, M.-K. Order processing in automated storage/retrieval systems with due dates. *IIE Transactions*, 28(7):567–577, 1996.

Elsayed, E., Lee, M.-K., Kim, S., and Scherer, E. Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals. *The International Journal of Production Research*, 31(3):727–738, 1993.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., and Sier, D. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127(1–4):21–144, 2004a.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, 2004b.

European Parliament, Council of the European Union. Council Directive 91/533/EEC of 14 October 1991 on an employer's obligation to inform employees of the conditions applicable to the contract or employment relationship, 1991.
`https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A31991L0533`.

European Parliament, Council of the European Union. Directive 2003/88/EC of the European Parliament and of the Council of 4 November 2003 concerning certain aspects of the organisation of working time, 2003.
`http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32003L0088`.

European Parliament, Council of the European Union. Directive (EU) 2019/1152 of the European Parliament and of the Council of 20 June 2019 on transparent and predictable working conditions in the European Union, 2019.
`https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32019L1152`.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3): 216–229, 2004.

Feillet, D., Gendreau, M., and Rousseau, L.-M. New refinements for the solution of vehicle routing problems with branch and price. *INFOR: Information Systems and Operational Research*, 45(4):239–256, 2007.

Fırat, M. and Hurkens, C. An improved mip-based approach for a multi-skill workforce scheduling problem. *Journal of Scheduling*, 15(3):363–380, 2012.

Fischetti, M., Martello, S., and Toth, P. The fixed job schedule problem with spread-time constraints. *Operations Research*, 35(6):849–858, 1987.

Fischetti, M., Martello, S., and Toth, P. The fixed job schedule problem with working-time constraints. *Operations Research*, 37(3):395–403, 1989.

Gérard, M., Clautiaux, F., and Sadykov, R. Column generation based approaches for a tour scheduling problem with a multi-skill heterogeneous workforce. *European Journal of Operational Research*, 252 (3):1019–1030, 2016.

Goel, A. Truck driver scheduling in the European Union. *Transportation Science*, 44(4):429–441, 2010.

Goel, A. and Irnich, S. An exact method for vehicle routing and truck driver scheduling problems. *Transportation Science*, 51(2):737–754, 2017.

Goel, A. and Kok, L. Truck driver scheduling in the United States. *Transportation Science*, 46(3):317–326, 2012.

Gurobi Optimization, L. Gurobi optimizer reference manual, version 9.0, 2020. URL `http://https://www.gurobi.com//wp-content//plugins//hd_documentations//documentation//9.0//refman.pdf`.

Henn, S. Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses. *Flexible Services and Manufacturing Journal*, 27(1):86–114, 2015.

Henn, S. and Schmid, V. Metaheuristics for order batching and sequencing in manual order picking systems. *Computers & Industrial Engineering*, 66(2):338–351, 2013.

Holder, A. Navy personnel planning and the optimal partition. *Operations Research*, 53(1):77–89, 2005.

Irnich, S. Resource extension functions: Properties, inversion, and generalization to segments. *OR Spectrum*, 30(1):113–148, 2008.

Irnich, S. and Desaulniers, G. Shortest path problems with resource constraints. In *Column generation*, pages 33–65. Springer, 2005.

Jans, R. Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. *INFORMS Journal on Computing*, 21(1):123–136, 2009.

Kallehauge, B., Larsen, J., Madsen, O. B., and Solomon, M. M. Vehicle routing problem with time windows. In *Column generation*, pages 67–98. Springer, 2005.

Kniffin, K. M., Wansink, B., Devine, C. M., and Sobal, J. Eating together at the firehouse: how workplace commensality relates to the performance of firefighters. *Human Performance*, 28(4):281–306, 2015.

Kolen, A. W., Lenstra, J. K., Papadimitriou, C. H., and Spieksma, F. C. Interval scheduling: A survey. *Naval Research Logistics*, 54(5):530–543, 2007.

Krishnamoorthy, M., Ernst, A. T., and Baatar, D. Algorithms for large scale shift minimisation personnel task scheduling problems. *European Journal of Operational Research*, 219(1):34–48, 2012.

Kroon, L. G., Salomon, M., and Van Wassenhove, L. N. Exact and approximation algorithms for the operational fixed interval scheduling problem. *European Journal of Operational Research*, 82(1):190–205, 1995.

Matusiak, M., de Koster, R., Kroon, L., and Saarinen, J. A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. *European Journal of Operational Research*, 236(3):968–977, 2014.

Matusiak, M., de Koster, R., and Saarinen, J. Utilizing individual picker skills to improve order batching in a warehouse. *European Journal of Operational Research*, 263(3):888–899, 2017.

Michel, R. 2016 Warehouse/DC operations survey: ready to confront complexity. *Supply Chain Management Review, November, 2016*, 2016.
`https://www.scmr.com/plus/SCMR1611{\_}SUP{\_}Warehouse{\_}DCBenchmarkStudy.pdf`, Accessed: 05-03-2020.

Ministry of Social Affairs and Employment. The working hours act, 2010.
`https://www.rijksoverheid.nl/binaries/rijksoverheid/documenten/brochures/2010/05/10/de-arbeidstijdenwet-engels/brochure-de-arbeidstijdenwet-engels.pdf`.

Pisinger, D. and Ropke, S. Large neighborhood search. In *Handbook of metaheuristics*, pages 399–419. Springer, 2010.

Quak, H. J. and de Koster, R. Exploring retailers' sensitivity to local sustainability policies. *Journal of Operations Management*, 25(6):1103–1122, 2007.

Rasmussen, M. S., Justesen, T., Dohn, A., and Larsen, J. The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3): 598–610, 2012.

Ropke, S. and Pisinger, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006.

Scholz, A., Schubert, D., and Wäscher, G. Order picking with multiple pickers and due dates – simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems. *European Journal of Operational Research*, 263(2):461–478, 2017.

Sherali, H. D. and Smith, J. C. Improving discrete model representations via symmetry considerations. *Management Science*, 47(10):1396–1407, 2001.

Smet, P., Ernst, A. T., and Berghe, G. V. Heuristic decomposition approaches for an integrated task scheduling and personnel rostering problem. *Computers & Operations Research*, 76:60–72, 2016.

Spliet, R., Dabia, S., and Van Woensel, T. The time window assignment vehicle routing problem with time-dependent travel times. *Transportation Science*, 52(2):261–276, 2018.

Stenger, A., Vigo, D., Enz, S., and Schwind, M. An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science*, 47(1):64–80, 2013.

Sungur, B., Özgüven, C., and Kariper, Y. Shift scheduling with break windows, ideal break periods, and ideal waiting times. *Flexible Services and Manufacturing Journal*, 29(2):203–222, 2017.

Thompson, G. M. *A comparison of techniques for scheduling non-homogeneous employees in a service environment subject to non-cyclical demand volume I, chapters l-7.* PhD thesis, 1988.

Thompson, G. M. Improved implicit optimal modeling of the labor shift scheduling problem. *Management Science*, 41(4):595–607, 1995.

Thompson, G. M. and Pullman, M. E. Scheduling workforce relief breaks in advance versus in real-time. *European Journal of Operational Research*, 181(1):139–155, 2007.

Tilk, C. and Goel, A. Bidirectional labeling for solving vehicle routing and truck driver scheduling problems. *European Journal of Operational Research*, 283(1):108–124, 2020.

Tilk, C. and Irnich, S. Dynamic programming for the minimum tour duration problem. *Transportation Science*, 51(2):549–565, 2017.

Tompkins, J. A., White, J. A., Bozer, Y. A., and Tanchoco, J. *Facilities Planning.* John Wiley & Sons, Hoboken, NJ, 2010.

Tsai, C.-Y., Liou, J. J., and Huang, T.-M. Using a multiple-ga method to solve the batch picking problem: Considering travel distance and order due time. *International Journal of Production Research*, 46(22): 6533–6555, 2008.

Van den Bergh, J., Beliën, J., de Bruecker, P., Demeulemeester, E., and De Boeck, L. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

Van Gils, T., Ramaekers, K., Braekers, K., Depaire, B., and Caris, A. Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions. *International Journal of Production Economics*, 197:243–261, 2018a.

Van Gils, T., Ramaekers, K., Caris, A., and de Koster, R. B. Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research*, 267(1):1–15, 2018b.

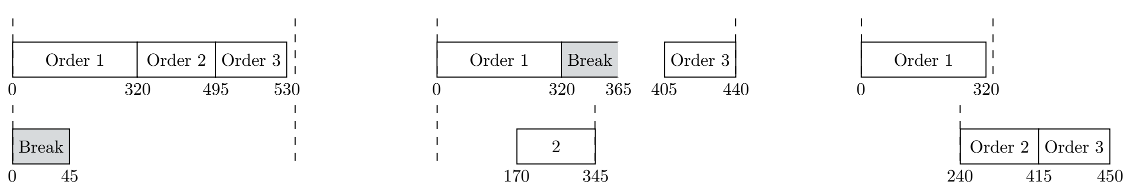
Vanderbeck, F. On dantzig-wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48(1):111–128, 2000.

Appendix A: Illustrative example

Consider the following example that illustrates the importance of shift start, end and break timing decisions along with assignment and sequencing decisions when orders have due time window constraints.

Example 1. *Consider a warehouse that employs flexible order pickers who can work for at most 540 minutes and are guaranteed a minimum payment equal to 330 minutes of work (independent of the amount of work actually performed). There are 2 possible shift start times: time 0 and time 240 (in minutes). Labor laws require that employees are given a 45-minute uninterrupted break after 330 minutes of work. There are three batch of orders to be picked: the first batch has a processing time of 320 minutes and the due time window is [0,320] in minutes. The second batch has a processing time of 175 minutes and the due time window is [345,415] in minutes. The third batch has a processing time of 35 minutes and the due time window is [440,540] in minutes.*

The available approach in the order picking literature suggests to implicitly include breaks as "work" that needs to be scheduled with fixed shift start and end times for all scheduled order pickers. This generic approach can result in a schedule with two order pickers as shown in Figure 9a. The first order picker picks all three batches consecutively and then the break is completed by a second order picker since the shift of the first order picker would violate the maximum shift length otherwise. The resulting shifts satisfy the shift duration limit as well as the due time windows to pick the batches. However, the first order picker has no scheduled break. Alternatively, when we assign breaks that comply to the labor laws but we only consider one shift start time, the order picking plan results in the schedule indicated in Figure 9b. In this schedule, the second order picker has to wait 170 minutes until she can start to pick items towards batch 2 because of the fixed shift start times. This schedule corresponds to 785 minutes of compensation for the scheduled order pickers. Finally, when we consider multiple shift start times, the shift of the second order picker can start at time 240, and she can immediately start to pick items for batch 2. See Figure 9c. This optimal schedule corresponds to 660 minutes of compensation for the scheduled order pickers. The distinction between these three examples clearly indicate the need to explicitly consider shift scheduling decisions in order picker planning problems.



(a) Schedule with implicit breaks and fixed shift start and end times

(b) Schedule with explicit breaks and flexible shift end times

(c) Schedule with explicit breaks and flexible start and end times

Figure 9 Illustration of the importance of scheduling breaks and shifts in manual order picking operations with temporal constraints. The duration between the dotted lines represents the time interval for which the order picker is compensated.

Appendix B: Additional constraints to OPSP formulation

The formulation of our OPSP in Section 3 suffers from symmetry. This means that interchanging an entire schedule of batches and breaks between different order pickers leaves the objective function unchanged. This issue can be partially addressed with the use of lexicographic ordering constraints (Sherali and Smith 2001, Jans 2009). Constraints (34) impose a hierarchy between order pickers who are scheduled to work and those who are not. In particular, order pickers are scheduled sequentially based on the start time of their shift (i.e., the start time of order picker 1 cannot exceed the start time of order picker 2, etc.).

$$\sum_{j \in S} j s_{jp-1} \leq \sum_{j \in S} j s_{jp} \qquad \forall p \in P \setminus \{1\} \tag{34}$$

The second set of additional constraints provides a lower bound on the number of order pickers that should be scheduled to work between time periods j and j', where $j' > j$. Constraints (35) specify that the number of time periods during which an order picker needs to be scheduled to perform an order picking task between the two period j and j' needs to at least the minimum workload that needs to be executed during those time periods to ensure that all batches are picked within their due time windows.

$$\sum_{p \in P} \left[\sum_{\bar{j}=1}^{j'} \left(j' - \left(\max\{\bar{j}, j\} - 1\right)\right) s_{\bar{j}p} - \sum_{\bar{j}=1}^{j'} \left(j' - \max\{\bar{j}, j-1\}\right) e_{\bar{j}p} \right]$$
$$\geq \frac{\sum_{i \in I} \left(t_i - \max\left\{ (d_i - j' \cdot l)^+ + ((j-1) \cdot l - (d_i - t_i))^+ , \mathbb{1}_{\{r_i \leq j' \cdot l\}} \cdot \left((j-1) \cdot l - (r_i - t_i)^+ \right)^+ \right\} \right)^+}{l} \qquad \forall j < j', j \in J, j' \in J \tag{35}$$

where $(A)^+ = \max\{A, 0\}$ and $\mathbb{1}_{\{condition\}}$ is an indicator function which has the value 1 if the condition is true and 0 otherwise. The left-hand side of this constraint is the summation of all periods that order pickers are scheduled to work during the time interval from period j until period j'. The first summation equals all time periods from period $\max\{\bar{j}, j\}$ until period j', where $\bar{j} \leq j'$ is the starting period of the shift for order picker p. However, the time periods after the shift ends until period j' need to be subtracted. Note that both summations are zero if the start time of a shift exceeds period j', both summations are equal if the start and end time of a shift preceeds period j, and the second summation is zero if the end time of a shift exceeds period j'. The term in between the brackets of the numerator of the right-hand side represents the minimum amount of execution time in the time interval from period j until period j' to pick the items of batch i. Note that period j starts at time $(j-1) \cdot l$ and period j' ends at time $j' \cdot l$. Consequently, the first term of the maximization is the maximum amount of time that batch i can be executed after period j' while still respecting the latest due time d_i and the second term is the maximum amount of time that batch i can be executed before period j while still respecting the earliest due time r_i. Subtracting this from the execution time t_i leaves the minimum amount of time units associated with the order picking task of batch i during the time periods j and j'. Consequently, the numerator in constraints (35) is equal to the minimum amount of time units that order pickers need to be scheduled during the time interval from period j until period j' to execute the batches $i \in I$. Since the denominator is the length of one time period, the ratio represents the minimum amount of time periods for which order pickers need to be scheduled to work between time period j and $j' > j$.

The final additional constraint is a lower bound on the objective function value in our OPSP formulation. This requires us to determine the minimum number of time units that order pickers need to be scheduled to perform $\sum_{i \in I} t_i$ time units of order picking tasks. For simplicity we assume that all order pickers have the same shift length, which is indicated by l_s time units. The scheduled time in each shift is assigned for order picking tasks and for breaks. To better understand the dynamics of order picking time and breaks during a shift, consider the following example: $l_s = 7.5$, $T_{break} = 3$ and $l_b = 1$. This means that the employee is scheduled for 6 time units to pick orders and 1.5 time units to take a break. However, when l_s changes to 4.5 time units, then the employee is scheduled for 3.5 time units to pick orders and 1 time unit to take a break. The following expression can be used to determine the number of time units that an employee is picking orders as a function of the shift length:

$$T_{work}(l_s) = \begin{cases} l_s - \left\lfloor \frac{l_s}{T_{break}+l_b} \right\rfloor \cdot l_b, & \text{if } l_s - \left\lfloor \frac{l_s}{T_{break}+l_b} \right\rfloor \cdot (T_{break}+l_b) \leq T_{break} \\ T_{break} \cdot \left(\left\lfloor \frac{l_s}{T_{break}+l_b} \right\rfloor + 1 \right), & \text{otherwise} \end{cases} \tag{36}$$

In the first condition, the shift stops while performing an order picking task and it stops while taking a break in the second condition. The minimum number of employees to cover $\sum_{i \in I} t_i$ time units of order picking tasks equals

$$N(l_s) = \left\lceil \frac{\sum_{i \in I} t_i}{T_{work}(l_s)} \right\rceil \tag{37}$$

The first $N(l_s) - 1$ order pickers are scheduled to perform order picking tasks for $T_{work}(l_s)$ time units and the additional worker only needs to be scheduled to cover the remaining work, which equals

$$\Delta(l_s) = \sum_{i \in I} t_i - (N(l_s) - 1) \cdot T_{work}(l_s). \tag{38}$$

If $\Delta(l_s) < T_{min}$, it can happen that it is not justifiable to assign this work to an N-th order picker since she has to be compensated for T_{min} time units. Instead it would be better to assign this work to the previous $N - 1$ order pickers by extending their shift length. The computations for this analysis depend on the actual shift length l_s, which we discuss first.

The question is what the best duration of the shift (i.e., l_s) needs to be such that the compensation for the scheduled order pickers is minimized. Note that it is most efficient to have a shift without a break. In that case, the productivity of the employee is 100% since a break is non-productive time that is scheduled. However, T_{min} can prevent a shift length of T_{break} time periods. Therefore, it is best to start with T_{break} time units of work and then add the minimum number of blocks of $l_b + T_{break}$ time units such that the total shift length equals or exceeds T_{min} time units for the first time. Consequently, the shift length cannot be extended without scheduling a break first. This corresponds to the following shift length:

$$l_s^{(1)} = \min \left\{ T_{break} + \left\lceil \frac{(T_{min} - T_{break})^+}{l_b + T_{break}} \right\rceil (l_b + T_{break}), T_{max} \right\} \tag{39}$$

When $N(l_s^{(1)}) - 1$ order pickers are scheduled with a shift length of $l_s^{(1)}$ time units, we can analyze how to add the additional $\Delta(l_s^{(1)})$ time units to the schedule (as introduced above). If they are assigned to an N-th order picker, we need to make sure that this order picker is scheduled at least T_{min} time units. Alternatively, the additional $\Delta(l_s^{(1)})$ time units can be added to the shifts of the other $N-1$ order pickers. However, this would require assigning a break to these employees first. Therefore, the time units that need to be scheduled to cover the additional $\Delta(l_s^{(1)})$ time units of order picking tasks equals

$$l_{s'}^{(1)} = \min\left\{\max\left\{\Delta(l_s^{(1)}) + \left(\left\lceil\frac{\Delta(l_s^{(1)})}{T_{break}}\right\rceil - 1\right)\cdot l_b, T_{min}\right\}, \Delta(l_s^{(1)}) + \left\lceil\frac{\Delta(l_s^{(1)})}{T_{break}}\right\rceil \cdot l_b\right\} \tag{40}$$

Consequently, the minimum total number of time units that the order pickers need to be scheduled is given by $LB^{(1)} = (N(l_s^{(1)}) - 1)\cdot l_s^{(1)} + l_{s'}^{(1)}$.

There can be one exceptional scenario. If the shift with a duration of T_{min} time units ends while taking a break (i.e., the condition in Equation (36) is not satisfied when $l_s = T_{min}$), then we also need to analyze shift lengths of $l_s^{(2)} = T_{min}$ time units. This means that it can happen that it is better to compensate order pickers for a partial break at the end of their shift instead of adding an entire block of $l_b + T_{break}$ time units to the shift length. Assigning the additional $\Delta(l_s^{(2)})$ time units to the $N(l_s^{(2)}) - 1$ order pickers who are already scheduled is more complex than the previous scenario, since the order pickers with a shift length of T_{min} time units end their shift while taking a break. It requires $\lceil \Delta(l_s^{(2)})/T_{break}\rceil$ order pickers to extend their shift with the additional break time of l'_b time units, where

$$l'_b = l_b - \left(T_{min} - \left\lfloor\frac{T_{min}}{T_{break}+l_b}\right\rfloor\cdot(T_{break}+l_b) - T_{break}\right) \tag{41}$$

Therefore, the time units that need to be scheduled to cover the additional $\Delta(l_s^{(2)})$ time units of order picking tasks equals

$$l_{s'}^{(2)} = \min\left\{T_{min}, \Delta(l_s^{(2)}) + \left\lceil\frac{\Delta(l_s^{(2)})}{T_{break}}\right\rceil\cdot l'_b\right\} \tag{42}$$

The corresponding minimum total number of time units that the order pickers need to be scheduled in this second scenario is given by $LB^{(2)} = (N(l_s^{(2)}) - 1)\cdot l_s^{(2)} + l_{s'}^{(2)}$.

The smallest value of $LB^{(1)}$ and $LB^{(2)}$ should be added as lower bound on the objective function. However, since order pickers are paid (or compensated) in integral multiples of the period length l, we include the following lower bound, LB, to the original problem formulation

$$\sum_{p\in P} m_p \geq LB = \begin{cases} l\cdot\left\lceil LB^{(1)}/l\right\rceil, & \text{if } T_{min} - \left\lfloor\frac{T_{min}}{T_{break}+l_b}\right\rfloor\cdot(T_{break}+l_b) \leq T_{break} \\ l\cdot\left\lceil \min\{LB^{(1)}, LB^{(2)}\}/l\right\rceil, & \text{otherwise}\end{cases} \tag{43}$$

Appendix C: Network flow formulation

Consider a graph $G = (V, A)$, where V is the set of nodes and A is the set of arcs. Next, we introduce O and D as dummy source and sink nodes, respectively. The start and end times of shifts are also indicated by dummy nodes and denoted by S and E, respectively. Each batch in I is also a node. Consequently, $V := \{\mathtt{o}\}\cup\{\mathtt{d}\}\cup I\cup S\cup E$. Let $r_i := 0$ and $d_i := T_{day}$ for $i \in \{\mathtt{o},\mathtt{d}\}$, whereas r_i andd_i present the start time of a shift for the nodes $i\in S$ and they present the end time of a shift for the nodes $i \in E$. The processing times t_i for the nodes $i\in V\setminus I$ are equal to zero. An example of the network graph is presented in Figure 10. In this network flow structure, traversing the graph is analogous to starting the shift at the time associated with node $i\in S$, executing the order picking tasks of the batches associated with nodes $i\in I$, and ending the shift at the time associated with node $i \in E$.

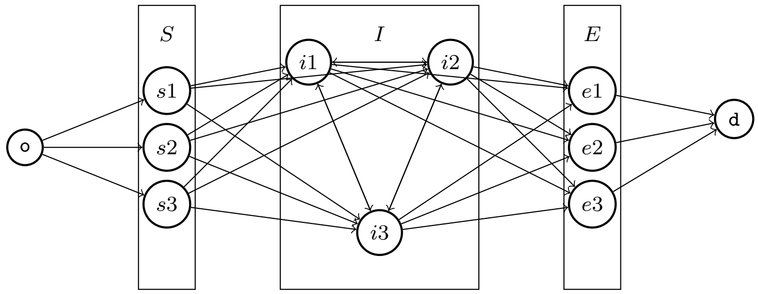


Figure 10 **A representation of a network graph with three shift start and end times and three batches**

The same input parameters are used as summarized in Table 2, whereas the following decision variables are used in the network flow formulation:

x_{ij}^p	is 1 if node $i \in V$ is visited before node $j \in V$ by order picker $p \in P$, else 0
y_i^p	is 1 if order picker $p \in P$ takes a break before visiting node $i \in I$, else 0
c_i^p	completion time at node $i \in N$ by order picker $p \in P$
b_i^p	number of time units without a break before completing the processing time t_i at node $i \in V$ by order picker $p \in P$
m_p	amount of time for which order picker $p \in P$ is compensated

The order picker scheduling problem (OPSP) is formulated as a network flow problem as follows:

$$\min \sum_{p \in P} m_p \tag{44}$$

subject to

$$\sum_{j \in S} x_{\mathtt{o}j}^p \leq 1 \qquad \forall p \in P \tag{45}$$

$$\sum_{i \in E} x_{i\mathtt{d}}^p \leq 1 \qquad \forall p \in P \tag{46}$$

$$\sum_{p \in P} \sum_{i \in I \cup S} x_{ij}^p = 1 \qquad \forall j \in I \tag{47}$$

$$\sum_{p \in P} \sum_{j \in I \cup E} x_{ij}^p = 1 \qquad \forall i \in I \tag{48}$$

$$\sum_{j \in V} x_{ij}^p = \sum_{j \in V} x_{ji}^p \qquad \forall i \in V \setminus \{\mathtt{o}, \mathtt{d}\}, p \in P \tag{49}$$

$$c_j^p \geq c_i^p + t_j + l_b y_j^p - (1 - x_{ij}^p) M \qquad \forall i \in V, j \in V, p \in P \tag{50}$$

$$c_i^p \geq r_i - (1 - \sum_{j \in N} x_{ij}^p) M \qquad \forall i \in V, p \in P \tag{51}$$

$$c_i^p \leq d_i + (1 - \sum_{j \in N} x_{ij}^p) M \qquad \forall i \in V, p \in P \tag{52}$$

$$b_j^p \geq t_j - (1 - x_{ij}^p) M \qquad \forall i \in S, j \in I, p \in P \tag{53}$$

$$b_j^p \geq b_i^p + (c_j^p - c_i^p) - (1 - x_{ij}^p + y_j^p) M \qquad \forall i \in I, j \in I, p \in P \tag{54}$$

$$b_i^p \geq t_i - (1 - y_i^p) M \qquad \forall i \in I, p \in P \tag{55}$$

$$b_i^p \leq T_{break} \qquad \forall i \in I, p \in P \tag{56}$$

$$\sum_{i \in I \cup S} x_{ij}^p \geq y_j^p \qquad \forall j \in I, p \in P \tag{57}$$

$$T_{min} \sum_{j \in S} x_{\mathtt{o}j}^p \leq m_p \qquad \forall p \in P \tag{58}$$

$$c_j^p - c_i^p \leq m_p \qquad \forall i \in S, j \in E, p \in P \tag{59}$$

$$\sum_{p \in P} \sum_{j \in S} x_{\mathtt{o}j}^p \leq p_{max} \tag{60}$$

$$\begin{aligned} &\sum_{p \in P} \Bigg(\sum_{j \in V \setminus S} x_{\mathtt{o}j}^p + \sum_{i \in S} \sum_{j \in V \setminus I} x_{ij}^p + \\ &\sum_{i \in I} \sum_{j \in V \setminus (I \cup E)} x_{ij}^p + \sum_{i \in E} \sum_{j \in V \setminus \{\mathtt{d}\}} x_{ij}^p + = 0 \\ &\sum_{j \in V} x_{\mathtt{d}j}^p \Bigg) \end{aligned} \tag{61}$$

$$b_i^p, c_i^p \geq 0 \qquad \forall i \in V, p \in P \tag{62}$$

$$x_{ij}^p \in \{0, 1\} \qquad \forall i \in V, j \in V, p \in P \tag{63}$$

$$y_i^p \in \{0, 1\} \qquad \forall i \in I, p \in P \tag{64}$$

$$0 \leq m_p \leq T_{max} \qquad \forall p \in P \tag{65}$$

The objective function is the same as the model formulation described in Section 3. Constraints (45) and (46) allow for an order picker to start and end her shift at most once. Constraints (47) and (48) ensure that each batch node is visited (i.e., executed) exactly once. Constraints (49) are flow conservation constraints for all nodes in S, I and E.

Constraints (50) determine the completion times at all nodes. Constraints (51) and (52) prevent earliness and tardiness, respectively, at the corresponding nodes. Constraints (53) and (54) determine the duration at nodes since the last break (including the processing times at the nodes). Constraints (55) reset the duration since the last break if a break is scheduled before visiting node i. Constraints (56) ensure that an order picker cannot be scheduled to visit nodes consecutively for more than T_{break} time units without a break. Constraints (57) make sure that a break before visiting node j can only be assigned to the same order picker who actually visits that node.

Constraints (58) ensure that an order picker is compensated for at least T_{min} time units if she is scheduled to work. Constraints (59) track the number of time units that an order picker is scheduled to work from the start of the shift to the end of the shift. Constraints (60) restrict the maximum number of order pickers that can be scheduled (i.e., paths that can be visited in the graph).

Constraints (61) ensure that the paths in the network can only traverse through the nodes of the graph that are allowed by restricting the flow on the arcs that are not allowed to be zero. Constraints (62) to (65) define the domain and range of the decision variables.

C.1. Additional constraints

The symmetry breaking constraints and additional constraints to tighten the model formulation from Appendix B can easily be translated to the network flow formulation. For instance, the equivalent of constraints (34) is

$$\sum_{j \in S} j x_{\circ j p-1} \leq \sum_{j \in S} j x_{\circ j p} \qquad \forall p \in P \setminus \{1\} \tag{66}$$

Constraints (35) can be transformed similarly and constraints (43) can directly be included without any modifications.

Appendix D: Extensions to OPSP formulation

A generic formulation of the OPSP and corresponding reduced problem is presented in Section 3 and Appendix G, respectively. However, there can be additional or different constraints for individual warehouses. In this appendix, we discuss and model full time employees as well as common break and labor law requirements that can easily be included to the original problem formulation.

Warehouses can have numerous restrictions pertaining the use of human labor dependent on specific labor laws. The European Union stipulates that all employees working more than 6 hours in a shift must be given a break, but the length of the break is specified by individual countries (European Parliament, Council of the European Union 2003). For example, the Dutch "Working Hours Act" mandates that each employee who works 5.5 hours must be given a break of 30 minutes, which could be reduced to 15 minutes conditional on agreements between the employer and labor unions[2] (Ministry of Social Affairs and Employment 2010). There can also be differences between companies within the same country or region, based on labor union agreements and company cultures regarding working time, start times and break times.

In the United States, the Fair Labor Standards Act (FLSA) does not require to give employees a rest or meal break. However, individual states can carry break laws in their legislature. Common constructs include a 10-15 minute rest break for every 3.5 or 4 hours worked, and an uninterrupted meal break of 30 minutes for employees who work more than five hours in a day. Such meal breaks must usually be provided sometime after the first two hours of work and before the last two hours of work. In Japan, the labor laws dictate that employees are entitled to a break of at least 45 minutes when they work six to eight hours and to a break of at least one hour when the working hours exceed eight hours. According to the Employment Standards Act in Canada, employees are entitled to at least 30 consecutive minutes of break during every 5-hour work period. The Employment Relations Act in New Zealand prescribes a 30-minute meal break and multiple 10-minute rest breaks based on the number of hours worked. Even though the Fair Work Act in Australia does not provide a statutory entitlement to any work breaks, the Fair Work Ombudsman prescribes similar rules as New Zealand.

The Fair Labor Standards Act (FLSA) in the United States does not address flexible work schedules. Alternative work arrangements such as flexible work schedules are a matter of agreement between the employer and the employee. This seems to be the standard in many countries. However, there are labour laws that specify a minimum compensation duration. For instance, employees in the United States, Canada and Australia must be paid for at least 3 hours each time they are required to report to work.

[2] A working day cannot exceed 11 hours and if a shift is longer than 10 hours, the total break time has to be at least 45 minutes. The overall break time can be split in multiple breaks, but need to be at least 15 consecutive minutes.

D.1. Full time employees

In our original problem formulation, we assume that all employees are flexible and part-time workers. However, if there are full-time employees with a fixed contract, the problem formulation can easily be extended to accommodate these circumstances. Consider the set of employees P, where the first $\overline{p}$ employees are full-timers and the remaining $p_{max} - \overline{p}$ employees are part-timers. If the shift start and end times are included in an employee's contract, this means that s_{jp} and e_{jp} for $p = 1, \dots, \overline{p}$ are predetermined values rather than decision variables. If only the shift length is included in the contract (let us denote this by l_s, which is not necessarily equal to T_{max}), then the duration between the start and end of the shift cannot exceed this:

$$\Big(\sum_{j \in E} j e_{jp} - \sum_{j \in S} (j-1) s_{jp}\Big) l \leq l_s, \qquad \forall p \in \{1, \dots, \overline{p}\} \tag{67}$$

However, full-time workers have a fixed compensation specified in their contractual agreements. Therefore, the values of m_p for $p = 1, \dots, \overline{p}$ are predetermined regardless of the tasks assigned to them.

Furthermore, there are often labor union agreements stipulating a maximum number of flexible employees that can be employed compared to the full-time employees (Bard et al. 2007). For instance, if there is a maximum ratio γ between the number of flexible order pickers divided by the number of full-time order pickers, then p_{max} is restricted:

$$\frac{p_{max} - \overline{p}}{\overline{p}} \leq \gamma \quad \Longleftrightarrow \quad p_{max} \leq (\gamma + 1)\overline{p}. \tag{68}$$

D.2. Maximum number of order pickers per time period

The total number of order pickers that are scheduled to work at the same time period can be restricted, for instance if equipment has a limited availability (such as pick trucks in a warehouse). This can be included in the original problem formulation by adding the following constraints:

$$\sum_{j'=1}^{j} \sum_{p \in P} (s_{j'p} - e_{j'-1,p}) \leq p^{eqp} \qquad \forall j \in J \tag{69}$$

where $e_{0,p} = 0$.

D.3. Designated break times and break time windows

In the original problem formulation, it is assumed that each order picker can take a break at any time. However, break times can be restricted to certain hours in the day because of opening times of cafeterias or labor union agreements. Kniffin et al. (2015) observe that commensality (i.e., eating together) is positively correlated to the performance of employees, and it may offer the only valuable and rare occasion for employees to socialize. Therefore, consider a set of periods $J' \subseteq J$ at which employees are allowed to take a break. The decision variables y_{kp} also need to include the time period. Define y_{jkp} as a binary decision variable that has the value 1 if a break is scheduled at the start of period $j \in J'$ which is the k-th position in the shift for order picker $p \in P$ and 0 otherwise. The following constraints make sure that $c_{kp} = (j-1) \cdot l + l_b$ if $y_{jkp} = 1$ for $j \in J'$ and c_{kp} is unconstrained otherwise:

$$c_{kp} + M \cdot (1 - y_{jkp}) \geq (j-1) \cdot l + l_b \qquad \forall j \in J', k \in K, p \in P \tag{70}$$

$$c_{kp} - M \cdot (1 - y_{jkp}) \leq (j-1) \cdot l + l_b \qquad \forall j \in J', k \in K, p \in P \tag{71}$$

Note that y_{kp} needs to be replaced by $\sum_{j\in J'} y_{jkp}$ in the original problem formulation.

More general, a break can also be restricted to start within a time window rather than to the beginning of a time period. Consequently, define J' as the set of break time windows over the planning horizon and $[r_j^{break}, d_j^{break}]$ as the start and end time of break window $j \in J'$. The decision variables y_{jkp} have a similar interpretation as before, except that it refers to a break in the j-th break time window (not the start of a time period). The following constraints make sure that the break starts and finishes within the associated break time window if $y_{jkp} = 1$ for $j \in J'$ and c_{kp} is unconstrained otherwise:

$$c_{kp} + M(1 - y_{jkp}) \geq r_j^{break} + l_b \qquad \forall j \in J', k \in K, p \in P \tag{72}$$

$$c_{kp} - M(1 - y_{jkp}) \leq d_j^{break} \qquad \forall j \in J', k \in K, p \in P \tag{73}$$

D.4. Multiple types of breaks

Company culture and contractual agreements can require to give order pickers multiple types of breaks. For instance, several short breaks of up to 15 minutes are given during a shift in addition to a lunch break of 30 minutes. Let Φ indicate the set of various types of breaks. For each break type $\phi \in \Phi$, denote the maximum amount of time an order picker can work without a break of type ϕ by T^{ϕ}_{break}, and the length of a break of type ϕ by l_b^{ϕ}. We should note that breaks often have a hierarchy between them, i.e., a break can reset the working time accumulated for a different type of break. For example, employees are given a coffee break of 15 minutes after 3 hours of work and a lunch break of 30 minutes after 5 hours of work. The lunch break will reset the working time accumulated for the coffee break but not the other way around. Let Φ^{ϕ} denote the set of breaks that reset the accumulated working time for break type $\phi \in \Phi$. To continue the example, the coffee break is type 1 and the lunch break is type 2. Consequently, $\Phi = \{1, 2\}$, $T^1_{break} = \{3\}$, $T^2_{break} = \{5\}$, $\Phi^1 = \{2\}$ and $\Phi^2 = \varnothing$.

We introduce a set of binary decision variables y^{ϕ}_{kp} that indicate if the break of type ϕ is scheduled at the k-th position in the shift for order picker $p \in P$ and 0 otherwise. The following constraints include the hierarchy as discussed earlier:

$$c_{kp} - \left(c_{hp} - \sum_{i\in I} t_i x_{ihp}\right) \leq T^{\phi}_{break} + M\left(\sum_{\phi'\in\Phi^{\phi}} \sum_{k'=h+1}^{k} y^{\phi'}_{k'p}\right) \qquad \forall k \in K, h < k, p \in P, \phi \in \Phi \tag{74}$$

Furthermore, y_{kp} needs to be replaced by $\sum_{\phi\in\Phi} y^{\phi}_{kp}$ and $l_b y_{kp}$ by $\sum_{\phi\in\Phi} y^{\phi}_{kp} l^{\phi}_b$ in the original problem formulation.

Appendix E: Pseudocode of Labeling Algorithm

Algorithm 1 Dynamic programming algorithm for the pricing problem

```
𝓛_i := {(i, T_min + ψ, T_i, (V_h^1 = 0, ..., V_h^{|V'|} = 0))}   i ∈ S     ▷ Initialization at shift start nodes
for i ∈ V' \ S do
    𝓛_i := ∅                                                           ▷ Initialization at remaining nodes
Δ := S                                                                  ▷ Unvisited vertices
while Δ ≠ ∅ do
    for i ∈ Δ do
        for j ∈ δ(i) do                                                 ▷ Feasible nodes only
            for L_h^i ∈ 𝓛_i do
                if V_{L_h^i}^j = 0 then                                 ▷ If j is not already visited
                    T_j := f(T_i, j)                                    ▷ Extension without break
                    if T_j is feasible then
                        ADD L_h^j = (j, c_h^j, T_j, V_h) to 𝓛_j
                    T_j := g(T_i, j)                                    ▷ Extension with break
                    if T_j is feasible then
                        ADD L_h^j = (j, c_h^j, T_j, V_h) to 𝓛_j
            if 𝓛_j changed then
                𝓛_j ← DOMINANCE(𝓛_j)                                   ▷ Eliminate dominated lables
                Δ ← Δ ∪ {j}
        Δ ← Δ \ {i}
return arg min_{L ∈ 𝓛_D} {c_L^D}
```

Appendix F: Proof of Proposition 2

Let x^* indicate a sequence of batches with cost $z(x^*)$ assigned to an order picker in the optimal solution where batch i precedes batch j. The completion times for batch i and batch j in x^* are presented by c_i^* and c_j^*, respectively. Furthermore, let γ_i^* be the earliest time the order picker can start picking items for batch i such that the sequence x^* remains optimal and q_j^* is the latest time the order picker can start to work on the batch succeeding batch j such that the sequence x^* remains optimal.

When the performance of the batches i and j is reversed, the new sequence is indicated by x' with cost $z(x')$ and the completion times of the batches i and j become c'_i and c'_j, respectively. To proof Proposition 2, it is sufficient to show that the new solution is feasible and the cost $z(x')$ equals $z(x^*)$. There are four possible sequences to consider in regard to scheduling breaks before performing batch i and batch j.

Case I: batch i and j are performed without a break before i or between i and j (i.e., $i|j$)

Sub-case (i): $\gamma_i^* + t_i < r_i = c_i^*$ and $\gamma_i^* + t_i + t_j < c_j^* \leq q_j^*$ (i.e., there is a wait between γ_i^* and c_i^*)

In the new solution x', the order picker might need to wait before the completion of batch j. If the order picker needs to wait, $\gamma_i^* + t_j < r_j = c_j'$ and $\gamma_i^* + t_j + t_i < c_i' \leq q_j^*$. If the order picker does not need to wait, $\gamma_i^* + t_j \geq r_i = c_j'$ and $\gamma_i^* + t_j + t_i = c_i' \leq q_j^*$. In either case, q_j^* is not violated and the new sequence is feasible with the same cost.

Sub-case (ii): $\gamma_i^* + t_i = c_i^* \geq r_i$ and $\gamma_i^* + t_i + t_j = c_j^* \leq q_j^*$ (i.e., there is no wait between γ_i^* and c_i^*)

When the batches are reversed, there would still be no waiting time as $t_j > t_i$. Consequently, $\gamma_i^* + t_j = c_j' > r_i$ and $\gamma_i^* + t_j + t_i = c_i' \leq q_j^*$. Furthermore, q_j^* is respected with the reversal of the batches. Consequently, $z(x') = z(x^*)$.

Case II: batch i and j are performed with a break before i but not between i and j (i.e., $y|i|j$)

Sub-case (i): $\gamma_i^* + l_b + t_i < r_i = c_i^*$ and $\gamma_i^* + l_b + t_i + t_j < c_j^* \leq q_j^*$ (i.e., there is a wait between γ_i^* and c_i^*)

If there is a wait in the new solution x', then $\gamma_i^* + l_b + t_j < r_j = c_j'$ and $\gamma_i^* + l_b + t_j + t_i < c_i' \leq q_j^*$. If the order picker does not need to wait, then $\gamma_i^* + l_b + t_j \geq r_i = c_j'$ and $\gamma_i^* + l_b + t_j + t_i = c_i' \leq q_j^*$. In either case, q_j^* is not violated, and the new sequence is feasible with the same cost. Furthermore, the amount of work done between the end of the break and the completion of batch i does not increase.

Sub-case (ii): $\gamma_i^* + l_b + t_i \geq r_i = c_i^*$ and $\gamma_i^* + l_b + t_i + t_j = c_j^* \leq q_j^*$ (i.e., there is no wait between γ_i^* and c_i^*)

The reversal of the batches i and j does not introduce any waiting time for the order picker. Furthermore, the amount of time since the break remains the same.

Case III: batch i and j are performed without a break before i but between i and j (i.e., $i|y|j$) Let the amount of work since the last break or since the start when the first batch is picked until the beginning when batch i is picked be denoted by b_r. If the sequence in which the batches i and j are performed is reversed, it is possible that $b_r + t_j > T_{break}$, which results in an infeasible solution. Therefore, batch j can not always precede batch i in this case.

Case IV: batch i and j are performed with a break before i and between i and j (i.e., $y|i|y|j$)

The same proof as Case I can be used, where the new value of t_i is increased by l_b and the new value of t_j by l_b.

Appendix G: Optimal schedule for one order picker

In the savings algorithm in Section 5.1 and the large neighborhood search in Section 5.2, the assignment of batches to an order picker is altered. In this appendix, we present an MILP formulation to optimally schedule the batches to an individual order picker such that the due time windows and break constraints are satisfied. Before we try to solve the reduced OPSP formulation with one order picker, we perform a set of infeasibility checks to verify whether a feasible solution can be found.

G.1. Infeasibility conditions

Even though the computation time of the MILP for the reduced problem is short, infeasibility checks can be performed first as pre-processing step to easily verify whether the order picking tasks cannot be combined in a feasible schedule. Let the set of batches to be included in the schedule be denoted by $B \subseteq I$. If any of the following infeasibility conditions is satisfied, the reduced problem is infeasible. The complexity of these checks are indicated within parentheses.

- $\sum_{i\in B} t_i + \left(\left\lceil (\sum_{i\in B} t_i)/T_{break}\right\rceil - 1\right) l_b > T_{max}$, (complexity: $\mathcal{O}(|B|)$), the total processing time of the batches and the necessary break times exceeds the maximum shift length.
- $\max_{i\in B}\{r_i\} - \min_{i\in B}\{d_i - t_i\} > T_{max}$, (complexity: $\mathcal{O}(|B|)$), the time difference between the latest earliest completion time and the earliest latest start time exceeds the maximum shift length (i.e., there are tasks that cannot be scheduled to be completed earlier or to start later such that the maximum shift length is not exceeded).
- $|B| + \left(\left\lceil (\sum_{i\in B} t_i)/T_{break}\right\rceil - 1\right) > \bar{k}$, (complexity: $\mathcal{O}(|B|)$), the minimum number of tasks (either order picking batches or breaks) to cover the workload in B exceeds the maximum number of tasks in a shift.
- $(\sum_{i\in B} t_i) > \max_{i\in B}\{d_i\} - (\min_{i\in B}\{r_i - t_i\})^+$, (complexity: $\mathcal{O}(|B|)$), the total processing time of all batches in B cannot be assigned to one order picker between the earliest start time and the latest completion time.
- $\exists i, j \in B : (r_i + t_j > d_j) \wedge (r_j + t_i > d_i)$, (complexity: $\mathcal{O}(|B|^2)$), the due time windows prohibit the order picker to perform batch j after i or to perform batch i after j (i.e., the due time windows are violated if both tasks need to be performed by the same order picker).
- Define the earliest start time of task $i \in B$ as $\gamma_i := r_i - t_i$. Let Γ_B and D_B denote the earliest start times and latest completion times of the batches in B, respectively. For any earliest start time $\gamma \in \Gamma_B$ and latest completion time $d \in D_B$ where $\gamma < d$, the set $I_{\gamma d}$ represents all batches in B that need to start at or after time γ and be completed at time d at the latest, i.e., $I_{\gamma d} := \{i \in B | (\gamma_i \geq \gamma) \wedge (d_i \leq d)\}$. The condition $\exists \gamma \in \Gamma_B, d \in D_B : \sum_{i\in I_{\gamma d}} t_i > (d - \gamma)$ indicates that the work load between γ and d cannot be completed by one order picker (complexity: $\mathcal{O}(|B|^3)$).

G.2. Reduced OPSP formulation with one order picker

When none of the infeasibility conditions are satisfied, we try to optimally sequence the tasks to the individual order picker while satisfying all constraints. This means that the original OPSP formulation of Section 3 is simplified by removing the multiple order pickers $p \in P$. Consequently, the set of batches that need to be scheduled for the single order picker is denoted by B and the decision variables in this reduced problem become the following:

x_{ik}	is 1 if batch $i \in B$ is scheduled to be picked at the k^{th} position in the shift for the order picker, where $k \in K$, else 0
y_k	is 1 if a break is scheduled at the k^{th} position in the shift for the order picker, where $k \in K$, else 0
s_j	is 1 if the order picker starts the shift at the beginning of period $j \in S$, else 0
e_j	is 1 if the order picker ends the shift at the end of period $j \in E$, else 0
c_k	completion time of the task scheduled at the k^{th} position in the shift of the order picker, where $k \in K$
m	amount of time for which the order picker is compensated

The reduced problem (RP) of the OPSP is formulated as a MILP model as follows:

RP:

$$\min m \tag{75}$$

subject to

$$\sum_{i\in B} x_{ik} + y_k \leq 1 \qquad \forall k \in K \qquad (76)$$

$$\sum_{k\in K} x_{ik} = 1 \qquad \forall i \in B \qquad (77)$$

$$\sum_{j\in J\setminus S} s_j + \sum_{j\in J\setminus E} e_j = 0 \qquad (78)$$

$$c_1 \geq \Big(\sum_{j\in S}(j-1)s_j\Big)l + \sum_{i\in B} t_i x_{i1} + l_b y_1 \qquad (79)$$

$$c_k \geq c_{k-1} + \sum_{i\in B} t_i x_{ik} + l_b y_k \qquad \forall k \in K\setminus\{1\} \qquad (80)$$

$$\sum_{j\in E}(je_j)l \geq c_{\bar{k}} \qquad (81)$$

$$c_k + M(1-x_{ik}) \geq r_i \qquad \forall i \in B, k \in K \qquad (82)$$

$$c_k - M(1-x_{ik}) \leq d_i \qquad \forall i \in B, k \in K \qquad (83)$$

$$c_k - \Big(c_h - \sum_{i\in B} t_i x_{ih}\Big) \leq T_{break} + M\Big(\sum_{k'=h+1}^{k} y_{k'}\Big) \qquad \forall h, k \in K, h < k \qquad (84)$$

$$\sum_{i\in B} x_{ik-1} + y_{jk-1} \geq \sum_{i\in B} x_{ik} + y_{jk} \qquad \forall k \in K\setminus\{1\} \qquad (85)$$

$$(\sum_{j\in E} je_j - \sum_{j\in S}(j-1)s_j)l \leq m \qquad (86)$$

$$c_k \geq 0 \qquad \forall k \in K \qquad (87)$$

$$x_{ik} \in \{0,1\} \qquad \forall i \in B, k \in K \qquad (88)$$

$$y_k \in \{0,1\} \qquad \forall k \in K \qquad (89)$$

$$s_j, e_j \in \{0,1\} \qquad \forall j \in J \qquad (90)$$

$$\max\left\{T_{min}, \left\lceil\left(\sum_{i\in B} t_i + \Big(\Big\lceil(\sum_{i\in B} t_i)/T_{break}\Big\rceil - 1\Big)l_b\right)/l\right\rceil l\right\} \leq m \leq T_{max} \qquad (91)$$

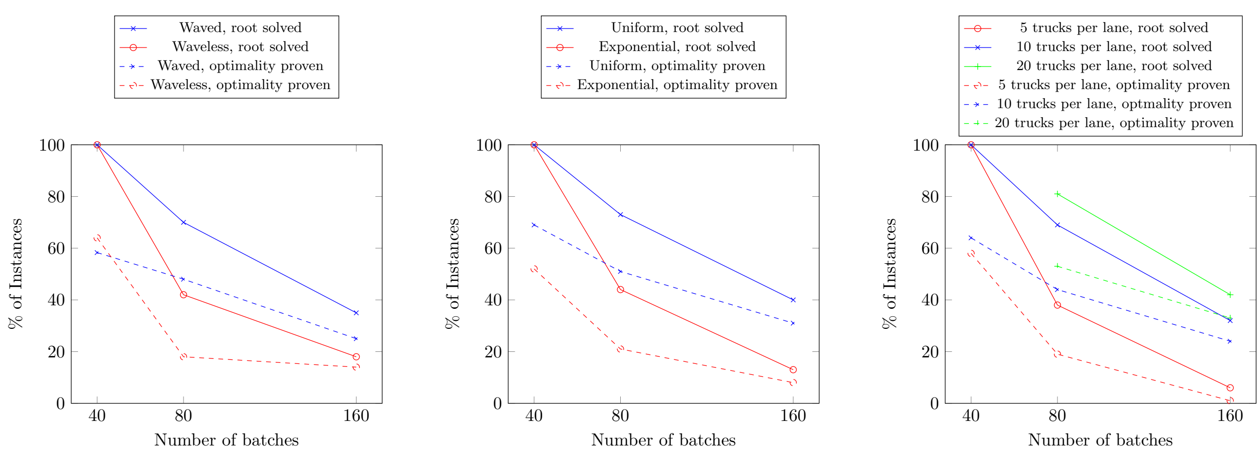
There is a one-to-one relation with the constraints in this RP formulation and the OPSP formulation in Section 3.

This reduced problem is strongly NP-hard as it is a generalization of the minimum tour duration problem (MTDP), which is known to be strongly NP-hard (Tilk and Irnich 2017). However, we can use the overall architecture of the solution approaches for the MTDP to efficiently solve the MILP since the architecture allows us to incorporate our additional operational constraints. Furthermore, the scale of the reduced problem as a subroutine of the savings algorithm and the LNS is small enough to be practically solvable. The abovementioned infeasibility tests also limit the number of times that the reduced problem is solved unnecessarily.

Appendix H: Detailed result of Section 6.2

Table 6 until 13 provide the results of all individual numeral instances that are summarized in Table 4. For the branch-and-price algorithm, we report the optimal solution and best integer solution found by column generation at the root node (this corresponds to lower bound LB^{LP} and upper bound UB^{LP}, respectively) as well as the best lower bound from branching and the best integer solution after branching (denoted by LB^{IP} and UB^{IP}, respectively). The computational time to solve the root node, perform branching and perform the overall branch-and-price algorithm is indicated by CPU^{LP}, CPU^{IP} and CPU^{BP}, respectively (all in seconds). The relative cost increase of the solutions found by the savings algorithm, commercial solver Gurobi Optimizer and the metaheuristic compared to performance of the branch-and-price algorithm is indicated by $\%\Delta^{S}$, $\%\Delta^{GUR}$ and $\%\Delta^{MH}$, respectively, where $\%\Delta^{X} = (z(X) - z(BP))/z(BP) \times 100$ and $z(X)$ is the best integer solution found by solution procedure X. The computational time to perform solution procedure X is provided by CPU^{X} (in seconds) for solution procedure X. Note that we terminate the solution procedure when the computational time reaches 1,800 seconds.

As discussed in Section 6.2, the performance of the branch-and-price algorithm depends on the truck departure pattern (waved or waveless), processing time distribution (uniform or exponential), and number of truck departures per staging lane. Figure 11a shows that waved instances are more difficult to solve than waveless instances. Figure 11b shows that instances with exponentially distributed processing times are more difficult to solve than when the processing times have a uniform distribution. Finally, Figure 11c shows that instances with more truck departures per staging lane are easier to solve than instances with less truck departures per staging lane.



(a) Waved and waveless instances

(b) Uniform and exponential order processing times

(c) Number of truck departures per staging lane

Figure 11 **Relative number of instances for which either the root node is solved or the optimal solution is found with the branch-and-price algorithm**

Table 6 Results for Instances with 40 batches, 5 truck departures per staging lane

	Branch & Price							**S**		**GUR**		**MH**	
	Linear Relaxation			Integer									
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{GUR}$	CPU^{GUR}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-40-45-SSTR1	4480.0	4800.0	1.5	4500.0	4800.0	1796.6	1800.0	0.0	1.9	0.0	1800.0	0.0	47.6
U-Waved-40-45-SSTR2	4200.0	4440.0	1.4	4260.0	4440.0	1795.1	1800.0	8.1	3.5	0.0	1800.0	-1.4	49.8
U-Waved-40-45-SSTR3	4152.0	4200.0	0.9	4200.0	4200.0	2.8	6.4	2.9	2.8	0.0	1800.0	0.0	44.8
U-Waved-40-45-SSTR4	4480.0	4800.0	5.2	4500.0	4800.0	1792.7	1800.0	0.0	2.1	0.0	1800.0	0.0	57.1
U-Waved-40-45-SSTR5	3480.0	3660.0	146.4	3480.0	3660.0	1650.4	1800.0	4.9	3.2	3.3	1800.0	0.0	195.2
U-Waved-40-45-SSTR6	2685.0	2760.0	4.7	2700.0	2700.0	100.3	107.8	4.4	2.8	4.4	1800.0	2.2	55.9
U-Waved-40-75-SSTR1	4980.0	4980.0	0.6	4980.0	4980.0	2.3	5.2	18.1	2.3	0.0	1800.0	0.0	31.2
U-Waved-40-75-SSTR2	4740.0	4740.0	0.8	4740.0	4740.0	2.5	5.9	10.1	2.5	3.8	1800.0	0.0	31.3
U-Waved-40-75-SSTR3	4660.0	4740.0	0.5	4680.0	4680.0	5.3	8.3	2.6	2.5	1.3	1800.0	1.3	32.2
U-Waved-40-75-SSTR4	4800.0	4800.0	0.8	4800.0	4800.0	2.5	5.7	21.3	2.5	0.0	1800.0	0.0	37.5
U-Waved-40-75-SSTR5	3840.0	3840.0	2.7	3840.0	3840.0	2.5	7.7	15.6	2.4	0.0	1800.0	0.0	40.3
U-Waved-40-75-SSTR6	3400.0	3480.0	1.1	3420.0	3420.0	5.4	9.0	12.3	2.5	5.3	1800.0	0.0	39.4
U-Waved-40-105-SSTR1	5600.0	5760.0	0.5	5640.0	5760.0	1797.5	1800.0	19.8	1.9	0.0	1800.0	0.0	25.4
U-Waved-40-105-SSTR2	5280.0	5400.0	3.4	5280.0	5400.0	1794.3	1800.0	22.2	2.4	5.6	1800.0	0.0	29.1
U-Waved-40-105-SSTR3	5160.0	5160.0	0.2	5160.0	5160.0	2.5	5.1	2.3	2.5	0.0	1800.0	0.0	31.2
U-Waved-40-105-SSTR4	5380.0	5520.0	5.9	5400.0	5520.0	1792.0	1800.0	31.5	2.0	5.4	1800.0	0.0	36.8
U-Waved-40-105-SSTR5	4936.0	5040.0	9.6	4980.0	4980.0	1792.1	1804.1	21.7	2.4	9.6	1800.0	1.2	45.2
U-Waved-40-105-SSTR6	4771.4	4800.0	3.4	4800.0	4800.0	2.8	8.9	11.3	2.7	8.8	1800.0	0.0	37.6
U-Waveless-40-45-SSTR1	3040.0	3360.0	1.5	3060.0	3360.0	1796.1	1800.0	1.8	2.4	0.0	1800.0	0.0	59.0
U-Waveless-40-45-SSTR2	2360.0	2520.0	2.0	2400.0	2520.0	1795.9	1800.0	9.5	2.1	0.0	1800.0	0.0	123.0
U-Waveless-40-45-SSTR3	2230.0	2340.0	12.3	2280.0	2280.0	99.4	114.8	10.5	3.0	7.9	1800.0	0.0	113.8
U-Waveless-40-45-SSTR4	2640.0	2940.0	117.2	2640.0	2940.0	1680.6	1800.0	16.3	2.3	0.0	1800.0	0.0	360.0
U-Waveless-40-45-SSTR5	2280.0	2340.0	477.3	2340.0	2340.0	2.9	483.1	25.6	2.9	7.7	1800.0	0.0	360.0
U-Waveless-40-45-SSTR6	2236.0	2280.0	10.3	2280.0	2280.0	2.5	15.3	13.2	2.5	7.9	1800.0	2.6	81.9
U-Waveless-40-75-SSTR1	4128.0	4320.0	7.4	4140.0	4320.0	1790.8	1800.0	22.2	1.8	0.0	1800.0	0.0	33.0
U-Waveless-40-75-SSTR2	3540.0	3600.0	3.8	3540.0	3540.0	17.9	23.6	39.0	2.0	6.8	1800.0	1.7	38.4
U-Waveless-40-75-SSTR3	3360.0	3420.0	4.4	3360.0	3360.0	37.2	43.5	12.5	1.9	8.9	1800.0	1.8	40.7
U-Waveless-40-75-SSTR4	3720.0	3900.0	125.3	3720.0	3900.0	1672.8	1800.0	24.6	2.0	10.8	1800.0	1.5	311.6
U-Waveless-40-75-SSTR5	3441.2	3540.0	207.6	3480.0	3540.0	1589.6	1800.0	10.2	2.8	6.8	1800.0	0.0	242.4
U-Waveless-40-75-SSTR6	3286.7	3360.0	15.1	3300.0	3360.0	1782.1	1800.0	12.5	2.8	7.1	1800.0	0.0	57.6
U-Waveless-40-105-SSTR1	6880.0	7200.0	0.4	6900.0	7200.0	1797.4	1800.0	13.3	2.3	0.0	1800.0	0.0	22.8
U-Waveless-40-105-SSTR2	5420.0	5460.0	0.4	5460.0	5460.0	2.3	5.0	20.9	2.3	1.1	1800.0	1.1	25.3
U-Waveless-40-105-SSTR3	4760.0	4800.0	0.5	4800.0	4800.0	1.8	4.1	5.0	1.8	0.0	1800.0	0.0	27.9
U-Waveless-40-105-SSTR4	5760.0	5760.0	1.6	5760.0	5760.0	2.0	5.5	41.7	1.9	0.0	1800.0	0.0	26.3
U-Waveless-40-105-SSTR5	5040.0	5040.0	0.9	5040.0	5040.0	2.1	5.0	28.6	2.1	16.7	1800.0	0.0	28.3
U-Waveless-40-105-SSTR6	4620.0	4620.0	0.8	4620.0	4620.0	2.3	5.3	15.6	2.2	10.4	1800.0	0.0	30.7
E-Waved-40-45-SSTR1	4140.0	4380.0	2.6	4140.0	4380.0	1795.1	1800.0	0.0	2.2	0.0	1800.0	0.0	65.0
E-Waved-40-45-SSTR2	3769.7	3960.0	1.4	3780.0	3960.0	1795.1	1800.0	4.5	3.6	4.5	1800.0	0.0	99.4
E-Waved-40-45-SSTR3	3718.3	3780.0	1.2	3720.0	3780.0	1795.6	1800.0	1.6	3.2	0.0	1800.0	0.0	69.1
E-Waved-40-45-SSTR4	4080.0	4320.0	2.2	4080.0	4320.0	1795.7	1800.0	11.1	2.2	0.0	1800.0	0.0	62.8
E-Waved-40-45-SSTR5	3156.6	3360.0	846.4	3180.0	3360.0	950.2	1800.0	8.9	3.3	7.1	1800.0	0.0	360.0
E-Waved-40-45-SSTR6	2464.5	2580.0	45.8	2520.0	2520.0	1757.1	1806.0	4.8	3.2	4.8	1800.0	0.0	173.9
E-Waved-40-75-SSTR1	5580.0	5880.0	0.7	5580.0	5880.0	1797.1	1800.0	0.0	2.2	0.0	1800.0	0.0	30.1
E-Waved-40-75-SSTR2	5046.0	5220.0	0.6	5100.0	5220.0	1796.7	1800.0	3.4	2.7	3.4	1800.0	0.0	31.5
E-Waved-40-75-SSTR3	5036.0	5100.0	4.5	5040.0	5100.0	1792.9	1800.0	2.4	2.7	2.4	1800.0	0.0	32.8
E-Waved-40-75-SSTR4	5460.0	5760.0	1.2	5460.0	5760.0	1796.9	1800.0	17.7	1.9	8.3	1800.0	0.0	29.9
E-Waved-40-75-SSTR5	4530.0	4620.0	1.3	4560.0	4620.0	1795.9	1800.0	9.1	2.9	9.1	1800.0	0.0	37.2
E-Waved-40-75-SSTR6	4260.0	4260.0	4.7	4260.0	4260.0	3.0	10.8	7.0	3.0	7.0	1800.0	0.0	34.2
E-Waved-40-105-SSTR1	5460.0	5460.0	0.1	5460.0	5460.0	2.2	4.5	7.7	2.2	7.7	1800.0	0.0	42.4
E-Waved-40-105-SSTR2	4980.0	4980.0	0.2	4980.0	4980.0	2.1	4.4	8.4	2.1	4.8	1800.0	0.0	37.7
E-Waved-40-105-SSTR3	4860.0	4860.0	0.2	4860.0	4860.0	2.8	5.8	3.7	2.8	3.7	1800.0	0.0	37.2
E-Waved-40-105-SSTR4	4800.0	4800.0	0.9	4800.0	4800.0	2.3	5.4	22.5	2.2	10.0	1800.0	0.0	40.1
E-Waved-40-105-SSTR5	3850.9	3900.0	349.6	3900.0	3900.0	2.6	354.7	21.5	2.5	7.7	1800.0	0.0	360.0
E-Waved-40-105-SSTR6	3448.2	3480.0	63.2	3480.0	3480.0	3.6	70.4	15.5	3.6	15.5	1800.0	0.0	273.4
E-Waveless-40-45-SSTR1	2920.0	2940.0	92.3	2940.0	2940.0	2.0	96.3	14.3	2.0	14.3	1800.0	0.0	235.0
E-Waveless-40-45-SSTR2	2352.3	2520.0	62.5	2400.0	2400.0	1743.2	1807.9	10.0	2.2	5.0	1800.0	5.0	140.0
E-Waveless-40-45-SSTR3	2101.8	2160.0	14.5	2160.0	2160.0	2.8	20.1	5.6	2.8	5.6	1800.0	0.0	113.9
E-Waveless-40-45-SSTR4	2720.0	2940.0	392.0	2760.0	2940.0	1406.0	1800.0	14.3	1.9	14.3	1800.0	0.0	360.0
E-Waveless-40-45-SSTR5	2349.9	2460.0	104.1	2400.0	2400.0	1700.6	1807.1	32.5	2.5	15.0	1800.0	5.0	357.6
E-Waveless-40-45-SSTR6	2111.8	2160.0	7.2	2160.0	2160.0	3.0	13.2	13.9	3.0	5.6	1800.0	0.0	123.6
E-Waveless-40-75-SSTR1	5070.0	5340.0	0.8	5100.0	5340.0	1797.5	1800.0	18.0	1.7	9.0	1800.0	0.0	28.8
E-Waveless-40-75-SSTR2	4121.5	4260.0	82.8	4140.0	4260.0	1714.9	1800.0	21.1	2.4	19.7	1800.0	0.0	194.9
E-Waveless-40-75-SSTR3	3983.4	4020.0	10.1	4020.0	4020.0	1.9	13.9	16.4	1.9	9.0	1800.0	0.0	53.1
E-Waveless-40-75-SSTR4	4294.5	4440.0	327.9	4320.0	4380.0	1470.3	1800.0	23.3	1.9	23.3	1800.0	1.4	360.0
E-Waveless-40-75-SSTR5	4056.1	4140.0	231.3	4080.0	4080.0	1446.5	1680.4	19.1	2.7	14.7	1800.0	1.5	351.1
E-Waveless-40-75-SSTR6	3936.5	4020.0	15.5	3960.0	3960.0	152.3	170.6	22.7	2.8	6.1	1800.0	0.0	51.3
E-Waveless-40-105-SSTR1	5280.0	5280.0	0.2	5280.0	5280.0	1.6	3.3	0.0	1.6	0.0	1800.0	0.0	34.3
E-Waveless-40-105-SSTR2	3960.0	3960.0	1.5	3960.0	3960.0	1.7	4.8	6.1	1.7	1.5	1800.0	0.0	34.8
E-Waveless-40-105-SSTR3	3402.0	3540.0	2.8	3480.0	3480.0	519.3	523.8	6.9	1.7	6.9	1800.0	0.0	35.1
E-Waveless-40-105-SSTR4	4800.0	4800.0	0.6	4800.0	4800.0	2.3	5.3	10.0	2.3	10.0	1800.0	0.0	40.7
E-Waveless-40-105-SSTR5	3900.0	3900.0	25.4	3900.0	3900.0	1.9	29.1	10.8	1.8	7.7	1800.0	0.0	82.4
E-Waveless-40-105-SSTR6	3400.9	3480.0	3.2	3420.0	3480.0	1794.8	1800.0	6.9	2.0	3.4	1800.0	0.0	48.3

Table 7 Results for Instances with 40 batches, 10 truck departures per staging lane

	Branch & Price							S		GUR		MH	
	Linear Relaxation			Integer									
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{GUR}$	CPU^{GUR}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-40-45-SSTR1	4800.0	4800.0	0.6	4800.0	4800.0	1.8	4.2	10.0	1.8	10.0	1800.0	0.0	30.5
U-Waved-40-45-SSTR2	4140.0	4140.0	0.3	4140.0	4140.0	2.1	4.5	8.7	2.1	1.4	1800.0	0.0	32.4
U-Waved-40-45-SSTR3	3660.0	3660.0	0.2	3660.0	3660.0	2.0	4.3	9.8	2.0	9.8	1800.0	0.0	33.1
U-Waved-40-45-SSTR4	2880.0	2880.0	16.4	2880.0	2880.0	2.0	20.3	50.0	2.0	33.3	1800.0	0.0	43.1
U-Waved-40-45-SSTR5	2700.0	3000.0	6.0	2760.0	2820.0	1791.8	1800.0	6.4	2.2	6.4	1800.0	6.4	61.7
U-Waved-40-45-SSTR6	2640.0	2820.0	1.3	2700.0	2760.0	1796.3	1800.0	2.2	2.4	2.2	1800.0	2.2	43.2
U-Waved-40-75-SSTR1	6240.0	6240.0	0.2	6240.0	6240.0	1.7	3.6	8.7	1.7	0.0	1800.0	0.0	22.9
U-Waved-40-75-SSTR2	5400.0	5400.0	0.2	5400.0	5400.0	1.7	3.5	6.7	1.7	1.1	1800.0	0.0	26.1
U-Waved-40-75-SSTR3	4920.0	4920.0	0.1	4920.0	4920.0	1.7	3.6	8.5	1.7	1.2	1800.0	0.0	23.8
U-Waved-40-75-SSTR4	4320.0	4320.0	0.4	4320.0	4320.0	1.8	3.9	11.1	1.7	0.0	1800.0	0.0	27.3
U-Waved-40-75-SSTR5	3780.0	3780.0	2.5	3780.0	3780.0	1.9	6.1	17.5	1.8	3.2	1800.0	0.0	32.2
U-Waved-40-75-SSTR6	3630.0	3660.0	1.0	3660.0	3660.0	2.0	4.9	11.5	2.0	6.6	1800.0	1.6	30.5
U-Waved-40-105-SSTR1	7680.0	7680.0	0.1	7680.0	7680.0	1.5	3.2	6.3	1.5	0.0	1800.0	0.0	21.3
U-Waved-40-105-SSTR2	6960.0	6960.0	0.1	6960.0	6960.0	1.5	3.2	8.6	1.5	4.3	1800.0	0.0	22.4
U-Waved-40-105-SSTR3	6480.0	6480.0	0.2	6480.0	6480.0	1.6	3.4	0.9	1.6	0.0	1800.0	0.0	23.1
U-Waved-40-105-SSTR4	5760.0	5760.0	0.5	5760.0	5760.0	1.5	3.6	26.0	1.5	0.0	1800.0	0.0	24.2
U-Waved-40-105-SSTR5	5220.0	5220.0	0.6	5220.0	5220.0	1.7	3.9	26.4	1.6	14.9	1800.0	0.0	27.2
U-Waved-40-105-SSTR6	5040.0	5040.0	0.3	5040.0	5040.0	1.9	4.0	0.0	1.9	0.0	1800.0	0.0	27.4
E-Waved-40-45-SSTR1	4000.0	4320.0	0.6	4020.0	4320.0	1797.3	1800.0	0.0	2.1	0.0	1800.0	0.0	43.1
E-Waved-40-45-SSTR2	3410.0	3600.0	0.6	3420.0	3600.0	1797.3	1800.0	0.0	2.1	0.0	1800.0	0.0	40.4
E-Waved-40-45-SSTR3	3110.0	3300.0	0.7	3120.0	3300.0	1797.0	1800.0	7.3	2.3	0.0	1800.0	0.0	40.9
E-Waved-40-45-SSTR4	2800.0	2940.0	3.3	2820.0	2940.0	1794.7	1800.0	16.3	2.1	0.0	1800.0	0.0	65.9
E-Waved-40-45-SSTR5	2600.0	2640.0	6.4	2640.0	2640.0	2.0	10.3	13.6	2.0	9.1	1800.0	0.0	66.0
E-Waved-40-45-SSTR6	2580.0	2640.0	4.7	2580.0	2640.0	1793.1	1800.0	6.8	2.3	4.5	1800.0	0.0	71.6
E-Waved-40-75-SSTR1	4680.0	5280.0	0.7	4800.0	5280.0	1797.6	1800.0	0.0	1.6	0.0	1800.0	0.0	33.7
E-Waved-40-75-SSTR2	3921.4	4260.0	0.9	3960.0	4260.0	1797.4	1800.0	18.3	1.7	9.9	1800.0	1.4	36.3
E-Waved-40-75-SSTR3	3441.4	3720.0	0.6	3480.0	3720.0	1797.5	1800.0	9.7	1.9	1.6	1800.0	0.0	40.7
E-Waved-40-75-SSTR4	3480.0	3840.0	5.7	3540.0	3840.0	1792.3	1800.0	12.5	2.0	0.0	1800.0	0.0	50.1
E-Waved-40-75-SSTR5	3180.0	3300.0	329.2	3240.0	3300.0	1468.8	1800.0	30.9	2.0	12.7	1800.0	0.0	360.0
E-Waved-40-75-SSTR6	2945.0	3060.0	26.2	3000.0	3060.0	1771.7	1800.0	13.7	2.1	11.8	1800.0	0.0	114.8
E-Waved-40-105-SSTR1	7260.0	7260.0	0.6	7260.0	7260.0	1.3	3.2	19.0	1.3	6.6	1800.0	0.0	22.6
E-Waved-40-105-SSTR2	6180.0	6180.0	11.8	6180.0	6180.0	1.3	14.4	16.5	1.3	6.8	1800.0	0.0	60.5
E-Waved-40-105-SSTR3	5760.0	5760.0	2.5	5760.0	5760.0	1.4	5.3	15.6	1.4	14.6	1800.0	0.0	34.3
E-Waved-40-105-SSTR4	6720.0	6720.0	0.1	6720.0	6720.0	1.3	2.8	7.1	1.3	0.0	1800.0	0.0	22.4
E-Waved-40-105-SSTR5	5760.0	5760.0	6.4	5760.0	5760.0	1.4	9.2	11.5	1.4	11.5	1800.0	0.0	36.1
E-Waved-40-105-SSTR6	5520.0	5520.0	0.5	5520.0	5520.0	1.6	3.7	7.6	1.6	7.6	1800.0	0.0	27.4

Table 8 Results for Instances with 80 batches, 5 truck departures per staging lane

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-80-45-SSTR1		9,120.0	1,800.0				1,800.0	5.3	8.1	0.0	360.0
U-Waved-80-45-SSTR2		8,520.0	1,800.0				1,800.0	6.3	12.1	-0.7	360.0
U-Waved-80-45-SSTR3	8,143.0	8,220.0	994.3	8,160.0	8,220.0	795.8	1,800.0	2.9	9.9	0.0	360.0
U-Waved-80-45-SSTR4		9,120.0	1,800.0				1,800.0	10.5	7.7	0.0	360.0
U-Waved-80-45-SSTR5		7,020.0	1,800.0				1,800.0	12.0	10.1	0.0	360.0
U-Waved-80-45-SSTR6		5,280.0	1,800.0				1,800.0	9.1	10.0	0.0	360.0
U-Waved-80-75-SSTR1	10,020.0	10,020.0	26.1	10,020.0	10,020.0	6.8	39.7	17.4	6.8	0.0	95.0
U-Waved-80-75-SSTR2	9,540.0	9,540.0	25.2	9,540.0	9,540.0	8.9	42.8	17.6	8.8	0.0	148.7
U-Waved-80-75-SSTR3	9,400.0	9,420.0	14.3	9,420.0	9,420.0	8.4	31.0	3.8	8.4	0.0	71.5
U-Waved-80-75-SSTR4	9,600.0	9,600.0	21.8	9,600.0	9,600.0	7.3	36.3	15.0	7.3	0.0	102.9
U-Waved-80-75-SSTR5	7,650.0	7,680.0	488.1	7,680.0	7,680.0	8.0	504.1	14.1	7.9	0.0	360.0
U-Waved-80-75-SSTR6	6,759.0	6,780.0	321.3	6,780.0	6,780.0	8.1	337.4	14.2	8.0	0.0	265.3
U-Waved-80-105-SSTR1	11,200.0	11,520.0	11.3	11,220.0	11,520.0	1,782.2	1,800.0	13.0	6.5	0.0	43.6
U-Waved-80-105-SSTR2	10,560.0	10,800.0	55.4	10,560.0	10,800.0	1,736.8	1,800.0	15.6	7.8	0.0	72.5
U-Waved-80-105-SSTR3	10,280.0	10,320.0	4.8	10,320.0	10,320.0	8.1	21.0	2.3	8.1	0.0	48.2
U-Waved-80-105-SSTR4	10,880.0	11,040.0	119.1	10,920.0	11,040.0	1,673.5	1,800.0	35.3	7.3	0.0	358.3
U-Waved-80-105-SSTR5	9,963.5	10,020.0	982.5	10,020.0	10,020.0	8.8	1,000.0	21.0	8.7	0.0	360.0
U-Waved-80-105-SSTR6	9,628.0	9,660.0	58.7	9,660.0	9,660.0	9.6	78.0	9.9	9.6	0.0	118.8
U-Waveless-80-45-SSTR1	5,760.0	5,760.0	2.9	5,760.0	5,760.0	6.7	16.2	9.4	6.7	0.0	71.3
U-Waveless-80-45-SSTR2	4,620.0	4,680.0	78.9	4,620.0	4,680.0	1,713.7	1,800.0	15.4	7.3	-1.3	85.9
U-Waveless-80-45-SSTR3		4,320.0	1,800.0				1,800.0	6.9	7.7	1.4	360.0
U-Waveless-80-45-SSTR4		4,980.0	1,800.0				1,800.0	39.8	7.6	-1.2	360.0
U-Waveless-80-45-SSTR5		4,620.0	1,800.0				1,800.0	16.9	8.0	-1.3	360.0
U-Waveless-80-45-SSTR6	3,915.3	4,020.0	1,628.4	3,960.0	4,020.0	161.7	1,800.0	19.4	9.8	4.5	360.0
U-Waveless-80-75-SSTR1		7,440.0	1,800.0				1,800.0	36.3	6.5	-0.8	360.0
U-Waveless-80-75-SSTR2		6,900.0	1,800.0				1,800.0	19.1	7.0	0.0	360.0
U-Waveless-80-75-SSTR3		6,720.0	1,800.0				1,800.0	17.0	7.0	0.9	360.0
U-Waveless-80-75-SSTR4		7,140.0	1,800.0				1,800.0	37.0	8.7	0.8	360.0
U-Waveless-80-75-SSTR5		6,900.0	1,800.0				1,800.0	10.4	7.4	0.0	360.0
U-Waveless-80-75-SSTR6		6,480.0	1,800.0				1,800.0	12.0	8.7	0.0	360.0
U-Waveless-80-105-SSTR1		9,840.0	1,800.0				1,800.0	26.8	6.7	0.0	360.0
U-Waveless-80-105-SSTR2	9,202.7	9,240.0	1,177.1	9,240.0	9,240.0	6.4	1,189.8	27.9	6.4	0.0	360.0
U-Waveless-80-105-SSTR3	9,030.7	9,120.0	397.5	9,060.0	9,120.0	1,395.5	1,800.0	12.5	7.1	0.0	360.0
U-Waveless-80-105-SSTR4		9,540.0	1,800.0				1,800.0	28.3	7.0	0.0	360.0
U-Waveless-80-105-SSTR5		9,180.0	1,800.0				1,800.0	19.0	7.7	0.0	360.0
U-Waveless-80-105-SSTR6	8,684.4	8,700.0	62.3	8,700.0	8,700.0	9.2	80.6	11.0	9.1	0.0	191.4
E-Waved-80-45-SSTR1		7,860.0	1,800.0				1,800.0	6.9	8.6	0.0	360.0
E-Waved-80-45-SSTR2		7,200.0	1,800.0				1,800.0	8.3	13.4	0.0	360.0
E-Waved-80-45-SSTR3	6,807.0	6,900.0	1,772.4	6,840.0	6,900.0	16.4	1,800.0	4.3	11.2	0.0	360.0
E-Waved-80-45-SSTR4		7,680.0	1,800.0				1,800.0	19.5	8.4	0.0	360.0
E-Waved-80-45-SSTR5		5,940.0	1,800.0				1,800.0	11.1	10.8	-1.0	360.0
E-Waved-80-45-SSTR6		4,620.0	1,800.0				1,800.0	13.0	13.0	0.0	360.0
E-Waved-80-75-SSTR1	12,131.3	12,180.0	31.6	12,180.0	12,180.0	6.5	44.7	3.9	6.5	0.0	88.5
E-Waved-80-75-SSTR2	10,487.9	10,740.0	20.7	10,500.0	10,740.0	1,771.8	1,800.0	3.9	7.5	0.0	78.9
E-Waved-80-75-SSTR3	10,030.0	10,080.0	8.6	10,080.0	10,080.0	8.7	26.0	2.4	8.7	0.0	53.4
E-Waved-80-75-SSTR4		9,660.0	1,800.0				1,800.0	26.1	7.1	0.0	166.0
E-Waved-80-75-SSTR5		8,520.0	1,800.0				1,800.0	19.0	7.9	0.0	360.0
E-Waved-80-75-SSTR6		8,160.0	1,800.0				1,800.0	5.9	8.5	0.7	360.0
E-Waved-80-105-SSTR1		12,600.0	1,800.0				1,800.0	2.9	7.5	0.0	360.0
E-Waved-80-105-SSTR2		10,740.0	1,800.0				1,800.0	1.7	8.1	0.6	360.0
E-Waved-80-105-SSTR3		10,020.0	1,800.0				1,800.0	0.0	9.0	0.0	360.0
E-Waved-80-105-SSTR4		9,600.0	1,800.0				1,800.0	30.6	7.2	0.0	360.0
E-Waved-80-105-SSTR5		8,040.0	1,800.0				1,800.0	17.2	9.0	-0.8	360.0
E-Waved-80-105-SSTR6		7,740.0	1,800.0				1,800.0	11.6	10.3	0.8	360.0
E-Waveless-80-45-SSTR1		4,800.0	1,800.0				1,800.0	30.0	6.9	0.0	360.0
E-Waveless-80-45-SSTR2		4,440.0	1,800.0				1,800.0	10.8	7.8	-4.2	360.0
E-Waveless-80-45-SSTR3		4,200.0	1,800.0				1,800.0	5.7	8.3	-1.4	360.0
E-Waveless-80-45-SSTR4		4,800.0	1,800.0				1,800.0	30.0	8.3	-5.3	360.0
E-Waveless-80-45-SSTR5		4,440.0	1,800.0				1,800.0	13.5	9.5	0.0	360.0
E-Waveless-80-45-SSTR6		4,020.0	1,800.0				1,800.0	10.4	12.4	0.0	360.0
E-Waveless-80-75-SSTR1	8,616.2	8,880.0	1,577.7	8,640.0	8,880.0	216.5	1,800.0	18.9	5.8	-0.7	360.0
E-Waveless-80-75-SSTR2		8,100.0	1,800.0				1,800.0	11.1	6.6	1.5	360.0
E-Waveless-80-75-SSTR3		7,800.0	1,800.0				1,800.0	6.2	7.1	0.0	360.0
E-Waveless-80-75-SSTR4		8,280.0	1,800.0				1,800.0	17.4	7.3	0.0	360.0
E-Waveless-80-75-SSTR5		8,100.0	1,800.0				1,800.0	11.1	7.2	0.7	360.0
E-Waveless-80-75-SSTR6		7,620.0	1,800.0				1,800.0	11.8	9.3	0.0	360.0
E-Waveless-80-105-SSTR1		8,460.0	1,800.0				1,800.0	14.2	6.0	0.0	360.0
E-Waveless-80-105-SSTR2	7,800.0	7,860.0	1,375.1	7,800.0	7,860.0	418.2	1,800.0	14.5	6.7	0.8	360.0
E-Waveless-80-105-SSTR3		7,560.0	1,800.0				1,800.0	12.7	8.8	1.6	360.0
E-Waveless-80-105-SSTR4	8,040.0	8,220.0	53.8	8,040.0	8,160.0	1,738.7	1,800.0	25.7	7.5	0.7	109.6
E-Waveless-80-105-SSTR5	7,800.0	7,860.0	1,273.0	7,800.0	7,860.0	519.0	1,800.0	9.9	8.1	1.5	360.0
E-Waveless-80-105-SSTR6		7,440.0	1,800.0				1,800.0	11.3	9.5	0.0	360.0

Table 9 Results for Instances with 80 batches, 10 truck departures per staging lane

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-80-45-SSTR1	9,600.0	9,600.0	2.1	9,600.0	9,600.0	7.1	16.2	5.0	7.0	0.0	43.0
U-Waved-80-45-SSTR2	8,280.0	8,280.0	1.7	8,280.0	8,280.0	8.4	18.5	9.4	8.4	0.0	42.3
U-Waved-80-45-SSTR3	7,320.0	7,320.0	1.6	7,320.0	7,320.0	7.8	17.2	10.7	7.8	0.0	45.3
U-Waved-80-45-SSTR4	5,760.0	5,760.0	201.4	5,760.0	5,760.0	7.5	216.4	33.3	7.4	1.0	118.0
U-Waved-80-45-SSTR5	5,505.0	5,760.0	974.9	5,520.0	5,760.0	816.9	1,800.0	16.7	8.3	0.0	360.0
U-Waved-80-45-SSTR6	5,314.3	5,460.0	72.2	5,340.0	5,460.0	1,718.7	1,800.0	5.5	9.0	0.0	203.1
U-Waved-80-75-SSTR1	11,880.0	11,880.0	43.2	11,880.0	11,880.0	5.7	54.5	9.6	5.6	0.0	45.9
U-Waved-80-75-SSTR2	10,320.0	10,320.0	63.6	10,320.0	10,320.0	6.5	76.6	12.8	6.5	0.0	93.9
U-Waved-80-75-SSTR3	9,360.0	9,360.0	941.4	9,360.0	9,360.0	6.1	953.5	10.9	6.0	0.0	341.7
U-Waved-80-75-SSTR4	7,740.0	7,740.0	48.4	7,740.0	7,740.0	6.5	61.4	43.4	6.4	0.0	145.8
U-Waved-80-75-SSTR5	7,440.0	7,440.0	164.5	7,440.0	7,440.0	6.9	178.2	23.4	6.8	0.8	148.3
U-Waved-80-75-SSTR6	7,120.0	7,200.0	30.4	7,140.0	7,200.0	1,762.0	1,800.0	14.2	7.6	-0.8	76.4
U-Waved-80-105-SSTR1	15,360.0	15,360.0	0.9	15,360.0	15,360.0	5.0	10.8	10.2	5.0	0.0	28.8
U-Waved-80-105-SSTR2	13,860.0	13,860.0	0.9	13,860.0	13,860.0	5.6	12.1	5.2	5.6	0.0	30.1
U-Waved-80-105-SSTR3	12,900.0	12,900.0	0.7	12,900.0	12,900.0	5.7	12.1	0.9	5.7	0.0	30.0
U-Waved-80-105-SSTR4	11,520.0	11,580.0	2.5	11,520.0	11,580.0	1,791.9	1,800.0	25.9	5.5	-0.5	31.3
U-Waved-80-105-SSTR5	10,275.0	10,320.0	6.2	10,320.0	10,320.0	6.1	18.3	16.9	6.0	0.0	41.9
U-Waved-80-105-SSTR6	9,960.0	9,960.0	2.3	9,960.0	9,960.0	6.8	15.9	1.2	6.8	0.6	40.9
U-Waveless-80-45-SSTR1	6,240.0	6,240.0	3.5	6,240.0	6,240.0	6.1	15.6	7.7	6.1	0.0	54.4
U-Waveless-80-45-SSTR2	5,400.0	5,400.0	5.4	5,400.0	5,400.0	6.6	18.4	20.0	6.5	0.0	56.9
U-Waveless-80-45-SSTR3		5,280.0	1,800.0				1,800.0	4.5	6.8	0.0	360.0
U-Waveless-80-45-SSTR4	5,280.0	5,460.0	963.0	5,280.0	5,460.0	830.6	1,800.0	31.9	6.4	-2.2	179.9
U-Waveless-80-45-SSTR5		4,800.0	1,800.0				1,800.0	13.8	6.8	0.0	360.0
U-Waveless-80-45-SSTR6	4,513.3	4,800.0	160.8	4,560.0	4,800.0	1,630.9	1,800.0	7.5	8.3	-2.5	349.7
U-Waveless-80-75-SSTR1	8,800.0	9,120.0	274.7	8,820.0	9,120.0	1,520.1	1,800.0	15.8	5.2	0.0	360.0
U-Waveless-80-75-SSTR2		8,040.0	1,800.0				1,800.0	20.9	5.7	0.7	360.0
U-Waveless-80-75-SSTR3		7,980.0	1,800.0				1,800.0	11.3	6.4	0.0	360.0
U-Waveless-80-75-SSTR4		7,800.0	1,800.0				1,800.0	36.2	6.2	0.8	360.0
U-Waveless-80-75-SSTR5		7,020.0	1,800.0				1,800.0	19.7	6.0	0.8	360.0
U-Waveless-80-75-SSTR6	6,775.6	6,900.0	86.1	6,780.0	6,900.0	1,707.2	1,800.0	15.7	6.7	0.0	194.1
U-Waveless-80-105-SSTR1	11,026.7	11,160.0	3.1	11,040.0	11,160.0	1,791.9	1,800.0	30.1	5.0	0.0	34.1
U-Waveless-80-105-SSTR2	10,110.0	10,140.0	19.9	10,140.0	10,140.0	5.3	30.5	19.5	5.3	0.0	58.7
U-Waveless-80-105-SSTR3	10,035.0	10,080.0	17.1	10,080.0	10,080.0	5.4	27.8	11.9	5.4	0.0	66.7
U-Waveless-80-105-SSTR4	10,145.8	10,200.0	909.2	10,200.0	10,200.0	5.7	920.5	39.4	5.6	0.0	360.0
U-Waveless-80-105-SSTR5	9,564.0	9,600.0	395.9	9,600.0	9,600.0	6.0	407.8	21.9	6.0	0.0	360.0
U-Waveless-80-105-SSTR6	9,269.0	9,360.0	9.0	9,300.0	9,360.0	1,784.5	1,800.0	10.9	6.5	0.0	48.5
E-Waved-80-45-SSTR1	7,356.0	7,680.0	321.7	7,380.0	7,680.0	1,471.7	1,800.0	7.0	6.6	0.0	85.0
E-Waved-80-45-SSTR2	6,354.0	6,660.0	678.2	6,360.0	6,660.0	1,114.2	1,800.0	12.6	7.6	0.9	360.0
E-Waved-80-45-SSTR3	5,814.0	6,060.0	1,091.9	5,820.0	6,060.0	700.7	1,800.0	13.9	7.4	1.0	199.8
E-Waved-80-45-SSTR4		5,580.0	1,800.0				1,800.0	32.3	7.0	4.1	360.0
E-Waved-80-45-SSTR5		5,460.0	1,800.0				1,800.0	15.4	8.1	0.0	360.0
E-Waved-80-45-SSTR6		5,160.0	1,800.0				1,800.0	18.6	8.8	1.1	360.0
E-Waved-80-75-SSTR1	10,140.0	10,140.0	3.1	10,140.0	10,140.0	6.2	15.5	13.6	6.2	0.0	47.3
E-Waved-80-75-SSTR2	8,610.0	8,640.0	10.0	8,640.0	8,640.0	6.7	23.2	17.4	6.6	0.0	360.0
E-Waved-80-75-SSTR3	7,830.0	7,860.0	2.2	7,860.0	7,860.0	6.5	15.2	5.3	6.5	0.0	52.6
E-Waved-80-75-SSTR4		6,960.0	1,800.0				1,800.0	37.9	6.5	0.0	84.3
E-Waved-80-75-SSTR5		6,600.0	1,800.0				1,800.0	29.1	6.7	1.8	360.0
E-Waved-80-75-SSTR6		6,480.0	1,800.0				1,800.0	14.8	7.6	-0.9	360.0
E-Waved-80-105-SSTR1	15,900.0	15,900.0	0.7	15,900.0	15,900.0	4.4	9.4	3.4	4.3	0.0	25.4
E-Waved-80-105-SSTR2	13,380.0	13,380.0	1.5	13,380.0	13,380.0	5.0	11.5	8.1	5.0	0.0	28.1
E-Waved-80-105-SSTR3	12,480.0	12,480.0	1.3	12,480.0	12,480.0	5.2	11.7	3.8	5.1	0.0	30.1
E-Waved-80-105-SSTR4	12,480.0	12,480.0	4.3	12,480.0	12,480.0	4.9	14.0	15.4	4.8	0.0	38.1
E-Waved-80-105-SSTR5	11,328.0	11,340.0	432.8	11,340.0	11,340.0	5.5	443.7	14.8	5.4	0.0	360.0
E-Waved-80-105-SSTR6	11,218.2	11,280.0	314.9	11,220.0	11,280.0	1,478.6	1,800.0	5.9	6.5	-0.5	206.6
E-Waveless-80-45-SSTR1	6,720.0	6,720.0	526.7	6,720.0	6,720.0	6.7	540.0	14.3	6.6	0.0	81.0
E-Waveless-80-45-SSTR2	5,660.0	5,700.0	953.3	5,700.0	5,700.0	6.2	965.6	11.6	6.1	0.0	118.7
E-Waveless-80-45-SSTR3		5,160.0	1,800.0				1,800.0	14.0	6.7	1.1	360.0
E-Waveless-80-45-SSTR4	6,080.0	6,240.0	15.9	6,120.0	6,240.0	1,777.7	1,800.0	8.7	6.4	0.0	151.3
E-Waveless-80-45-SSTR5		5,340.0	1,800.0				1,800.0	18.0	7.2	0.0	360.0
E-Waveless-80-45-SSTR6		4,980.0	1,800.0				1,800.0	18.1	7.7	0.0	360.0
E-Waveless-80-75-SSTR1		7,680.0	1,800.0				1,800.0	37.5	6.3	0.0	360.0
E-Waveless-80-75-SSTR2		6,660.0	1,800.0				1,800.0	34.2	6.2	1.8	360.0
E-Waveless-80-75-SSTR3		6,540.0	1,800.0				1,800.0	14.7	6.2	0.0	360.0
E-Waveless-80-75-SSTR4		7,200.0	1,800.0				1,800.0	40.0	6.3	0.8	360.0
E-Waveless-80-75-SSTR5		6,360.0	1,800.0				1,800.0	21.7	6.4	0.9	360.0
E-Waveless-80-75-SSTR6		6,180.0	1,800.0				1,800.0	20.4	7.2	0.0	360.0
E-Waveless-80-105-SSTR1	12,930.0	13,140.0	1.1	12,960.0	13,140.0	1,794.7	1,800.0	6.4	4.2	0.0	28.0
E-Waveless-80-105-SSTR2	11,030.0	11,160.0	3.6	11,040.0	11,160.0	1,791.7	1,800.0	8.1	4.7	0.0	36.8
E-Waveless-80-105-SSTR3	10,537.8	10,680.0	626.9	10,560.0	10,680.0	1,168.0	1,800.0	9.0	5.1	0.0	360.0
E-Waveless-80-105-SSTR4	12,060.0	12,060.0	5.2	12,060.0	12,060.0	4.9	14.9	20.4	4.8	0.0	39.6
E-Waveless-80-105-SSTR5		10,680.0	1,800.0				1,800.0	8.4	5.2	0.0	360.0
E-Waveless-80-105-SSTR6	10,342.4	10,380.0	226.7	10,380.0	10,380.0	5.8	238.3	7.5	5.8	0.0	360.0

Table 10 **Results for Instances with 80 batches, 20 truck departures per staging lane**

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-80-45-SSTR1	8,160.0	8,160.0	1.2	8,160.0	8,160.0	4.9	10.9	5.9	4.9	0.0	38.9
U-Waved-80-45-SSTR2	7,500.0	7,500.0	0.4	7,500.0	7,500.0	5.6	11.6	0.0	5.6	0.0	41.1
U-Waved-80-45-SSTR3	7,140.0	7,140.0	1.0	7,140.0	7,140.0	5.3	11.6	3.4	5.3	0.0	39.3
U-Waved-80-45-SSTR4	6,720.0	6,720.0	12.0	6,720.0	6,720.0	5.8	23.5	28.6	5.7	0.0	43.7
U-Waved-80-45-SSTR5	6,160.0	6,240.0	10.4	6,180.0	6,240.0	1,783.7	1,800.0	16.3	5.9	0.0	64.8
U-Waved-80-45-SSTR6	6,020.0	6,180.0	1.5	6,060.0	6,180.0	1,792.4	1,800.0	1.0	6.2	-1.0	41.4
U-Waved-80-75-SSTR1	10,320.0	10,320.0	1.6	10,320.0	10,320.0	4.3	10.1	12.2	4.2	0.0	26.0
U-Waved-80-75-SSTR2	9,600.0	9,600.0	1.0	9,600.0	9,600.0	4.6	10.2	10.6	4.6	0.0	30.3
U-Waved-80-75-SSTR3	9,120.0	9,120.0	1.4	9,120.0	9,120.0	4.5	10.4	7.2	4.5	0.0	28.8
U-Waved-80-75-SSTR4	8,160.0	8,220.0	87.6	8,160.0	8,220.0	1,707.8	1,800.0	35.8	4.6	0.0	204.4
U-Waved-80-75-SSTR5		7,740.0	1,800.0				1,800.0	21.7	5.0	1.5	360.0
U-Waved-80-75-SSTR6	7,625.0	7,680.0	20.0	7,680.0	7,680.0	5.5	30.9	16.4	5.4	0.0	92.1
U-Waved-80-105-SSTR1	13,440.0	13,440.0	0.4	13,440.0	13,440.0	3.8	7.9	8.0	3.8	0.0	25.8
U-Waved-80-105-SSTR2	12,240.0	12,240.0	0.9	12,240.0	12,240.0	4.3	9.5	13.7	4.3	0.0	27.8
U-Waved-80-105-SSTR3	12,240.0	12,240.0	0.3	12,240.0	12,240.0	4.3	8.9	4.4	4.3	0.0	26.1
U-Waved-80-105-SSTR4	11,760.0	11,760.0	1.3	11,760.0	11,760.0	4.2	9.6	28.6	4.1	0.0	28.3
U-Waved-80-105-SSTR5	10,725.0	10,740.0	3.8	10,740.0	10,740.0	4.5	12.8	12.3	4.5	0.0	39.1
U-Waved-80-105-SSTR6	10,260.0	10,260.0	2.3	10,260.0	10,260.0	5.1	12.5	7.0	5.1	0.0	35.0
E-Waved-80-45-SSTR1	6,240.0	6,240.0	4.9	6,240.0	6,240.0	5.1	15.0	23.1	5.0	0.0	49.9
E-Waved-80-45-SSTR2	5,700.0	5,700.0	3.7	5,700.0	5,700.0	5.6	14.9	6.3	5.6	0.0	58.2
E-Waved-80-45-SSTR3	5,400.0	5,400.0	2.8	5,400.0	5,400.0	5.4	13.5	18.9	5.3	0.0	59.3
E-Waved-80-45-SSTR4	5,208.0	5,340.0	47.4	5,220.0	5,340.0	1,746.9	1,800.0	34.8	5.7	0.0	168.8
E-Waved-80-45-SSTR5	5,025.0	5,160.0	239.8	5,040.0	5,160.0	1,554.2	1,800.0	14.0	6.1	0.0	360.0
E-Waved-80-45-SSTR6	4,980.0	4,980.0	223.7	4,980.0	4,980.0	6.7	237.1	15.7	6.7	0.0	360.0
E-Waved-80-75-SSTR1	9,840.0	10,080.0	4.0	9,840.0	10,080.0	1,791.9	1,800.0	14.3	4.1	0.0	32.1
E-Waved-80-75-SSTR2	8,890.0	9,120.0	36.9	8,940.0	9,120.0	1,758.8	1,800.0	4.6	4.3	0.0	55.6
E-Waved-80-75-SSTR3	8,320.0	8,520.0	40.6	8,340.0	8,520.0	1,755.1	1,800.0	7.0	4.2	0.0	88.5
E-Waved-80-75-SSTR4	9,120.0	9,180.0	5.3	9,120.0	9,180.0	1,790.3	1,800.0	20.9	4.5	-0.7	40.9
E-Waved-80-75-SSTR5		8,400.0	1,800.0				1,800.0	19.3	4.8	0.0	360.0
E-Waved-80-75-SSTR6	8,007.9	8,100.0	631.1	8,040.0	8,100.0	1,163.6	1,800.0	8.9	5.3	0.0	360.0
E-Waved-80-105-SSTR1		9,600.0	1,800.0				1,800.0	15.0	4.1	0.0	360.0
E-Waved-80-105-SSTR2		9,000.0	1,800.0				1,800.0	8.0	4.4	0.0	360.0
E-Waved-80-105-SSTR3		8,880.0	1,800.0				1,800.0	8.1	4.5	0.0	360.0
E-Waved-80-105-SSTR4		8,340.0	1,800.0				1,800.0	43.9	4.6	0.0	360.0
E-Waved-80-105-SSTR5		8,100.0	1,800.0				1,800.0	23.7	4.8	0.0	360.0
E-Waved-80-105-SSTR6	7,900.6	7,920.0	9.4	7,920.0	7,920.0	5.5	20.3	11.4	5.4	0.0	52.8

Table 11 Results for Instances with 160 batches, 5 truck departures per staging lane

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-160-45-SSTR1		17,760.0	1,800.0				1,800.0	0.3	29.3	0.0	360.0
U-Waved-160-45-SSTR2		16,620.0	1,800.0				1,800.0	7.9	48.9	0.0	360.0
U-Waved-160-45-SSTR3		16,380.0	1,800.0				1,800.0	2.9	41.8	-0.7	360.0
U-Waved-160-45-SSTR4		17,760.0	1,800.0				1,800.0	8.8	30.6	0.0	360.0
U-Waved-160-45-SSTR5		13,920.0	1,800.0				1,800.0	7.3	36.3	4.9	360.0
U-Waved-160-45-SSTR6		10,620.0	1,800.0				1,800.0	6.8	41.9	0.6	360.0
U-Waved-160-75-SSTR1		20,220.0	1,800.0				1,800.0	15.7	28.3	0.0	360.0
U-Waved-160-75-SSTR2		19,260.0	1,800.0				1,800.0	17.1	36.7	0.0	360.0
U-Waved-160-75-SSTR3		18,900.0	1,800.0				1,800.0	3.5	33.0	0.0	360.0
U-Waved-160-75-SSTR4		19,200.0	1,800.0				1,800.0	14.4	29.4	0.0	360.0
U-Waved-160-75-SSTR5		15,600.0	1,800.0				1,800.0	11.2	32.1	-1.2	360.0
U-Waved-160-75-SSTR6		13,740.0	1,800.0				1,800.0	7.9	32.7	0.4	360.0
U-Waved-160-105-SSTR1	22,400.0	22,560.0	406.3	22,440.0	22,560.0	1,367.8	1,800.0	23.1	25.9	0.0	360.0
U-Waved-160-105-SSTR2	21,120.0	21,240.0	700.8	21,120.0	21,240.0	1,068.6	1,800.0	22.6	30.6	2.8	360.0
U-Waved-160-105-SSTR3	20,520.0	20,520.0	988.9	20,520.0	20,520.0	31.1	1,051.0	6.4	31.0	0.6	360.0
U-Waved-160-105-SSTR4		21,840.0	1,800.0				1,800.0	30.2	29.2	2.2	360.0
U-Waved-160-105-SSTR5		20,220.0	1,800.0				1,800.0	18.4	32.6	1.2	360.0
U-Waved-160-105-SSTR6		19,320.0	1,800.0				1,800.0	12.4	37.3	1.2	360.0
U-Waveless-160-45-SSTR1		9,240.0	1,800.0				1,800.0	19.5	24.2	3.8	360.0
U-Waveless-160-45-SSTR2		8,400.0	1,800.0				1,800.0	12.9	26.3	4.1	360.0
U-Waveless-160-45-SSTR3		7,680.0	1,800.0				1,800.0	10.2	29.6	3.0	360.0
U-Waveless-160-45-SSTR4		9,240.0	1,800.0				1,800.0	25.3	26.6	1.3	360.0
U-Waveless-160-45-SSTR5		8,280.0	1,800.0				1,800.0	11.6	28.9	1.4	360.0
U-Waveless-160-45-SSTR6		7,620.0	1,800.0				1,800.0	7.9	34.4	3.1	360.0
U-Waveless-160-75-SSTR1		15,600.0	1,800.0				1,800.0	23.5	22.6	0.4	360.0
U-Waveless-160-75-SSTR2		13,560.0	1,800.0				1,800.0	19.0	23.5	3.4	360.0
U-Waveless-160-75-SSTR3		12,600.0	1,800.0				1,800.0	12.4	26.1	1.9	360.0
U-Waveless-160-75-SSTR4		15,540.0	1,800.0				1,800.0	27.0	24.9	0.4	360.0
U-Waveless-160-75-SSTR5		13,860.0	1,800.0				1,800.0	9.5	25.9	2.9	360.0
U-Waveless-160-75-SSTR6		12,480.0	1,800.0				1,800.0	14.9	29.4	1.9	360.0
U-Waveless-160-105-SSTR1	19,960.0	20,160.0	1,271.9	19,980.0	20,160.0	506.6	1,800.0	24.4	21.6	0.3	360.0
U-Waveless-160-105-SSTR2		18,480.0	1,800.0				1,800.0	27.3	23.6	1.0	360.0
U-Waveless-160-105-SSTR3		17,280.0	1,800.0				1,800.0	15.3	25.1	0.3	360.0
U-Waveless-160-105-SSTR4		19,740.0	1,800.0				1,800.0	25.5	23.8	3.5	360.0
U-Waveless-160-105-SSTR5		18,480.0	1,800.0				1,800.0	17.2	25.5	1.9	360.0
U-Waveless-160-105-SSTR6		17,160.0	1,800.0				1,800.0	10.5	30.2	0.3	360.0
E-Waved-160-45-SSTR1		13,260.0	1,800.0				1,800.0	13.1	32.6	-3.3	360.0
E-Waved-160-45-SSTR2		12,060.0	1,800.0				1,800.0	8.0	49.9	0.0	360.0
E-Waved-160-45-SSTR3		11,700.0	1,800.0				1,800.0	3.1	52.4	0.0	360.0
E-Waved-160-45-SSTR4		12,480.0	1,800.0				1,800.0	24.5	36.2	0.0	360.0
E-Waved-160-45-SSTR5		10,140.0	1,800.0				1,800.0	20.1	55.2	2.3	360.0
E-Waved-160-45-SSTR6		8,280.0	1,800.0				1,800.0	13.8	65.7	1.4	360.0
E-Waved-160-75-SSTR1		19,920.0	1,800.0				1,800.0	19.0	28.4	0.0	360.0
E-Waved-160-75-SSTR2		17,760.0	1,800.0				1,800.0	8.8	34.1	0.3	360.0
E-Waved-160-75-SSTR3		17,100.0	1,800.0				1,800.0	1.8	32.1	0.0	360.0
E-Waved-160-75-SSTR4		16,980.0	1,800.0				1,800.0	34.3	29.7	0.0	360.0
E-Waved-160-75-SSTR5		14,820.0	1,800.0				1,800.0	19.0	33.8	3.5	360.0
E-Waved-160-75-SSTR6		14,280.0	1,800.0				1,800.0	9.2	35.2	2.1	360.0
E-Waved-160-105-SSTR1		20,520.0	1,800.0				1,800.0	18.4	25.6	0.0	360.0
E-Waved-160-105-SSTR2		18,780.0	1,800.0				1,800.0	16.0	29.2	0.6	360.0
E-Waved-160-105-SSTR3		18,360.0	1,800.0				1,800.0	3.3	30.6	0.3	360.0
E-Waved-160-105-SSTR4		18,180.0	1,800.0				1,800.0	35.3	28.7	1.6	360.0
E-Waved-160-105-SSTR5		17,220.0	1,800.0				1,800.0	13.2	32.9	0.7	360.0
E-Waved-160-105-SSTR6		16,200.0	1,800.0				1,800.0	8.5	38.2	1.1	360.0
E-Waveless-160-45-SSTR1		10,140.0	1,800.0				1,800.0	18.3	26.8	0.0	360.0
E-Waveless-160-45-SSTR2		8,820.0	1,800.0				1,800.0	7.5	29.4	-0.7	360.0
E-Waveless-160-45-SSTR3		7,920.0	1,800.0				1,800.0	7.6	36.8	1.5	360.0
E-Waveless-160-45-SSTR4		9,840.0	1,800.0				1,800.0	17.7	31.3	1.2	360.0
E-Waveless-160-45-SSTR5		8,640.0	1,800.0				1,800.0	7.6	30.8	1.4	360.0
E-Waveless-160-45-SSTR6		7,740.0	1,800.0				1,800.0	8.5	41.7	2.3	360.0
E-Waveless-160-75-SSTR1		15,360.0	1,800.0				1,800.0	18.8	24.0	0.4	360.0
E-Waveless-160-75-SSTR2		14,220.0	1,800.0				1,800.0	11.4	24.6	1.3	360.0
E-Waveless-160-75-SSTR3		13,380.0	1,800.0				1,800.0	9.9	27.3	1.8	360.0
E-Waveless-160-75-SSTR4		14,760.0	1,800.0				1,800.0	21.1	25.8	3.5	360.0
E-Waveless-160-75-SSTR5		14,280.0	1,800.0				1,800.0	11.3	27.8	2.5	360.0
E-Waveless-160-75-SSTR6		13,320.0	1,800.0				1,800.0	7.7	31.4	1.8	360.0
E-Waveless-160-105-SSTR1		19,320.0	1,800.0				1,800.0	22.0	21.6	0.9	360.0
E-Waveless-160-105-SSTR2		16,980.0	1,800.0				1,800.0	15.9	22.2	1.0	360.0
E-Waveless-160-105-SSTR3		15,960.0	1,800.0				1,800.0	9.0	26.0	0.7	360.0
E-Waveless-160-105-SSTR4		18,660.0	1,800.0				1,800.0	19.3	23.7	-0.6	360.0
E-Waveless-160-105-SSTR5		16,860.0	1,800.0				1,800.0	12.1	26.5	1.1	360.0
E-Waveless-160-105-SSTR6		15,720.0	1,800.0				1,800.0	7.3	32.2	0.8	360.0

Table 12 **Results for Instances with 160 batches, 10 truck departures per staging lane**

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-160-45-SSTR1	19,200.0	19,200.0	38.2	19,200.0	19,200.0	26.8	91.7	5.0	26.7	0.0	134.1
U-Waved-160-45-SSTR2	16,410.0	16,440.0	44.1	16,440.0	16,440.0	30.8	105.6	7.3	30.7	0.0	143.1
U-Waved-160-45-SSTR3	14,490.0	14,520.0	21.8	14,520.0	14,520.0	29.0	79.6	8.3	28.9	0.0	108.7
U-Waved-160-45-SSTR4		11,700.0	1,800.0				1,800.0	40.5	27.7	3.5	360.0
U-Waved-160-45-SSTR5		11,760.0	1,800.0				1,800.0	18.4	31.5	-0.5	360.0
U-Waved-160-45-SSTR6		10,800.0	1,800.0				1,800.0	5.0	36.7	0.6	360.0
U-Waved-160-75-SSTR1	24,480.0	24,480.0	7.7	24,480.0	24,480.0	22.3	52.2	4.2	22.2	0.0	78.1
U-Waved-160-75-SSTR2	21,510.0	21,540.0	39.1	21,540.0	21,540.0	23.2	85.3	6.7	23.1	0.0	97.4
U-Waved-160-75-SSTR3	19,590.0	19,620.0	11.7	19,620.0	19,620.0	25.8	63.3	5.2	25.8	0.0	102.6
U-Waved-160-75-SSTR4	16,800.0	16,800.0	333.4	16,800.0	16,800.0	26.1	385.3	20.4	25.8	0.0	263.1
U-Waved-160-75-SSTR5		15,120.0	1,800.0				1,800.0	21.0	26.7	0.0	360.0
U-Waved-160-75-SSTR6	14,080.0	14,220.0	218.0	14,100.0	14,220.0	1,552.0	1,800.0	18.1	30.0	-0.4	360.0
U-Waved-160-105-SSTR1	30,720.0	30,720.0	7.6	30,720.0	30,720.0	19.8	47.2	2.0	19.8	0.0	62.9
U-Waved-160-105-SSTR2	27,720.0	27,720.0	9.4	27,720.0	27,720.0	21.4	52.2	2.2	21.3	0.0	80.8
U-Waved-160-105-SSTR3	25,800.0	25,800.0	11.3	25,800.0	25,800.0	21.6	54.5	0.5	21.5	0.0	80.1
U-Waved-160-105-SSTR4	23,040.0	23,100.0	39.1	23,040.0	23,100.0	1,739.7	1,800.0	17.4	21.2	-0.3	97.0
U-Waved-160-105-SSTR5	20,340.0	20,340.0	469.0	20,340.0	20,340.0	23.4	515.5	20.6	23.2	0.0	360.0
U-Waved-160-105-SSTR6	19,890.0	19,980.0	706.2	19,920.0	19,980.0	1,067.8	1,800.0	0.9	26.1	0.0	360.0
U-Waveless-160-45-SSTR1		12,720.0	1,800.0				1,800.0	13.2	21.8	-0.5	360.0
U-Waveless-160-45-SSTR2		11,160.0	1,800.0				1,800.0	9.7	23.0	-3.3	360.0
U-Waveless-160-45-SSTR3		9,660.0	1,800.0				1,800.0	8.7	26.7	1.2	360.0
U-Waveless-160-45-SSTR4		11,820.0	1,800.0				1,800.0	21.8	24.1	3.0	360.0
U-Waveless-160-45-SSTR5		10,200.0	1,800.0				1,800.0	20.6	25.6	1.7	360.0
U-Waveless-160-45-SSTR6		9,180.0	1,800.0				1,800.0	12.4	31.6	1.3	360.0
U-Waveless-160-75-SSTR1		16,980.0	1,800.0				1,800.0	21.6	19.3	0.4	360.0
U-Waveless-160-75-SSTR2		14,280.0	1,800.0				1,800.0	14.3	19.7	0.4	360.0
U-Waveless-160-75-SSTR3		13,440.0	1,800.0				1,800.0	21.0	21.4	0.4	360.0
U-Waveless-160-75-SSTR4		17,040.0	1,800.0				1,800.0	24.3	20.2	0.0	360.0
U-Waveless-160-75-SSTR5		14,220.0	1,800.0				1,800.0	14.8	21.8	0.8	360.0
U-Waveless-160-75-SSTR6		13,200.0	1,800.0				1,800.0	18.6	24.5	0.5	360.0
U-Waveless-160-105-SSTR1	23,640.0	23,640.0	33.1	23,640.0	23,640.0	16.4	65.9	27.9	16.3	0.0	112.2
U-Waveless-160-105-SSTR2	19,440.0	19,440.0	202.5	19,440.0	19,440.0	16.8	236.0	25.3	16.7	0.0	315.1
U-Waveless-160-105-SSTR3	18,225.8	18,300.0	1,308.0	18,240.0	18,300.0	473.8	1,800.0	15.7	18.2	0.0	360.0
U-Waveless-160-105-SSTR4	23,640.0	23,640.0	82.5	23,640.0	23,640.0	19.3	120.9	13.2	19.2	0.0	186.9
U-Waveless-160-105-SSTR5	19,440.0	19,440.0	933.2	19,440.0	19,440.0	19.9	972.8	22.8	19.7	0.0	360.0
U-Waveless-160-105-SSTR6	18,204.3	18,240.0	1,208.9	18,240.0	18,240.0	23.8	1,256.5	14.5	23.7	0.0	360.0
E-Waved-160-45-SSTR1	15,300.0	15,420.0	530.2	15,300.0	15,420.0	1,243.7	1,800.0	12.8	26.1	-0.4	360.0
E-Waved-160-45-SSTR2		13,320.0	1,800.0				1,800.0	18.0	31.0	0.0	360.0
E-Waved-160-45-SSTR3		12,120.0	1,800.0				1,800.0	19.3	30.4	-0.5	360.0
E-Waved-160-45-SSTR4		11,880.0	1,800.0				1,800.0	27.8	27.3	-0.5	360.0
E-Waved-160-45-SSTR5		11,460.0	1,800.0				1,800.0	13.1	30.8	0.5	360.0
E-Waved-160-45-SSTR6		10,260.0	1,800.0				1,800.0	9.9	36.9	1.2	360.0
E-Waved-160-75-SSTR1		16,560.0	1,800.0				1,800.0	19.2	23.2	1.1	360.0
E-Waved-160-75-SSTR2		14,820.0	1,800.0				1,800.0	18.2	25.1	1.2	360.0
E-Waved-160-75-SSTR3		13,800.0	1,800.0				1,800.0	19.6	25.3	0.4	360.0
E-Waved-160-75-SSTR4		13,260.0	1,800.0				1,800.0	31.7	26.6	5.6	360.0
E-Waved-160-75-SSTR5		13,020.0	1,800.0				1,800.0	18.4	27.5	2.7	360.0
E-Waved-160-75-SSTR6		12,300.0	1,800.0				1,800.0	13.2	32.8	1.4	360.0
E-Waved-160-105-SSTR1	26,520.0	26,520.0	14.1	26,520.0	26,520.0	18.3	50.6	7.2	18.2	0.0	101.0
E-Waved-160-105-SSTR2		22,620.0	1,800.0				1,800.0	13.3	19.6	-0.3	201.8
E-Waved-160-105-SSTR3		20,940.0	1,800.0				1,800.0	9.7	20.8	0.0	360.0
E-Waved-160-105-SSTR4		21,660.0	1,800.0				1,800.0	20.2	19.8	0.3	183.0
E-Waved-160-105-SSTR5		19,920.0	1,800.0				1,800.0	14.5	22.1	1.2	360.0
E-Waved-160-105-SSTR6		19,020.0	1,800.0				1,800.0	9.8	22.6	0.0	360.0
E-Waveless-160-45-SSTR1	11,180.0	11,640.0	294.6	11,220.0	11,640.0	1,485.3	1,800.0	19.6	20.1	0.0	360.0
E-Waveless-160-45-SSTR2		9,540.0	1,800.0				1,800.0	20.1	21.4	5.4	360.0
E-Waveless-160-45-SSTR3		9,060.0	1,800.0				1,800.0	14.6	24.8	1.9	360.0
E-Waveless-160-45-SSTR4		11,100.0	1,800.0				1,800.0	34.6	22.6	0.5	360.0
E-Waveless-160-45-SSTR5		9,240.0	1,800.0				1,800.0	29.2	26.6	4.9	360.0
E-Waveless-160-45-SSTR6		8,760.0	1,800.0				1,800.0	19.2	29.1	2.7	360.0
E-Waveless-160-75-SSTR1		15,000.0	1,800.0				1,800.0	25.2	19.7	0.0	360.0
E-Waveless-160-75-SSTR2		12,720.0	1,800.0				1,800.0	20.8	20.0	0.9	360.0
E-Waveless-160-75-SSTR3		12,180.0	1,800.0				1,800.0	18.7	23.3	0.5	360.0
E-Waveless-160-75-SSTR4		14,640.0	1,800.0				1,800.0	29.1	21.6	-0.4	360.0
E-Waveless-160-75-SSTR5		12,660.0	1,800.0				1,800.0	24.2	23.0	1.9	360.0
E-Waveless-160-75-SSTR6		11,760.0	1,800.0				1,800.0	15.3	26.5	0.5	360.0
E-Waveless-160-105-SSTR1		22,260.0	1,800.0				1,800.0	16.7	15.4	0.0	360.0
E-Waveless-160-105-SSTR2		18,900.0	1,800.0				1,800.0	15.9	16.4	0.6	360.0
E-Waveless-160-105-SSTR3		18,000.0	1,800.0				1,800.0	13.0	17.4	0.7	360.0
E-Waveless-160-105-SSTR4		21,360.0	1,800.0				1,800.0	19.7	17.2	0.3	360.0
E-Waveless-160-105-SSTR5		18,840.0	1,800.0				1,800.0	15.6	18.0	0.6	360.0
E-Waveless-160-105-SSTR6		17,820.0	1,800.0				1,800.0	8.4	19.8	0.7	360.0

Table 13 Results for Instances with 160 batches, 20 truck departures per staging lane

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-160-45-SSTR1	13,920.0	13,920.0	30.9	13,920.0	13,920.0	18.6	68.1	10.8	18.5	0.0	91.7
U-Waved-160-45-SSTR2	13,200.0	13,200.0	4.2	13,200.0	13,200.0	20.9	46.0	0.5	20.9	0.0	88.7
U-Waved-160-45-SSTR3	12,810.0	12,840.0	3.5	12,840.0	12,840.0	21.0	45.5	2.3	21.0	-0.5	73.2
U-Waved-160-45-SSTR4		11,340.0	1,800.0				1,800.0	40.7	21.3	-0.5	180.0
U-Waved-160-45-SSTR5		10,860.0	1,800.0				1,800.0	18.8	23.1	2.2	360.0
U-Waved-160-45-SSTR6	10,397.1	10,620.0	53.7	10,440.0	10,620.0	1,720.1	1,800.0	11.3	26.1	0.0	194.7
U-Waved-160-75-SSTR1	20,940.0	20,940.0	12.9	20,940.0	20,940.0	16.7	46.1	8.3	16.6	0.0	85.0
U-Waved-160-75-SSTR2	19,320.0	19,320.0	3.2	19,320.0	19,320.0	19.2	41.4	7.5	19.1	0.0	56.9
U-Waved-160-75-SSTR3	18,600.0	18,660.0	9.7	18,600.0	18,660.0	1,773.2	1,800.0	4.2	17.1	0.0	84.1
U-Waved-160-75-SSTR4	16,800.0	16,860.0	421.4	16,800.0	16,860.0	1,360.2	1,800.0	22.8	18.4	-0.4	360.0
U-Waved-160-75-SSTR5		15,840.0	1,800.0				1,800.0	14.8	19.5	-0.4	360.0
U-Waved-160-75-SSTR6		15,300.0	1,800.0				1,800.0	17.3	20.1	0.4	360.0
U-Waved-160-105-SSTR1	26,040.0	26,040.0	6.0	26,040.0	26,040.0	14.7	35.3	11.3	14.7	0.0	55.0
U-Waved-160-105-SSTR2	23,760.0	23,760.0	14.4	23,760.0	23,760.0	16.1	46.5	8.1	16.0	0.0	48.4
U-Waved-160-105-SSTR3	23,640.0	23,640.0	3.4	23,640.0	23,640.0	16.1	35.5	4.1	16.0	0.0	52.9
U-Waved-160-105-SSTR4	22,440.0	22,440.0	17.0	22,440.0	22,440.0	15.7	48.3	29.1	15.6	0.0	97.6
U-Waved-160-105-SSTR5	20,820.0	20,820.0	175.1	20,820.0	20,820.0	17.2	209.3	11.2	17.0	0.0	221.6
U-Waved-160-105-SSTR6	20,200.0	20,220.0	826.7	20,220.0	20,220.0	19.4	865.3	7.7	19.3	0.0	360.0
U-Waveless-160-45-SSTR1		10,860.0	1,800.0				1,800.0	24.3	18.3	0.0	360.0
U-Waveless-160-45-SSTR2		10,260.0	1,800.0				1,800.0	19.9	19.8	0.0	360.0
U-Waveless-160-45-SSTR3		9,660.0	1,800.0				1,800.0	18.6	20.4	0.0	360.0
U-Waveless-160-45-SSTR4		11,400.0	1,800.0				1,800.0	31.1	20.1	-3.3	360.0
U-Waveless-160-45-SSTR5		10,380.0	1,800.0				1,800.0	20.2	21.9	2.3	360.0
U-Waveless-160-45-SSTR6		8,940.0	1,800.0				1,800.0	16.1	23.3	2.6	360.0
U-Waveless-160-75-SSTR1	16,200.0	16,200.0	1,298.1	16,200.0	16,200.0	15.9	1,329.6	27.4	15.6	0.7	360.0
U-Waveless-160-75-SSTR2		14,940.0	1,800.0				1,800.0	25.3	17.4	1.2	360.0
U-Waveless-160-75-SSTR3		14,400.0	1,800.0				1,800.0	19.6	17.4	0.8	360.0
U-Waveless-160-75-SSTR4		15,240.0	1,800.0				1,800.0	41.7	17.5	1.9	360.0
U-Waveless-160-75-SSTR5		14,220.0	1,800.0				1,800.0	23.6	18.8	2.5	360.0
U-Waveless-160-75-SSTR6		13,380.0	1,800.0				1,800.0	19.3	20.7	0.4	360.0
U-Waveless-160-105-SSTR1	22,800.0	22,800.0	18.9	22,800.0	22,800.0	13.3	45.4	31.3	13.2	0.3	88.0
U-Waveless-160-105-SSTR2	20,220.0	20,220.0	24.4	20,220.0	20,220.0	14.4	53.1	26.7	14.3	0.0	73.8
U-Waveless-160-105-SSTR3	19,350.0	19,380.0	43.1	19,380.0	19,380.0	14.6	72.2	11.5	14.5	0.0	125.7
U-Waveless-160-105-SSTR4		20,340.0	1,800.0				1,800.0	37.2	14.8	0.9	360.0
U-Waveless-160-105-SSTR5		18,840.0	1,800.0				1,800.0	25.8	15.7	0.6	360.0
U-Waveless-160-105-SSTR6	17,892.0	17,940.0	1,125.4	17,940.0	17,940.0	17.9	1,161.0	9.4	17.8	0.0	360.0
E-Waved-160-45-SSTR1		13,020.0	1,800.0				1,800.0	11.1	20.1	-5.9	360.0
E-Waved-160-45-SSTR2		11,640.0	1,800.0				1,800.0	17.5	20.8	1.0	360.0
E-Waved-160-45-SSTR3		11,100.0	1,800.0				1,800.0	14.6	21.0	2.1	360.0
E-Waved-160-45-SSTR4		10,800.0	1,800.0				1,800.0	38.3	21.8	5.3	360.0
E-Waved-160-45-SSTR5		10,020.0	1,800.0				1,800.0	21.6	24.3	1.8	360.0
E-Waved-160-45-SSTR6		9,600.0	1,800.0				1,800.0	16.3	26.2	0.6	360.0
E-Waved-160-75-SSTR1	17,760.0	17,760.0	88.8	17,760.0	17,760.0	17.9	124.5	13.9	17.8	0.0	163.6
E-Waved-160-75-SSTR2	16,260.0	16,260.0	230.8	16,260.0	16,260.0	18.7	268.2	17.0	18.6	0.4	221.8
E-Waved-160-75-SSTR3	15,900.0	15,900.0	60.5	15,900.0	15,900.0	19.2	98.8	14.7	19.1	0.4	360.0
E-Waved-160-75-SSTR4		15,780.0	1,800.0				1,800.0	25.9	19.3	3.3	360.0
E-Waved-160-75-SSTR5		15,420.0	1,800.0				1,800.0	15.6	19.8	0.8	360.0
E-Waved-160-75-SSTR6		15,060.0	1,800.0				1,800.0	10.4	23.1	0.0	360.0
E-Waved-160-105-SSTR1	21,600.0	21,660.0	46.3	21,600.0	21,660.0	1,738.2	1,800.0	6.4	15.6	0.0	115.6
E-Waved-160-105-SSTR2	19,560.0	19,620.0	108.6	19,560.0	19,620.0	1,675.3	1,800.0	10.7	16.1	0.0	155.6
E-Waved-160-105-SSTR3		18,960.0	1,800.0				1,800.0	9.8	17.5	0.0	360.0
E-Waved-160-105-SSTR4		18,780.0	1,800.0				1,800.0	18.2	17.4	0.6	360.0
E-Waved-160-105-SSTR5		17,640.0	1,800.0				1,800.0	14.6	17.9	0.7	360.0
E-Waved-160-105-SSTR6		17,220.0	1,800.0				1,800.0	8.0	19.2	0.3	360.0
E-Waveless-160-45-SSTR1		11,820.0	1,800.0				1,800.0	21.8	18.4	-1.0	360.0
E-Waveless-160-45-SSTR2		10,560.0	1,800.0				1,800.0	13.1	19.0	0.6	360.0
E-Waveless-160-45-SSTR3		9,600.0	1,800.0				1,800.0	21.9	20.1	2.4	360.0
E-Waveless-160-45-SSTR4		10,560.0	1,800.0				1,800.0	40.9	21.4	0.0	360.0
E-Waveless-160-45-SSTR5		10,140.0	1,800.0				1,800.0	17.2	21.2	1.7	360.0
E-Waveless-160-45-SSTR6		9,120.0	1,800.0				1,800.0	21.1	23.4	2.6	360.0
E-Waveless-160-75-SSTR1	16,920.0	17,400.0	128.3	16,920.0	17,400.0	1,656.3	1,800.0	15.9	15.4	0.0	360.0
E-Waveless-160-75-SSTR2		15,060.0	1,800.0				1,800.0	13.1	16.2	0.0	360.0
E-Waveless-160-75-SSTR3		14,100.0	1,800.0				1,800.0	18.3	16.6	0.0	360.0
E-Waveless-160-75-SSTR4		16,980.0	1,800.0				1,800.0	22.3	17.5	0.0	360.0
E-Waveless-160-75-SSTR5		14,700.0	1,800.0				1,800.0	14.3	18.9	0.4	360.0
E-Waveless-160-75-SSTR6		13,740.0	1,800.0				1,800.0	11.8	20.2	0.4	360.0
E-Waveless-160-105-SSTR1	22,620.0	22,620.0	31.7	22,620.0	22,620.0	14.8	61.3	10.3	14.8	0.0	137.7
E-Waveless-160-105-SSTR2	19,950.0	19,980.0	154.5	19,980.0	19,980.0	14.9	184.3	9.0	14.9	0.0	360.0
E-Waveless-160-105-SSTR3	19,106.4	19,140.0	1,217.6	19,140.0	19,140.0	15.5	1,248.5	6.6	15.4	0.0	360.0
E-Waveless-160-105-SSTR4	20,700.0	20,700.0	17.6	20,700.0	20,700.0	16.1	49.7	15.9	16.0	0.0	65.4
E-Waveless-160-105-SSTR5		18,300.0	1,800.0				1,800.0	14.4	16.2	0.3	360.0
E-Waveless-160-105-SSTR6	17,993.2	18,000.0	532.2	18,000.0	18,000.0	19.0	570.1	6.7	18.9	0.0	360.0

Appendix I: Including different break time structures in solution procedures

The shift structure to include breaks as discussed in Section 3 is very flexible, whereas there might be other break types as discussed in Appendix D. This is actually the situation in our case study as discussed in Section 6.3, where there are three break types: two 15-minute breaks after 2 hours and 6 hours of starting the shift and one 30-minute break after 3.5 hours of starting the shift. To find a solution to the OPSP with the metaheuristic where these new break structures are included, we can directly include the constraints as formulated in Appendix D to the problem formulation. In particular, $\Phi = \{1, 2, 3\}$ where break type 1 and 2 indicate the two shorter breaks and break type 3 indicates the longer break. Consequently, the duration of the each break type is given by $l_b^1 = l_b^2 = 0.25$ and $l_b^3 = 0.5$, whereas the start time of each break type is given by $r^1 = 2$, $r^2 = 5$ and $r^3 = 3.5$. In the case where the start time of the breaks can have a 15-minute flexibility (i.e., scheduled at most 15 minutes earlier or later), we indicate the flexibility in the timing of the breaks by Ψ, which is set to 15 minutes (or $\Psi = 0.25$ hours). When there is no such flexibility (i.e., in the scenarios 1, 5, 6 and 7), the value of Ψ equals zero.

To use the metaheuristic, we have to reformulate the reduced problem to verify whether two schedules can be combined for one order picker (as formulated in Appendix G). Let us first redefine some of the decision variables:

- x_{ik} is 1 if batch $i \in B$ is scheduled to be picked at the k^{th} position in the shift for the order picker, where $k \in K$, else 0
- y_k^ϕ is 1 if a break $\phi \in \Phi$ is scheduled at the k^{th} position in the shift for the order picker, where $k \in K$, else 0
- s_j is 1 if the order picker starts the shift at the beginning of period $j \in S$, else 0
- e_j is 1 if the order picker ends the shift at the end of period $j \in E$, else 0
- c_k completion time of the task scheduled at the k^{th} position in the shift of the order picker, where $k \in K$
- m amount of time for which the order picker is compensated

The reduced problem (RP) of the OPSP for the case study is then reformulated as a MILP model as follows:

RPCase:

$$\min m \tag{92}$$

subject to

$$\sum_{i \in B} x_{ik} + \sum_{\phi \in \Phi} y_k^\phi \leq 1 \qquad \forall k \in K \tag{93}$$

$$\sum_{k \in K} x_{ik} = 1 \qquad \forall i \in B \tag{94}$$

$$\sum_{j \in J \setminus S} s_j + \sum_{j \in J \setminus E} e_j = 0 \tag{95}$$

$$c_1 \geq \Big(\sum_{j \in S} (j-1) s_j\Big) l + \sum_{i \in B} t_i x_{i1} + \sum_{\phi \in \Phi} l_b^\phi y_k^\phi \tag{96}$$

$$c_k \geq c_{k-1} + \sum_{i \in B} t_i x_{ik} + \sum_{\phi \in \Phi} l_b^\phi y_k^\phi \qquad \forall k \in K \setminus \{1\} \tag{97}$$

$$\sum_{j \in E} (j e_j) l \geq c_{\bar{k}} \tag{98}$$

$$c_k + M(1 - x_{ik}) \geq r_i \qquad \forall i \in B, k \in K \tag{99}$$

$$c_k - M(1 - x_{ik}) \leq d_i \qquad \forall i \in B, k \in K \tag{100}$$

$$\sum_{k \in K} y_k^1 = \sum_{k \in K} y_k^3 = 1 \tag{101}$$

$$M \sum_{k \in K} y_k^2 \geq m - 360 \tag{102}$$

$$c_k + M(1 - y_k^\phi) - \Big(\sum_{j \in S}(j-1)s_j\Big)l \geq r^\phi - \Psi \qquad \forall k \in K, \phi \in \Phi \tag{103}$$

$$c_k + M(1 - y_k^\phi) - \Big(\sum_{j \in S}(j-1)s_j\Big)l \leq r^\phi + \Psi \qquad \forall k \in K, \phi \in \Phi \tag{104}$$

$$\sum_{i \in B} x_{ik-1} + \sum_{\phi \in \Phi} y_k^\phi \geq \sum_{i \in B} x_{ik} + \sum_{\phi \in R} y_k^\phi \qquad \forall k \in K \setminus \{1\} \tag{105}$$

$$(\sum_{j \in E} je_j - \sum_{j \in S}(j-1)s_j)l \leq m \tag{106}$$

$$c_k \geq 0 \qquad \forall k \in K \tag{107}$$

$$x_{ik} \in \{0, 1\} \qquad \forall i \in B, k \in K \tag{108}$$

$$y_k^\phi \in \{0, 1\} \qquad \forall k \in K, \phi \in \Phi \tag{109}$$

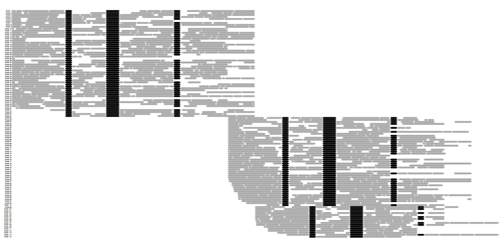
$$s_j, e_j \in \{0, 1\} \qquad \forall j \in J \tag{110}$$

$$T_{min} \leq m \leq T_{max} \tag{111}$$

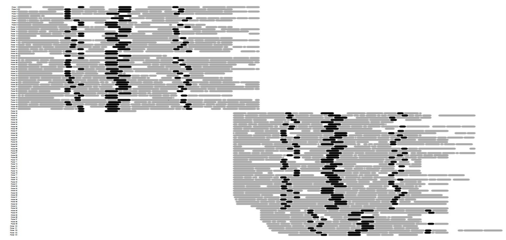
Most of the constraints in this reformulation are the same as the original reduced problem formulation in Appendix G. Here, we only explain the constraints that are different. Constraint (101) enforces that the first short break and the long break have to be scheduled, whereas constraint (102) indicates that the second short break only has to be scheduled if the order picker is employed for more than 6 hours. Constraints (103) and (104) regulate the start and completion times of individual breaks according to the flexibility Ψ for the break times.

Appendix J: Detailed result of Section 6.3

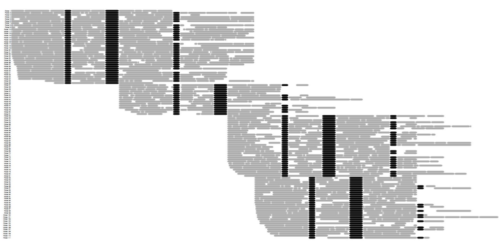
The schedules resulting from the OPSP on Day 1 of the normal week are illustrated in Figure 12 for the scenarios where T_{min} equals 6 hours. The total number of hours that the order pickers are compensated for is indicated in Table 14, whereas as the number of scheduled order pickers is presented in Table 15. When dividing these two numbers for each instance, we obtain the average shift length. The number of order pickers who are compensated for exactly T_{min} time unit is included in Table 16. A summary of these results is included in Figure 8.



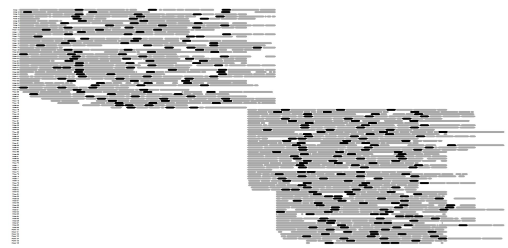
(a) Schedule scenario 1: fixed break times, 3 start times



(b) Schedule scenario 2: 15-minute flexible break times, 3 start times



(c) Schedule scenario 5: fixed break times, 4 start times



(d) Schedule scenario 11: completely flexible break times, 3 start times

Figure 12 **Gantt charts illustrating the scheduling of order picking tasks (in grey) and breaks (in black) for the four scenarios where $T_{min} = 6$ hours**

Table 14 **Number of hours that scheduled order pickers need to be compensated for per scenario**

		Scenario											
Week	Day	1	2	3	4	5	6	7	8	9	10	11	12
Normal	1	897	800	851	916	813	828	910	779	791	831	783	750
	2	933	839	899	984	928	996	1,056	838	885	912	861	844
	3	1,071	991	1,023	1,091	1,074	1,090	1,163	956	1,002	1,072	970	932
	4	1,004	915	966	1,010	958	1,023	1,064	892	917	1,010	899	870
	5	1,131	1,021	1,053	1,123	1,069	1,119	1,144	994	1,017	1,049	971	971
	6	1,297	1,177	1,221	1,264	1,221	1,293	1,315	1,144	1,166	1,203	1,149	1,114
	7	1,283	1,158	1,213	1,274	1,262	1,339	1,455	1,148	1,161	1,274	1,170	1,146
Busy	1	783	726	758	816	753	819	861	705	712	777	706	686
	2	929	835	835	879	854	854	906	811	830	830	823	779
	3	990	901	901	967	921	945	945	860	873	876	878	829
	4	1,063	971	1,013	1,014	1,006	1,054	1,090	955	977	975	972	945
	5	1,544	1,420	1,495	1,495	1,454	1,454	1,574	1,367	1,392	1,398	1,435	1,393
	6	1,713	1,625	1,639	1,668	1,669	1,744	1,730	1,584	1,619	1,651	1,620	1,612
	7	1,412	1,288	1,375	1,375	1,374	1,391	1,447	1,256	1,280	1,350	1,293	1,239

Appendix K: Preemptive breaks

The problem formulation in Section 3 does not allow preemptive order picking, i.e., the order picking of a batch has to be completed before the order picker can take a break. In this section, we study the operational environment where the items of a batch can be partially picked before the order picker takes a break. We can derive an exact algorithm for the preemptive scheduling case after making the necessary changes to the resource extension functions of the pricing problem for the non-preemptive problem presented in Section 3. All other components of the pricing problem (e.g., the resources, their definition, feasibility windows and dominance rules) remain the same as in the non-preemptive scheduling problem.

Table 15 Number of scheduled order pickers per scenario

		Scenario											
Week	Day	1	2	3	4	5	6	7	8	9	10	11	12
Normal	1	115	104	108	111	111	105	110	106	104	101	110	105
	2	125	117	118	120	128	132	129	115	116	112	117	124
	3	140	135	133	134	144	139	140	131	132	131	133	132
	4	133	123	127	123	132	133	130	123	122	123	126	123
	5	147	137	136	136	139	144	138	134	133	127	135	135
	6	167	153	155	153	161	166	157	151	151	146	154	150
	7	170	153	155	154	170	171	174	155	147	154	156	151
Busy	1	101	96	97	98	100	102	103	97	91	92	96	93
	2	111	110	103	105	109	105	108	106	102	100	105	104
	3	120	114	112	115	117	120	111	110	109	105	109	109
	4	131	124	126	121	128	130	130	122	121	117	126	124
	5	185	175	182	177	188	176	186	172	173	166	173	176
	6	206	199	199	197	204	206	201	196	197	195	195	201
	7	172	160	167	165	170	170	170	161	159	163	159	162

Table 16 Number of scheduled order pickers that get compensated for T_{min} time units per scenario

		Scenario											
Week	Day	1	2	3	4	5	6	7	8	9	10	11	12
Normal	1	24	22	48	83	37	46	80	18	42	64	51	41
	2	41	47	64	96	50	76	105	42	72	94	47	59
	3	30	43	66	115	52	61	97	48	65	98	45	48
	4	36	40	76	97	52	71	106	49	68	91	49	49
	5	39	45	67	101	35	71	98	53	72	90	52	52
	6	32	27	59	113	44	71	98	33	68	103	28	35
	7	56	44	70	112	62	81	111	40	45	83	52	35
Busy	1	28	28	46	66	36	35	66	14	31	54	41	33
	2	8	21	32	66	25	31	66	29	39	62	18	21
	3	13	20	37	68	24	36	54	27	39	65	21	27
	4	12	24	38	75	28	38	80	26	36	69	31	29
	5	13	25	44	98	37	39	100	21	40	88	25	28
	6	20	22	48	105	30	30	79	32	54	96	27	21
	7	19	22	32	110	27	39	83	21	37	102	30	32

The resource extension functions without breaks are given by the same functions $f(\cdot)$ as specified in Section 4.2 4.2. The only difference is in extension functions with breaks. A distinction has to be made between two conditions. First, if $T_i^{work} + t_j$ is greater than T_{break}, then batch j cannot be completed before the order picker takes a break. The portion of batch j that can be completed before the break is scheduled has a duration of $T_i^{work} + t_j - T_{break}$ time units and the remaining items of the batch can be picked after the break. The resource extension functions under this first condition (denoted as $g'(\cdot)$) are defined as:

$$T_j^{time} = g'^{time}(T_i, j) := \max\{T_i^{time} + t_j + l_b, r_j\} \tag{112}$$

$$T_j^{dur} = g'^{dur}(T_i, j) := \max\{T_i^{dur} + t_j + l_b, r_j - T_i^{start}\} \tag{113}$$

$$T_j^{work} = g'^{work}(T_i, j) := T_i^{work} + t_j - T_{break} \tag{114}$$

$$T_j^{brk} = g'^{brk}(T_i, j) := \min\{d_j - T_i^{work} - t_j + T_{break}, \infty\} \tag{115}$$

Under the second condition, if $T_i^{work} + t_j$ is less than or equal to T_{break}, then the complete batch can be picked before the order picker takes a break. The resource extension functions under this second condition are denoted by $g''(\cdot)$ and defined as follows:

$$T_j^{time} = g''^{time}(T_i, j) := \max\{T_i^{time} + t_j + l_b, r_j\} \tag{116}$$

$$T_j^{dur} = g''^{dur}(T_i, j) := \max\{T_i^{dur} + t_j + l_b, r_j - T_i^{start}\} \tag{117}$$

$$T_j^{work} = g''^{work}(T_i, j) := 0 \tag{118}$$

$$T_j^{brk} = g''^{brk}(T_i, j) := \min\{d_j, \infty\} \tag{119}$$

For a numerical test bed with instances of 40 batches and 5 or 10 truck departures per staging lane, we present the performance of the branch-and-price algorithm when preemptive order scheduling is allowed in Table 17 and 18, respectively. Note that the counterparts with non-preemptive order scheduling are presented in Table 6 and 7, respectively. The last column in these two tables indicates the relative cost increase of using preemption against non-preemption, which is expressed as $\%\Delta^P = ((z^P - z^{NP})/z^{NP}) \times 100$, where z^P and z^{NP} are the best branch-and-price solutions for preemtive and non-preemptive order scheduling, respectively.

The results illustrate the adaptability of our solution framework. For three of the instances, the algorithm for preemptive order picking was able to generate only inferior solutions compared to the algorithm for non-preemptive order picking within the limited run time. Excluding these three instances, the cost savings of preemptive order scheduling are on average 0.4% compared to non-preemptive order scheduling, with the largest cost savings of 8.3% However, the problem formulation with preemptive order scheduling assumes that breaks occur instantaneously and they ignore the time it takes to travel between the location of the last picked item in the pick tour and the location where the order picker takes a break. A more accurate approach to include preemptive order picking would be to include this extra travel time. Consequently, preemptive order scheduling can be more expensive than non-preemptive order scheduling. Therefore, studying preemptive order scheduling can have a high practical and academic relevance but is beyond the scope of our work.

Table 17 Results for Instances with 40 batches, 5 truck departures per staging lane, preemptive order picking

	Branch & Price							
	Linear Relaxation			Integer				
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{P}$
U-Waved-40-45-SSTR1	4,480.0	4,800.0	4.7	4,500.0	4,800.0	1,793.3	1,800.0	0.0
U-Waved-40-45-SSTR2	4,200.0	4,440.0	3.4	4,200.0	4,440.0	1,793.0	1,800.0	0.0
U-Waved-40-45-SSTR3	4,152.0	4,200.0	2.1	4,200.0	4,200.0	2.8	7.7	0.0
U-Waved-40-45-SSTR4	4,480.0	4,800.0	4.4	4,500.0	4,800.0	1,793.4	1,800.0	0.0
U-Waved-40-45-SSTR5	3,480.0	3,660.0	148.4	3,480.0	3,660.0	1,649.0	1,800.0	0.0
U-Waved-40-45-SSTR6	2,685.0	2,760.0	17.2	2,700.0	2,700.0	1,636.2	1,656.2	0.0
U-Waved-40-75-SSTR1	4,980.0	4,980.0	1.8	4,980.0	4,980.0	2.1	6.0	0.0
U-Waved-40-75-SSTR2	4,740.0	4,740.0	1.4	4,740.0	4,740.0	2.4	6.2	0.0
U-Waved-40-75-SSTR3	4,660.0	4,740.0	1.2	4,680.0	4,680.0	8.0	11.5	0.0
U-Waved-40-75-SSTR4	4,800.0	4,800.0	2.0	4,800.0	4,800.0	2.3	6.6	0.0
U-Waved-40-75-SSTR5	3,840.0	3,840.0	40.0	3,840.0	3,840.0	2.4	44.7	0.0
U-Waved-40-75-SSTR6	3,400.0	3,420.0	6.1	3,420.0	3,420.0	2.4	10.8	0.0
U-Waved-40-105-SSTR1	5,600.0	5,760.0	1.1	5,640.0	5,760.0	1,796.9	1,800.0	0.0
U-Waved-40-105-SSTR2	5,280.0	5,400.0	0.8	5,280.0	5,400.0	1,797.0	1,800.0	0.0
U-Waved-40-105-SSTR3	5,160.0	5,160.0	0.4	5,160.0	5,160.0	2.5	5.5	0.0
U-Waved-40-105-SSTR4	5,320.0	5,460.0	6.9	5,340.0	5,460.0	1,790.7	1,800.0	-1.1
U-Waved-40-105-SSTR5	4,936.0	5,040.0	14.3	4,980.0	5,040.0	1,783.0	1,800.0	1.2
U-Waved-40-105-SSTR6	4,771.2	4,800.0	3.5	4,800.0	4,800.0	2.8	8.9	0.0
U-Waveless-40-45-SSTR1	3,040.0	3,360.0	7.5	3,060.0	3,360.0	1,790.4	1,800.0	0.0
U-Waveless-40-45-SSTR2	2,360.0	2,520.0	14.1	2,400.0	2,520.0	1,783.9	1,800.0	0.0
U-Waveless-40-45-SSTR3	2,181.2	2,280.0	27.0	2,220.0	2,280.0	1,770.7	1,800.0	0.0
U-Waveless-40-45-SSTR4	2,640.0	2,940.0	335.7	2,640.0	2,940.0	1,462.0	1,800.0	0.0
U-Waveless-40-45-SSTR5	2,280.0	2,340.0	480.6	2,280.0	2,340.0	1,317.1	1,800.0	0.0
U-Waveless-40-45-SSTR6	2,168.2	2,280.0	29.7	2,220.0	2,280.0	1,767.9	1,800.0	0.0
U-Waveless-40-75-SSTR1	3,960.0	3,960.0	16.1	3,960.0	3,960.0	1.7	19.5	-8.3
U-Waveless-40-75-SSTR2	3,505.3	3,600.0	59.8	3,540.0	3,540.0	383.4	445.0	0.0
U-Waveless-40-75-SSTR3	3,360.0	3,420.0	11.4	3,360.0	3,360.0	986.5	1,000.3	0.0
U-Waveless-40-75-SSTR4	3,720.0	3,900.0	129.7	3,720.0	3,900.0	1,668.3	1,800.0	0.0
U-Waveless-40-75-SSTR5	3,441.1	3,540.0	268.4	3,480.0	3,540.0	1,529.6	1,800.0	0.0
U-Waveless-40-75-SSTR6	3,281.1	3,360.0	18.0	3,300.0	3,360.0	1,779.9	1,800.0	0.0
U-Waveless-40-105-SSTR1	6,830.0	6,840.0	0.7	6,840.0	6,840.0	1.6	3.8	-5.0
U-Waveless-40-105-SSTR2	5,382.9	5,400.0	0.6	5,400.0	5,400.0	1.7	3.9	-1.1
U-Waveless-40-105-SSTR3	4,740.0	4,800.0	0.8	4,740.0	4,800.0	1,797.5	1,800.0	0.0
U-Waveless-40-105-SSTR4	5,760.0	5,760.0	3.5	5,760.0	5,760.0	1.9	7.2	0.0
U-Waveless-40-105-SSTR5	5,040.0	5,040.0	2.2	5,040.0	5,040.0	1.9	6.0	0.0
U-Waveless-40-105-SSTR6	4,620.0	4,620.0	1.5	4,620.0	4,620.0	2.1	5.7	0.0
E-Waved-40-45-SSTR1	4,140.0	4,380.0	6.5	4,140.0	4,380.0	1,791.5	1,800.0	0.0
E-Waved-40-45-SSTR2	3,769.7	3,960.0	3.5	3,780.0	3,960.0	1,793.0	1,800.0	0.0
E-Waved-40-45-SSTR3	3,718.3	3,780.0	2.3	3,720.0	3,780.0	1,794.5	1,800.0	0.0
E-Waved-40-45-SSTR4	4,080.0	4,320.0	5.6	4,080.0	4,320.0	1,792.2	1,800.0	0.0
E-Waved-40-45-SSTR5	3,156.6	3,360.0	1,069.5	3,180.0	3,360.0	727.7	1,800.0	0.0
E-Waved-40-45-SSTR6	2,464.5	2,580.0	55.8	2,520.0	2,580.0	1,741.1	1,800.0	2.4
E-Waved-40-75-SSTR1	5,580.0	5,880.0	1.5	5,580.0	5,880.0	1,796.3	1,800.0	0.0
E-Waved-40-75-SSTR2	5,010.0	5,160.0	1.2	5,040.0	5,160.0	1,796.5	1,800.0	-1.1
E-Waved-40-75-SSTR3	5,000.0	5,040.0	3.4	5,040.0	5,040.0	2.3	8.1	-1.2
E-Waved-40-75-SSTR4	5,460.0	5,760.0	4.7	5,460.0	5,760.0	1,793.2	1,800.0	0.0
E-Waved-40-75-SSTR5	4,530.0	4,620.0	5.1	4,560.0	4,620.0	1,792.6	1,800.0	0.0
E-Waved-40-75-SSTR6	4,260.0	4,260.0	2.1	4,260.0	4,260.0	2.3	6.6	0.0
E-Waved-40-105-SSTR1	5,460.0	5,460.0	0.4	5,460.0	5,460.0	2.2	4.8	0.0
E-Waved-40-105-SSTR2	4,980.0	4,980.0	0.6	4,980.0	4,980.0	2.5	5.5	0.0
E-Waved-40-105-SSTR3	4,860.0	4,860.0	0.5	4,860.0	4,860.0	2.7	5.8	0.0
E-Waved-40-105-SSTR4	4,800.0	4,800.0	3.2	4,800.0	4,800.0	2.4	7.9	0.0
E-Waved-40-105-SSTR5	3,846.3	3,900.0	727.3	3,900.0	3,900.0	2.9	732.9	0.0
E-Waved-40-105-SSTR6	3,442.2	3,480.0	162.1	3,480.0	3,480.0	3.1	168.2	0.0
E-Waveless-40-45-SSTR1	2,920.0	2,940.0	163.1	2,940.0	2,940.0	1.9	166.9	0.0
E-Waveless-40-45-SSTR2	2,352.3	2,520.0	165.9	2,400.0	2,400.0	1,060.3	1,228.2	0.0
E-Waveless-40-45-SSTR3	2,099.1	2,160.0	27.4	2,100.0	2,160.0	1,770.6	1,800.0	0.0
E-Waveless-40-45-SSTR4	2,720.0	2,940.0	691.1	2,760.0	2,940.0	1,106.9	1,800.0	0.0
E-Waveless-40-45-SSTR5	2,349,9	2,520.0	168.1	2,400.0	2,460.0	1,641.1	1,800.0	2.5
E-Waveless-40-45-SSTR6	2,111.8	2,160.0	23.8	2,160.0	2,160.0	2.6	28.8	0.0
E-Waveless-40-75-SSTR1	5,070.0	5,340.0	5.0	5,100.0	5,340.0	1,793.3	1,800.0	0.0
E-Waveless-40-75-SSTR2	4,120.0	4,260.0	128.9	4,140.0	4,260.0	1,669.4	1,800.0	0.0
E-Waveless-40-75-SSTR3	3,983.4	4,020.0	25.3	4,020.0	4,020.0	1.8	28.9	0.0
E-Waveless-40-75-SSTR4	4,279.1	4,380.0	535.8	4,320.0	4,380.0	1,262.5	1,800.0	0.0
E-Waveless-40-75-SSTR5	4,035.0	4,080.0	267.0	4,080.0	4,080.0	2.1	271.0	0.0
E-Waveless-40-75-SSTR6	3,925.7	3,960.0	19.3	3,960.0	3,960.0	2.2	23.7	0.0
E-Waveless-40-105-SSTR1	5,280.0	5,280.0	0.6	5,280.0	5,280.0	1.6	3.7	0.0
E-Waveless-40-105-SSTR2	3,960.0	3,960.0	4.5	3,960.0	3,960.0	1.6	7.7	0.0
E-Waveless-40-105-SSTR3	3,402.0	3,480.0	8.7	3,420.0	3,480.0	1,789.6	1,800.0	0.0
E-Waveless-40-105-SSTR4	4,800.0	4,800.0	2.8	4,800.0	4,800.0	1.9	6.5	0.0
E-Waveless-40-105-SSTR5	3,900.0	3,900.0	51.9	3,900.0	3,900.0	1.9	55.6	0.0
E-Waveless-40-105-SSTR6	3,399.4	3,480.0	6.5	3,420.0	3,480.0	1,791.6	1,800.0	0.0

Table 18 Results for Instances with 40 batches, 10 truck departures per staging lane, preemptive order picking

	Branch & Price for Preemptive Order Picking							
	Linear Relaxation			Integer				
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{P}$
U-Waved-40-45-SSTR1	4,800.00	4,800.00	4	4,800.00	4,800.00	1.8	7.6	0.0
U-Waved-40-45-SSTR2	4,140.00	4,140.00	0.4	4,140.00	4,140.00	2.1	4.6	0.0
U-Waved-40-45-SSTR3	3,660.00	3,660.00	0.6	3,660.00	3,660.00	2	4.6	0.0
U-Waved-40-45-SSTR4	2,880.00	3,360.00	21.8	2,880.00	2,880.00	192	216.1	0.0
U-Waved-40-45-SSTR5	2,700.00	3,000.00	70.6	2,700.00	2,700.00	1,115.10	1,185.70	-4.3
U-Waved-40-45-SSTR6	2,640.00	2,760.00	4.1	2,640.00	2,760.00	1,793.50	1,800.00	0.0
U-Waved-40-75-SSTR1	6,240.00	6,240.00	0.2	6,240.00	6,240.00	1.5	3.2	0.0
U-Waved-40-75-SSTR2	5,400.00	5,400.00	0.6	5,400.00	5,400.00	1.7	4	0.0
U-Waved-40-75-SSTR3	4,920.00	4,920.00	0.4	4,920.00	4,920.00	1.7	3.8	0.0
U-Waved-40-75-SSTR4	4,320.00	4,320.00	1.6	4,320.00	4,320.00	1.7	4.9	0.0
U-Waved-40-75-SSTR5	3,780.00	3,780.00	4.5	3,780.00	3,780.00	1.8	8	0.0
U-Waved-40-75-SSTR6	3,630.00	3,660.00	3.3	3,660.00	3,660.00	2.1	7.4	0.0
U-Waved-40-105-SSTR1	7,680.00	7,680.00	0.1	7,680.00	7,680.00	1.4	2.8	0.0
U-Waved-40-105-SSTR2	6,960.00	6,960.00	0.2	6,960.00	6,960.00	1.5	3.2	0.0
U-Waved-40-105-SSTR3	6,480.00	6,480.00	0.2	6,480.00	6,480.00	1.5	3.2	0.0
U-Waved-40-105-SSTR4	5,760.00	5,760.00	0.9	5,760.00	5,760.00	1.5	3.9	0.0
U-Waved-40-105-SSTR5	5,190.00	5,220.00	1.1	5,220.00	5,220.00	1.7	4.5	0.0
U-Waved-40-105-SSTR6	5,040.00	5,040.00	0.3	5,040.00	5,040.00	2	4.2	0.0
E-Waved-40-45-SSTR1	4,000.00	4,320.00	2.9	4,020.00	4,320.00	1,795.20	1,800.00	0.0
E-Waved-40-45-SSTR2	3,410.00	3,600.00	3.3	3,420.00	3,600.00	1,794.60	1,800.00	0.0
E-Waved-40-45-SSTR3	3,110.00	3,300.00	1.4	3,120.00	3,300.00	1,796.40	1,800.00	0.0
E-Waved-40-45-SSTR4	2,760.00	2,940.00	102.7	2,760.00	2,940.00	1,695.40	1,800.00	0.0
E-Waved-40-45-SSTR5	2,600.00	2,640.00	481.8	2,640.00	2,640.00	2.2	486	0.0
E-Waved-40-45-SSTR6	2,580.00	2,640.00	98.2	2,580.00	2,640.00	1,699.60	1,800.00	0.0
E-Waved-40-75-SSTR1	4,680.00	5,280.00	2.8	4,680.00	5,280.00	1,795.60	1,800.00	0.0
E-Waved-40-75-SSTR2	3,921.40	4,260.00	6.5	3,960.00	4,260.00	1,791.80	1,800.00	0.0
E-Waved-40-75-SSTR3	3,441.40	3,720.00	2.1	3,480.00	3,720.00	1,796.10	1,800.00	0.0
E-Waved-40-75-SSTR4	3,480.00	3,840.00	22.4	3,480.00	3,840.00	1,775.80	1,800.00	0.0
E-Waved-40-75-SSTR5	3,180.00	3,300.00	387.1	3,180.00	3,300.00	1,411.00	1,800.00	0.0
E-Waved-40-75-SSTR6	2,943.50	3,060.00	45.6	3,000.00	3,060.00	1,752.20	1,800.00	0.0
E-Waved-40-105-SSTR1	6,840.00	6,840.00	0.8	6,840.00	6,840.00	1.3	3.3	-5.8
E-Waved-40-105-SSTR2	5,991.10	6,000.00	73.1	6,000.00	6,000.00	1.4	75.9	-2.9
E-Waved-40-105-SSTR3	5,567.10	5,580.00	35.9	5,580.00	5,580.00	1.4	38.6	-3.1
E-Waved-40-105-SSTR4	6,300.00	6,300.00	0.5	6,300.00	6,300.00	1.4	3.2	-6.3
E-Waved-40-105-SSTR5	5,753.30	5,820.00	11.3	5,760.00	2,760.00	17.3	33.2	0.0
E-Waved-40-105-SSTR6	5,480.00	5,520.00	3	5,520.00	5,520.00	1.6	6.2	0.0